

Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 10.4(x)

First Published: 2023-08-18

Last Modified: 2026-02-10

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html. Cisco product warranty information is available at https://www.cisco.com/c/en/us/products/warranty-listing.html. US Federal Communications Commission Notices are found here https://www.cisco.com/c/en/us/products/us-fcc-notice.html.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023–2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

Trademarks ?

PREFACE

Preface xxix

Audience xxix

Document Conventions xxix

Related Documentation for Cisco Nexus 9000 Series Switches xxx

Documentation Feedback xxx

Communications, services, and additional information xxx

Cisco Bug Search Tool xxxi

Documentation feedback xxxi

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

Overview 7

Licensing Requirements 7

Supported Platforms 8

Authentication, authorization, and accounting 8

RADIUS and TACACS+ Security Protocols 9

LDAP 9

SSH and Telnet 9

User Accounts and Roles 9

IP ACLs 10

MAC ACLs 10

VACLs 10

DHCP Snooping 10

Dynamic ARP Inspection 11
IP Source Guard 11
Password Encryption 11
Keychain Management 11
Control Plane Policing 12
Rate Limits 12
Software Image 12
Virtual Device Contexts 12
SGT Tagged Packet 12
TLS Protocol Support 13

CHAPTER 3 **Configuring FIPS** 15
About FIPS 15
FIPS Self-Tests 15
FIPS error state 16
Prerequisites for FIPS 16
Guidelines and limitations for FIPS 17
Default settings for FIPS 17
Configuring FIPS 17
Enable FIPS mode 17
Disable FIPS 18
FIPS configuration verification commands 19
Create 2048 bit RSA key 19
Configuration example for FIPS 20
Additional references for FIPS 20

CHAPTER 4 **Configuring AAA** 23
About AAA 23
AAA security services 23
Benefits of using AAA 24
Remote AAA services 24
AAA server groups 25
AAA service configuration options 25
Authentication and authorization process for user login 27

AES password encryption and primary encryption keys 29
Prerequisites for AAA 29
Guidelines and limitations for AAA 29
Default settings for AAA 30
Configuring AAA 31
Process for configuring AAA 31
Configure console login authentication methods 32
Configure default login authentication methods 33
Disable fallback to local authentication 35
Enable the default user role for AAA authentication 36
Enable login authentication failure messages 37
Log successful and failed login attempts 38
Configure login block per user 39
Create AAA bypass user 41
Enable CHAP authentication 42
Enable MSCHAP or MSCHAP V2 authentication 43
Configure AAA authorization on LDAP servers 45
Configure AAA SSH-Cert-Authorization on TACACS servers 46
Configure AAA accounting default methods 47
Using AAA server VSAs with Cisco NX-OS devices 48
About VSAs 49
VSA format 49
Specifying Cisco NX-OS user roles and SNMPv3 parameters on AAA servers 50
Configuring secure login features 50
Configure login parameters 50
Restrict user login sessions 51
Restrict the password length 52
Enable the password prompt for the username 53
Configure the shared secret for RADIUS or TACACS+ 53
Monitor and clear the local AAA accounting log 54
AAA configuration verification commands 55
Configuration examples for AAA 56
Configuration examples for login parameters 56
Configuration examples for the password prompt feature 57

Additional references for AAA 58

CHAPTER 5 **Configuring RADIUS 59**

About RADIUS 59

RADIUS network environments 59

RADIUS operation 60

RADIUS server monitoring 61

Vendor-specific attributes 61

About RADIUS change of authorization 62

Session reauthentication 63

Session termination 63

Prerequisites for RADIUS 63

Guidelines and limitations for RADIUS 63

Guidelines and limitations for RadSec 64

Guidelines and limitations for RADIUS change of authorization 65

Default settings for RADIUS 65

Configuring RADIUS servers 66

RADIUS server configuration process 66

Configure RADIUS server hosts 66

Configure global RADIUS keys 67

Configure a key for a specific RADIUS server 68

Configure the RADIUS attribute message authenticator 70

Configure RadSec 71

About RADIUS with DTLS 72

Configure RADIUS with DTLS 72

Configure a RADIUS server group 74

Configure the global source interface for a RADIUS server group 75

Allow a user to specify a RADIUS server at login 76

Configure the global RADIUS transmission retry count and timeout interval 77

Configure the RADIUS transmission retry count and timeout interval for a server 78

Configure accounting and authentication attributes for a RADIUS server 79

Configure global periodic RADIUS server monitoring 80

Configure periodic RADIUS server monitoring on individual servers 81

Configure the RADIUS Dead-Time interval 82

Configuring One-Time passwords 83
Manually monitor a RADIUS server or group 84
Enable or disable the dynamic author server 84
Configure RADIUS change of authorization 85
RADIUS configuration verification commands 86
RADIUS change of authorization configuration verification commands 86
Monitor RADIUS servers 87
Clear the RADIUS server statistics 87
Configuration example for RADIUS 88
Configuration examples of RADIUS change of authorization 89
Where to go next 89
Additional references for RADIUS 89

CHAPTER 6 **Configuring TACACS+ 91**
TACACS+ 91
TACACS+ Operation for User Login 92
Default TACACS+ server obfuscation type and secret key 93
Command authorization support for TACACS+ servers 93
TACACS+ server monitoring 93
Vendor-specific attributes for TACACS+ 94
Cisco VSA format for TACACS+ 94
Prerequisites for TACACS+ 95
Guidelines and limitations for TACACS+ 95
Default settings for TACACS+ 96
One-time password support 96
Configuring TACACS+ 97
TACACS+ server configuration process 97
Enable TACACS+ 97
Configure TACACS+ server hosts 98
Configure global TACACS+ keys 99
Configure a key for a specific TACACS+ server 100
Configure TACACS+ server groups 101
Configure the global source interface for TACACS+ server groups 102
Allowing users to specify a TACACS+ server at login 103

Configure the timeout interval for a TACACS+ server 104
Configure TCP ports 105
Configure global periodic TACACS+ server monitoring 106
Configure periodic TACACS+ server monitoring on individual servers 107
Configure the TACACS+ Dead-Time interval 109
Configure ASCII authentication 110
Configure command authorization on TACACS+ servers 111
Testing command authorization on TACACS+ servers 112
Enable and disable command authorization verification 113
Configure X.509 Certificate-Based SSH authorization using TACACS server 114
Permitting or denying commands for users of privilege roles 114
Manually monitoring TACACS+ servers or groups 115
Disable TACACS+ 116
Monitor TACACS+ servers 116
Clear TACACS+ server statistics 117
TACACS+ configuration verification commands 117
Configuration examples for TACACS+ 118
Where to go next 119
Additional references for TACACS+ 119

CHAPTER 7 **Configuring LDAP** 121
About LDAP 121
LDAP authentication and authorization 121
LDAP operations for user logins 122
LDAP server monitoring 123
Vendor-specific attributes for LDAP 123
Cisco VSA format for LDAP 124
Virtualization support for LDAP 124
Prerequisites for LDAP 124
Guidelines and limitations for LDAP 124
Default settings for LDAP 125
Configuring LDAP 126
LDAP server configuration process 126
Enable the LDAP feature 126

Configure LDAP server hosts 127
Configure the RootDN for an LDAP server 128
Configure LDAP server groups 129
Configure the global LDAP timeout interval 131
Configure the timeout interval for an LDAP server 132
Configure the TCP ports 132
Configure LDAP search maps 134
Configure periodic LDAP server monitoring 135
Configure the LDAP dead-time interval 136
Configure AAA authorization on LDAP servers 137
Configure LDAP SSH public key authorization 138
Configure LDAP SSH certificate authorization 139
Monitor LDAP servers 140
Clear LDAP server statistics 141
LDAP configuration verification commands 142
LDAP configuration examples 142
Where to go next 143
Additional references for LDAP 143

CHAPTER 8 **Configuring SSH and Telnet** 145
About SSH and telnet 145
SSH server 145
SSH client 145
SSH server keys 146
SSH authentication using digital certificates 146
Telnet server 147
Post-quantum key exchange for SSH 147
Prerequisites for SSH and telnet 147
Guidelines and limitations for SSH and telnet 147
Default settings for SSH and telnet 149
Configuring SSH 149
Generating SSH server keys 149
Specifying the SSH public keys for user accounts 151
Specify the SSH public keys in IETF SECSH format 151

Specify the SSH public keys in OpenSSH format 152
Configure a maximum number of SSH login attempts 153
Starting SSH sessions 154
Starting SSH sessions from boot mode 154
Configure SSH passwordless file copy 155
Configure SCP and SFTP servers 157
Configure x.509v3 Certificate-Based SSH authentication 158
Configuring SSH-Cert-Authorization on TACACS servers 160
Configure legacy SSH algorithm SupportCustomizing SSH cryptographic algorithms 160
Algorithms supported - FIPS mode enabled 164
Change the default SSH server port 166
Clear SSH hosts 167
Disable the SSH server 167
Delete SSH server keys 168
Clear SSH sessions 169
Configuring telnet 170
Enable the telnet server 170
Starting telnet sessions to remote devices 171
Clear telnet sessions 171
SSH and telnet configuration verification commands 172
Configuration example for SSH 173
Configuration example for SSH passwordless file copy 174
Configuration example for x.509v3 Certificate-Based SSH authentication 175
Additional references for SSH and telnet 176

CHAPTER 9 **Configuring PKI** 179
About PKI 179
CAs and digital certificates 179
Trust model, trust points, and identity CAs 180
CA Certificate Hierarchy 181
Importing CA Bundle 181
Import the CA Certificate Bundle in PKCS7 Format 182
RSA key pairs and identity certificates 183
Multiple trusted CA support 184

PKI enrollment support 185
Manual enrollment using Cut-and-Paste 185
Multiple RSA key pair and identity CA support 186
Peer certificate verification 186
Certificate revocation checking 187
CRL support 187
Ndcpp: OCSP for syslog 187
Import and export support for certificates and associated key pairs 188
Guidelines and limitations for PKI 188
Default settings for PKI 189
Configuring CAs and digital certificates 190
Configure the hostname and IP domain name 190
Generate an RSA key pair 191
Generate an ECC key pair 192
Create a trust point CA association 193
Configure certificate mapping filters 195
Authenticate the CA 197
Configure certificate revocation checking methods 199
Generate a certificate request 200
Install an identity certificate 201
Ensuring trust point configurations persist across reboots 203
Export identity information in PKCS 12 format 204
Import identity information in PKCS 12 or PKCS 7 format 205
Configure a CRL 206
Delete certificates from the CA configuration 207
Delete RSA key pairs from a device 209
PKI configuration verification commands 210
Configuration examples for PKI 211
Configure certificates on a device 211
Download a CA certificate 213
Request an identity certificate 217
Revoke a certificate 223
Generate and publish the CRL 225
Download the CRL 227

Import the CRL 229
Additional references for PKI 231
Related documents for PKI 232
Standards for PKI 232

CHAPTER 10 Configuring User Accounts and RBAC 233
About user accounts and RBAC 233
User accounts 233
Characteristics of strong passwords 234
User roles 234
User role rules 235
Guidelines and limitations for user accounts and RBAC 236
Default settings for user accounts and RBAC 237
Enable Password-Strength checking 238
Enable consecutive characters check in passwords 238
Configure user accounts 239
Configuring roles 242
Configure user roles and rules 242
Create feature groups 243
Change user role interface policies 245
Change user role VLAN policies 246
Change user role VRF policies 247
About no service Password-Recovery 248
Enable no service Password-Recovery 248
User accounts and RBAC configuration verification commands 251
Configuration examples for user accounts and RBAC 251
Additional references for user accounts and RBAC 253

CHAPTER 11 Configuring 802.1X 255
About 802.1x 255
Device roles 255
Authentication initiation and message exchange 257
Authenticator PAE status for interfaces 258
Ports in authorized and unauthorized states 259

MAC authentication bypass 260
Dynamic VLAN assignment based on MAC-Based authentication (MAB) 261
VLAN assignment from RADIUS 262
Single host and multiple hosts support 262
Supported topology 263
About Per-User DACLs 263
Critical authentication 264
About DACL 264
Prerequisites for 802.1x 265
802.1X guidelines and limitations 265
Guidelines and limitations for Per-User DACL support for 802.1x 269
Guidelines and limitations for critical authentication 270
Default settings for 802.1x 270
Configuring 802.1x 271
Configure the 802.1x feature 272
Enable the 802.1x feature 272
Configure AAA authentication methods for 802.1x 273
Control 802.1x authentication on an interface 274
Configure the EAP-TLS profile for 802.1x authentication 275
Create or remove an authenticator PAE on an interface 276
Enable critical authentication 278
Enable periodic reauthentication for an interface 279
Manually reauthenticate supplicants 281
Change 802.1x authentication timers for an interface 281
Enable MAC authentication bypass 284
Configure the default 802.1x authentication method using MAB 285
Create a dynamic access list 286
Configure Per-User DACLs 287
Enable single host or multiple hosts mode 288
Disable 802.1x authentication on the Cisco NX-OS device 289
Disable the 802.1x feature 290
Reset the 802.1x interface configuration to the default values 291
Set the maximum Authenticator-to-Supplicant frame for an interface 292
Enable RADIUS accounting for 802.1x authentication 293

Configure AAA accounting methods for 802.1x 294
Set the maximum reauthentication retry count on an interface 295
802.1X configuration verification commands 296
802.1X support for VXLAN EVPN 298
Guidelines and limitations for 802.1x support for VXLAN EVPN 298
Configure 802.1x support for VXLAN EVPN 299
802.1X support for VXLAN EVPN verification commands 300
Critical authentication verification 302
Monitor 802.1x statistics 303
802.1X port configuration commands 303
Configuration example for Per-User DACL 305
Additional references for 802.1x 306

CHAPTER 12 **Configure IP ACLs** 307
About ACLs 307
ACL types and applications 307
Order of ACL application 309
About rules 310
Protocols for IP ACLs and MAC ACLs 310
Source and destination 311
Implicit rules for IP and MAC ACLs 311
Additional filtering options 312
Sequence numbers 313
Logical operators and logical operation units 314
ACL logging 314
Time ranges 315
Policy-based ACLs 316
Kernel Stack ACL 317
Statistics and ACLs 318
Atomic ACL updates 318
Session manager support for IP ACLs 319
ACL TCAM regions 319
Maximum label sizes supported for ACL types 325
Prerequisites for IP ACLs 326

Guidelines and limitations for IP ACLs 326
Default settings for IP ACLs 337
Configuring IP ACLs 337
Create an IP ACL 337
Modify an IP ACL 339
Create a VTY ACL 341
Change sequence numbers in an IP ACL 342
Remove an IP ACL 343
Configure ACL TCAM region sizes 344
Configure ACL TCAM region sizes using templates 351
Configure TCAM carving 353
Configure UDF-Based port ACLs 360
Applying an IP ACL as a router ACL 362
Applying an IP ACL as a port ACL 364
Applying an IP ACL as a VACL 365
Apply an IP ACL rule prioritization over SUP rule 365
Configure IPv4 ACL logging 367
Configure ACLs using HTTP methods to redirect requests 370
Configure an ACL for IPv6 extension headers 372
IP ACL configuration verification commands 373
Monitoring and clearing IP ACL statistics 375
Configuration examples for IP ACLs 376
About system ACLs 377
Carving a TCAM region 378
Configure system ACLs 379
Configuration and show command examples for the system ACLs 379
Configuring object groups 381
Session manager support for object groups 381
Create and change an IPv4 address object group 381
Create and change an IPv6 address object group 383
Create and change a protocol port object group 384
Remove an object group 385
Object-group configuration verification commands 386
Configuring Time-Ranges 386

Session manager support for Time-Ranges 386
Create a Time-Range 386
Change a Time-Range 388
Remove a Time-Range 389
Change sequence numbers in a time range 390
Time-range configuration verification commands 391
Additional references for IP ACLs 391

CHAPTER 13 Configuring MAC ACLs 393
MAC ACLs 393
MAC packet classification 393
Guidelines for configuring MAC ACLs 394
Default settings for MAC ACLs 395
Configuring MAC ACLs 395
Create a MAC ACL 395
Configure a UDF-Based MAC ACL 396
Modify a MAC ACL 398
Change sequence numbers in a MAC ACL 400
Remove a MAC ACL 400
Apply a MAC ACL as a port ACL 401
Apply a MAC ACL as a VACL 402
Enable or disable MAC packet classification 402
Prioritize a MAC ACL rule over a SUP rule 404
MAC ACL configuration verification commands 405
MAC ACL statistics monitoring and clearing 405
Examples of MAC ACL configuration 406
Additional references for MAC ACLs 406

CHAPTER 14 Configuring VLAN ACLs 407
VLAN ACLs 407
VLAN access maps and entries 407
Actions in VACLs 407
VACL statistics 408
VACLs in session manager 408

Prerequisite requirements for VACLs 408
Best practices and limitations for VACLs 408
Default settings for VACLs 409
Configuring VACLs 410
Create a VACL entry 410
Remove a VACL or VACL entry 411
Apply a VACL to your VLAN 412
VACL configuration verification commands 413
Commands for monitor and clear VACL statistics 414
Configuration example for VACLs 414
Additional references for VACLs 414

CHAPTER 15 Configuring Port Security 415
About port security 415
Secure MAC address learning 416
Static method 416
Dynamic method 416
Sticky method 416
Dynamic address aging 417
Secure MAC address maximums 417
Security violations and actions 418
Port security and port types 419
Port security and Port-Channel interfaces 419
Port type changes 421
Prerequisites for port security 421
Default settings for port security 421
Guidelines and limitations for port security 422
Guidelines and limitations for port security on vPCs 423
Configuring port security 424
Enable or disable port security globally 424
Enable or disable port security on a layer 2 interface 424
Enable or disable sticky MAC address learning 426
Add a static secure MAC address on an interface 427
Remove a static secure MAC address on an interface 428

Remove a sticky secure MAC address 429
Remove a dynamic secure MAC address 430
Configure a maximum number of MAC addresses 431
Configure an address aging type and time 432
Configure a security violation action 433
Port security configuration verification commands 434
Displaying secure MAC addresses 435
Configuration example for port security 435
Configuration examples for port security in a vPC domain 435
Example: configuring port security on an orphan port 435
Example: configuring port security on the vPC leg 436
Additional references for port security 437
Port security support for VXLAN EVPN 437
Guidelines and limitations for port security support for VXLAN EVPN 437
Port security support for VXLAN EVPN verification commands 438

CHAPTER 16 **Configuring DHCP** 441
About DHCP snooping 441
Trusted and untrusted sources 442
DHCP snooping binding database 442
DHCP snooping in a vPC environment 443
Synchronizing DHCP snooping binding entries 443
Packet validation 443
DHCP snooping option 82 data insertion 444
About the DHCP Relay Agent 447
DHCP relay agent 447
DHCP relay agent option 82 447
VRF support for the DHCP relay agent 449
DHCP smart relay agent 450
About the DHCPv6 Relay Agent 451
Dhcpv6 relay agent 451
VRF support for the DHCPv6 relay agent 451
IPv6 availability for delegated prefix through the v6 relay agent 452
DHCPv6 Smart Relay Agent 452

Guidelines and Limitations for DHCPv6 Smart Relay 453
About DHCP client 453
Prerequisites for DHCP 453
Guidelines and limitations for DHCP 453
Default settings for DHCP 455
Configuring DHCP 456
Configure the minimum DHCP settings 456
Enable or disable the DHCP feature 457
Configuring DHCP snooping 458
Enable or disable DHCP snooping globally 458
Enable or disable DHCP snooping on a VLAN 458
Enable or disable DHCP snooping MAC address verification 459
Enable or disable option 82 data insertion and removal 460
Enable or disable strict DHCP packet validation 462
Configure an interface as trusted or untrusted 463
Enable or disable the DHCP relay trusted port functionality 464
Configure an interface as a DHCP relay trusted or untrusted port 465
Configure all interfaces as trusted or untrusted 467
Enable or disable the DHCP relay agent 468
Enable or disable option 82 for the DHCP relay agent 468
Enable or disable VRF support for the DHCP relay agent 470
Disable the server identifier override option 471
Configure DHCP server addresses on an interface 472
Configure the DHCP relay source interface 473
Enable or disable DHCP smart relay globally 474
Enable or disable DHCP smart relay on a layer 3 interface 475
Configure the DHCP relay Subnet-Selection 477
Configuring DHCPv6 478
Enable or disable the DHCPv6 relay agent 478
Enable or disable VRF support for the DHCPv6 relay agent 479
Enable or Disable DHCPv6 Smart Relay Globally 480
Enable or Disable DHCPv6 Smart Relay on a Layer 3 Interface 481
Configure DHCPv6 server addresses on an interface 482
Enable DHCPv6 option 79 484

Configure the DHCPv6 relay source interface 484
Configure the IPv6 RA guard feature 485
Enable the DHCP client 486
Configuring UDP relay 488
About UDP relay 488
Guidelines and limitations for UDP relay 489
Configure the UDP relay feature 490
Configuration example for UDP relay 491
Verifying the UDP Relay Configuration 492
DHCP configuration verification commands 492
Displaying IPv6 RA guard statistics 494
Displaying DHCP snooping bindings 494
Clearing the DHCP snooping binding database 495
Monitoring DHCP 495
Clearing DHCP snooping statistics 496
Clearing DHCP relay statistics 496
Clearing DHCPv6 relay statistics 496
Clearing DHCPv6-PD binding 497
Configuration examples for DHCP 497
Configuration examples for DHCP client 498
Additional references for DHCP 499

CHAPTER 17 **Configuring IPv6 First Hop Security** 501
About First-Hop security 501
IPv6 global policies 502
IPv6 First-Hop security binding table 502
Guidelines and limitations of First-Hop security 502
About vPC First-Hop security configuration 503
DHCP relay on-stack 504
DHCP relay on vPC leg 505
DHCP client relay on orphan ports 506
RA guard 508
Overview of IPv6 RA guard 508
IPv6 RA router advertisement and the flags 508

Guidelines and limitations of IPv6 RA guard 509
Dhcpv6 guard 510
Overview of DHCP—DHCPv6 guard 510
Limitation of DHCPv6 guard 511
IPv6 snooping 511
Overview of IPv6 snooping 511
Guidelines and limitations for IPv6 snooping 512
How to configure IPv6 FHS 512
Configure the IPv6 RA guard policy on the device 512
Configure IPv6 RA guard on an interface 514
Configure DHCPv6 guard 514
Configure an IPv6 snooping policy 516
Verify and troubleshoot IPv6 snooping 518
Configuration examples 519
Example: IPv6 RA guard configuration 519
Example: configuring DHCPv6 guard 519
Example: configuring IPv6 First-Hop security binding table 519
Example: configuring IPv6 snooping 520
Additional references for IPv6 First-Hop security 520

CHAPTER 18 **Configuring Dynamic ARP Inspection** 521
About DAI 521
ARP 521
ARP spoofing attacks 522
DAI and ARP spoofing attacks 522
Interface trust states and network security 523
Logging DAI packets 524
DHCP relay with dynamic ARP inspection 524
Prerequisites for DAI 525
Guidelines and limitations for DAI 525
Guidelines and limitations for DHCP relay with DAI 526
Default settings for DAI 526
Configuring DAI 527
Enable or disable DAI on VLANs 527

Configure the DAI trust state of a layer 2 interface 528
Enable or disable additional validation 529
Configure the DAI logging buffer size 530
Configure DAI log filtering 530
Enable DHCP relay with DAI 531
DAI configuration verification commands 532
Monitoring and clearing DAI statistics 532
Configuration examples for DAI 532
Two devices support DAI 532
Configure device a 533
Configure device b 535
Examples for DHCP relay with DAI 537
Additional references for DAI 537
Related documents 537
Standards 537

CHAPTER 19 **Configuring IP Source Guard** 539
IP source guard overview 539
Prerequisites for IP source guard 540
Recommendation for IP source guard configuration 540
Default settings for IP source guard 541
Configuring IP source guard 541
Enable or disable IP source guard on a layer 2 interface 541
Add or remove a static IP source entry 542
Configure IP source guard on trunk ports 543
Displaying IP source guard bindings 544
Clearing IP source guard statistics 544
IP source guard configuration example 544
Additional references 545
Reference documents for ACL TCAM regions and DHCP snooping 545

CHAPTER 20 **Configuring Password Encryption** 547
AES password encryption and primary encryption keys 547
Guidelines and limitations for password encryption 548

Password encryption default settings 550
Configuring password encryption 550
Configure a primary key and enable the AES password encryption feature 550
Convert existing passwords to Type-6 encrypted passwords 551
Recover original passwords from Type-6 encrypted values 552
Enable Type-6 encryption on MACsec keys 552
Delete Type-6 encrypted passwords 554
Password encryption configuration verification 554
Configuration examples for password encryption 554

CHAPTER 21 Configuring Keychain Management 557
About keychain management 557
Prerequisites for keychain management 558
Guidelines and limitations for keychain management 558
Default settings for keychain management 559
Configuring keychain management 559
Create a keychain 559
Remove a keychain 560
Configure a primary key and enable the AES password encryption feature 561
Configure text for a key 562
Configure accept and send lifetimes for a key 564
Configure a key for OSPFv2 cryptographic authentication 566
Determining active key lifetimes 567
Keychain management configuration verification commands 567
Configuration example for keychain management 567
Where to go next 568
Additional references for keychain management 568

CHAPTER 22 Configuring Traffic Storm Control 569
About traffic storm control 569
Licensing requirements for traffic storm control 571
Guidelines and limitations for traffic storm control 571
Default settings for traffic storm control 575
Configure traffic storm control for one-level threshold 575

Prioritizing storm-control policer over the CoPP policer 577
Configure traffic storm control for two-level threshold 577
Traffic storm control configuration verification commands 579
Monitoring traffic storm control counters 579
Configuration examples for traffic storm control 580
System log examples for traffic storm control 581
Additional references for traffic storm control 581

CHAPTER 23 Configuring Unicast RPF 583
About unicast RPF 583
How unicast RPF works 584
Guidelines and limitations for unicast RPF 585
Unicast RPF parameter settings 587
Configure unicast RPF for Cisco Nexus 9500 switches with -R line cards 588
Configure unicast RPF for Cisco Nexus 9300 switches 589
Configuration examples for unicast RPF 591
Unicast RPF configuration verification 592
Reference materials for unicast RPF features 592

CHAPTER 24 Configuring Switchport Blocking 595
About switchport blocking 595
Guidelines and limitations for switchport blocking 595
Default settings for switchport blocking 596
Configure switchport blocking 596
Switchport blocking configuration verification commands 597
Configuration example for switchport blocking 597

CHAPTER 25 Configure Control Plane Policing 599
About CoPP 599
Control plane protection 600
Control plan packet types 600
Classification for CoPP 601
Egress CoPP 601
Rate controlling mechanisms 601

Dynamic and static CoPP ACLs 602
Default policing policies 603
Modular QoS Command-Line interface 616
Copp and the management interface 617
CoPP configuration guidelines 617
Default settings for CoPP 621
Configuring CoPP 622
Configure a control plane class map 622
Configure a control plane policy map 623
Configure the control plane service policy 625
Configure the CoPP scale factor per line card 627
Change or reapply the default CoPP policy 628
Copying the CoPP best practice policy 628
Protocol ACL filtering added for egress CoPP 629
Configure ARP ACL filtering for egress CoPP 629
Configure IP ACL filtering for egress CoPP 630
Copp configuration verification commands 632
Display the CoPP configuration status 634
Monitor CoPP 634
Monitoring CoPP with SNMP 634
Clear the CoPP statistics 635
Configuration examples for CoPP 635
Copp configuration example 636
Changing or reapplying the default CoPP policy using the setup utility 637
Changing CoPP policy limit 638
Additional references for CoPP 638

CHAPTER 26 **Configuring Rate Limits** 639
About rate limits 639
Guidelines and limitations for rate limits 640
Default settings for rate limits 641
Configure rate limits 641
Monitor rate limits 643
Clear the rate limit statistics 643

Rate limit configuration verification commands 644
Configuration examples for rate limits 644
Additional references for rate limits 645

CHAPTER 27 Configure MACsec 647
About MACsec 647
Key Lifetime and Hitless Key Rollover 648
Fallback Key 648
Licensing requirements for MACsec 648
Guidelines and limitations for MACsec 648
Enable MACsec 655
Disable MACsec 656
Configure a MACsec keychain and keys 657
MACsec packet-number exhaustion 659
Configure MACsec fallback key 659
Configure a MACsec policy 660
Configure MACsec EAP 662
QKD integration with SKIP on MACsec 663
About QKD integration with secure key integration protocol 663
Guidelines and limitations 663
Configuring point-to-point MACsec link encryption using SKIP 665
Enable postquantum cryptography 665
Enable MACsec and MKA features 666
Configure quantum key distribution profile 667
Configure a MACsec policy and assign a PPK crypto-QKD profile 667
Configuration examples 668
About Configurable EAPOL Destination and Ethernet Type 671
Enable EAPOL configuration 671
Disable EAPOL configuration 673
MACsec configuration verification commands 673
Display MACsec Statistics 676
Configuration Example for MACsec 679
XML output for MACsec show commands 683
Macsec supported MIBs and download locations 690

Related documentation 690

CHAPTER 28 **Configure TCP authentication option 691**

About TCP authentication option 691

TCP-AO key chain 691

TCP-AO key rollover 693

Guidelines and limitations 694

Configure TCP key chain and keys 694

TCP keychain verification commands 697

Configuration example for a TCP keychain 698



Preface

This preface includes the following sections:

- Audience, on page xxix
- Document Conventions, on page xxix
- Related Documentation for Cisco Nexus 9000 Series Switches, on page xxx
- Documentation Feedback, on page xxx
- Communications, services, and additional information, on page xxx

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information that you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

https://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

Communications, services, and additional information

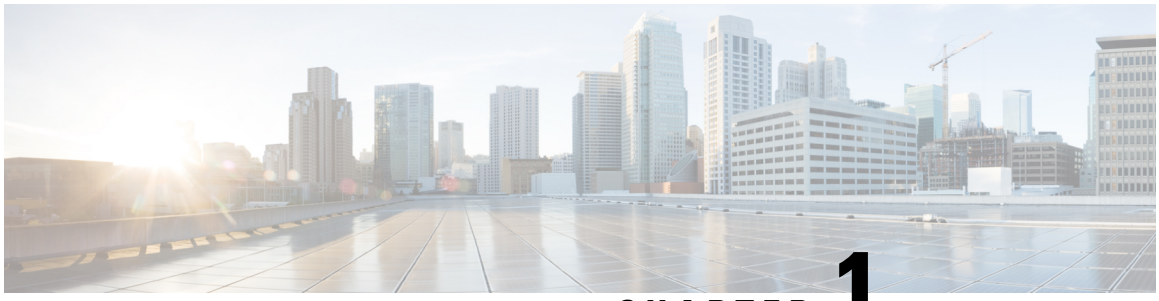
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.



CHAPTER 1

New and Changed Information

This chapter includes the new and changed features for the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 10.4(x).

- New and Changed Information, on page 1

New and Changed Information

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Class E in NX-OS Fabric	Added support for Security Group ACL with ESG on Class E IP addresses.	10.4(3)F	Guidelines and limitations for IP ACLs, on page 326
MACsec	Added support for MACsec on Cisco Nexus 9364C-H1 switches.	10.4(3)F	Guidelines and limitations for MACsec, on page 648
Security ACL	Added support for Security ACL on the Cisco Nexus 9364C-H1 switches.	10.4(3)F	Guidelines and limitations for IP ACLs, on page 326
ACL/CoPP	Added support for ACL and CoPP on the Cisco Nexus 9364C-H1 switches.	10.4(3)F	Guidelines and limitations for IP ACLs, on page 326 CoPP configuration guidelines, on page 617
Support flexible TCAM templates/enhancement	Added support for flexible TCAM configuration on the Cisco Nexus 9364C-H1 switches.	10.4(3)F	Guidelines and limitations for IP ACLs, on page 326

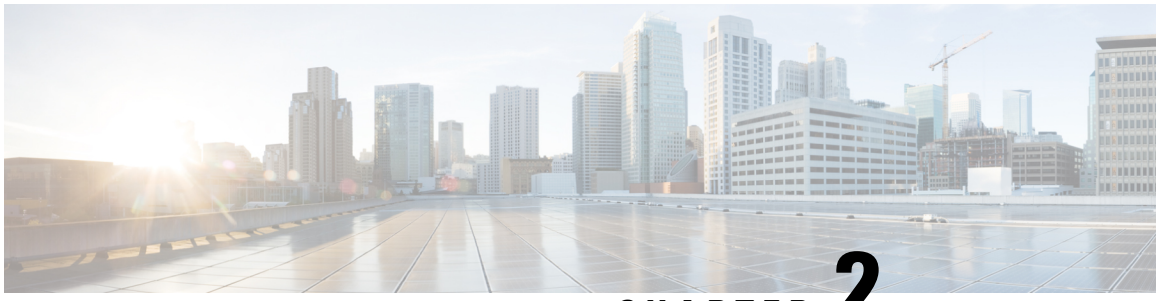
Feature	Description	Changed in Release	Where Documented
TLS v1.3	Added Transport Layer Security protocol version 1.3 support for Cisco Nexus applications.	10.4(3)F	TLS Protocol Support, on page 13 Guidelines and limitations for RadSec, on page 64 Guidelines and limitations for LDAP, on page 124 802.1X guidelines and limitations, on page 265
QKD (Quantum Key Distribution) integration with SKIP on MACsec	Added support for QKD on Cisco Nexus 9000 Series Switches.	10.4(3)F	QKD integration with SKIP on MACsec, on page 663
X.509 certificate based SSH Authorization using TACACS	Added support for SSH-based authorization of x509v3-certificates using TACACS+ server.	10.4(3)F	Guidelines and limitations for AAA, on page 29 Configure AAA SSH-Cert-Authorization on TACACS servers, on page 46 TACACS+, on page 91 Guidelines and limitations for TACACS+, on page 95 Configure X.509 Certificate-Based SSH authorization using TACACS server, on page 114
Default route	Added support for configuring default route for ACL QoS.	10.4(2)F	Guidelines and limitations for IP ACLs, on page 326 Configuration examples for IP ACLs, on page 376
Support flexible TCAM templates/enhancement	Added support for flexible TCAM configuration on the Cisco Nexus 93400LD-H1 switches.	10.4(2)F	Guidelines and limitations for IP ACLs, on page 326
Security ACL	Added support for Security ACL on the Cisco Nexus 93400LD-H1 switches.	10.4(2)F	Guidelines and limitations for IP ACLs, on page 326
MACsec	Added support for MACsec on Cisco Nexus 93400LD-H1, and 93108TC-FX3 switches.	10.4(2)F	Guidelines and limitations for MACsec, on page 648

Feature	Description	Changed in Release	Where Documented
ACL/CoPP	Added support for ACL and CoPP ACL on Cisco Nexus 93400LD-H1, and 93108TC-FX3 switches.	10.4(2)F	CoPP configuration guidelines, on page 617 Guidelines and limitations for IP ACLs, on page 326
CR multiline support	Added configuration replace feature support for LDAP.	10.4(2)F	Guidelines and limitations for LDAP, on page 124
BGP support for TCP Authentication Option (TCP-AO)	Added support for TCP authentication option on Cisco Nexus 9000 platform switches.	10.4(2)F	Configure TCP authentication option, on page 691
MACsec rate counter support on FX3 platforms	Added support for MACsec rate counter for "show macsec secy statistics command" and rate related OIDs in CISCO-SECY-EXT-MIB on Cisco Nexus 9300-FX3 switches.	10.4(2)F	Guidelines and limitations for MACsec, on page 648
Flexible configuration of SSH to customize Ciphers, MACs, and Keytypes	Added CLI options to configure SSH Algorithm.	10.4(2)F	Configure legacy SSH algorithm SupportCustomizing SSH cryptographic algorithms, on page 160
Certification based authentication for MACsec	Added EAP-TLS support to 802.1X Port-based Authentication for uplink ports where MACsec is required.	10.4(1)F	Prerequisites for 802.1x, on page 265 802.1X guidelines and limitations, on page 265 Configure the EAP-TLS profile for 802.1x authentication, on page 275 802.1X configuration verification commands, on page 296 Configure MACsec EAP, on page 662

Feature	Description	Changed in Release	Where Documented
Support to redirect/deny 'all' packets using ePBR policy	Added new ACE **all** keyword to prioritize the IP/IPv6/MAC ACL rule over SUP rule.	10.4(1)F	Guidelines and limitations for IP ACLs, on page 326 Apply an IP ACL rule prioritization over SUP rule, on page 365 Guidelines for configuring MAC ACLs, on page 394 Prioritize a MAC ACL rule over a SUP rule, on page 404
Radius over DTLS Support	Radius over DTLS protocol support is added. This protocol is for transporting RADIUS datagrams over a secure channel.	10.4(1)F	About RADIUS with DTLS, on page 72 Configure RADIUS with DTLS, on page 72
AAA	Added support for AAA on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	Guidelines and limitations for AAA, on page 29
RADIUS	Added support for RADIUS on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	Guidelines and limitations for RADIUS, on page 63
TACACS+	Added support for TACACS+ on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	Guidelines and limitations for TACACS+, on page 95
LDAP	Added support for LDAP on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	Guidelines and limitations for LDAP, on page 124
MACsec	Added support for MACsec on Cisco Nexus 9348GC-FX3, 9348GC-FX3PH and 9332D-H2R switches	10.4(1)F	Guidelines and limitations for MACsec, on page 648
Security ACL	Added support for Security ACL on the Cisco Nexus 9332D-H2R switches.	10.4(1)F	Guidelines and limitations for IP ACLs, on page 326

Feature	Description	Changed in Release	Where Documented
ACL/CoPP	Added support for ACL and CoPP on the Cisco Nexus 9332D-H2R switches.	10.4(1)F	Guidelines and limitations for IP ACLs, on page 326 CoPP configuration guidelines, on page 617
Support flexible TCAM templates/enhancement	Added support for flexible TCAM configuration on the Cisco Nexus 9332D-H2R switches.	10.4(1)F	Guidelines and limitations for IP ACLs, on page 326
ACL Consistency Checker	Added support for ACL Consistency Checker on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	Guidelines and limitations for IP ACLs, on page 326
ACL - Sup, CoPP	Added support for CoPP ACL on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	CoPP configuration guidelines, on page 617
RACL with statistics	Added support for RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	Guidelines and limitations for IP ACLs, on page 326
DHCP relay	Added support for DHCP relay on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	Guidelines and limitations for DHCP, on page 453
CoPP Limit	Added CoPP Limit configuration for PTP interface on Cisco Nexus 9348GC-FX3	10.4(1)F	Changing CoPP policy limit, on page 638

New and Changed Information



CHAPTER 2

Overview

The Cisco NX-OS software supports security features that can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

This chapter includes the following sections:

- Licensing Requirements, on page 7
- Supported Platforms, on page 8
- Authentication, authorization, and accounting, on page 8
- RADIUS and TACACS+ Security Protocols, on page 9
- LDAP, on page 9
- SSH and Telnet, on page 9
- User Accounts and Roles, on page 9
- IP ACLs, on page 10
- MAC ACLs, on page 10
- VACLs, on page 10
- DHCP Snooping, on page 10
- Dynamic ARP Inspection, on page 11
- IP Source Guard, on page 11
- Password Encryption, on page 11
- Keychain Management, on page 11
- Control Plane Policing, on page 12
- Rate Limits, on page 12
- Software Image, on page 12
- Virtual Device Contexts, on page 12
- SGT Tagged Packet, on page 12
- TLS Protocol Support, on page 13

Licensing Requirements

See the Cisco NX-OS Licensing Guide and Cisco NX-OS Licensing Options Guide for Cisco NX-OS licensing recommendations and instructions to obtain and apply licenses.

Supported Platforms

See the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

Authentication, authorization, and accounting

Authentication, authorization, and accounting (AAA) are security functions that

- establish user identity and access permissions
- manage user privileges and resource access, and
- track user activities for auditing and billing.

Authentication

Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.

Authorization

Remote security servers, such as RADIUS and TACACS+, use attribute-value (AV) pairs to associate specific rights with users, thereby authorizing them. They provide the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.

Accounting

Accounting provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. It allows tracking of user services and the amount of network resources being consumed.

Note You can configure authentication outside of AAA. However, you must configure AAA if you want to use RADIUS or TACACS+, or if you want to configure a backup authentication method.

For more information, see the Configuring AAA, on page 23 chapter.

RADIUS and TACACS+ Security Protocols

AAA uses security protocols to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS or TACACS+ security server.

The chapters in this guide describe how to configure the following security server protocols:

RADIUS

A distributed client/server system implemented through AAA that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

TACACS+

A security application implemented through AAA that provides a centralized validation of users who are attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.

For more information, see the Configuring TACACS+, on page 91 chapter and the Configuring RADIUS, on page 59 chapter.

LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a Cisco NX-OS device. LDAP allows a single access control server (the LDAP daemon) to provide authentication and authorization independently.

For more information, see the Configuring LDAP, on page 121 chapter.

SSH and Telnet

You can use the Secure Shell (SSH) server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device. SSH uses strong encryption for authentication. The SSH server and client in the Cisco NX-OS software can interoperate with publicly and commercially available SSH clients.

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

For more information, see the Configuring SSH and Telnet, on page 145 chapter.

User Accounts and Roles

You can create and manage user accounts and assign roles that limit access to operations on the Cisco NX-OS device. Role-based access control (RBAC) lets you define rules to assign roles, which restrict the user's authorization to access management operations.

For more information, see the Configuring User Accounts and RBAC, on page 233 chapter.

IP ACLs

IP ACLs are ordered sets of rules that you can use to filter traffic based on IPv4 information in the Layer 3 header of packets. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Cisco NX-OS software determines that an IP ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether a packet is permitted or denied, or if there is no match, the Cisco NX-OS software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.

For more information, see the Configure IP ACLs, on page 307 chapter.

MAC ACLs

MAC ACLs filter traffic using information in the Layer 2 header of each packet. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Cisco NX-OS software determines that a MAC ACL applies to a packet, it tests the packet against all rules. If the packet matches a rule, the software permits or denies the packet according to the rule. If there is no match, the software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.

VACLs

A VLAN ACL (VACL) is one application of an IP ACL or MAC ACL. You can configure VACLs to apply to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VACLs are strictly for security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

For more information, see the Configuring VLAN ACLs, on page 407 chapter.

DHCP Snooping

DHCP snooping provides firewall-like protection between untrusted hosts and trusted DHCP servers. DHCP snooping performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Builds and maintains the DHCP snooping binding database. This database contains information about untrusted hosts with leased IP addresses.
- Uses the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

Dynamic ARP inspection (DAI) and IP Source Guard (IPSG) also use information stored in the DHCP snooping binding database.

Dynamic ARP Inspection

Dynamic ARP inspection (DAI) ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, a Cisco NX-OS device performs these activities:

- Intercepts all ARP requests and responses on untrusted ports.
- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination.
- Drops invalid ARP packets.

DAI can determine the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a DHCP snooping binding database. This database can also contain static entries that you create. If the ARP packet is received on a trusted interface, the device forwards the packet without any checks. On untrusted interfaces, the device forwards the packet only if it is valid.

IP Source Guard

IP Source Guard is a per-interface traffic filter. It permits IP traffic only when the IP address and MAC address of each packet match one of two sources of IP and MAC address bindings:

- Entries in the DHCP snooping binding table.
- Static IP source entries that you configure.

Filtering on trusted IP and MAC address bindings helps prevent attacks that rely on IP address spoofing. To circumvent IP Source Guard, an attacker would have to spoof both the IP address and the MAC address of a valid host.

Password Encryption

The Advanced Encryption Standard (AES) password encryption feature stores all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) in the strong and reversible type-6 encrypted format. A primary encryption key is used to encrypt and decrypt the passwords. You can also use this feature to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

For more information, see the Configuring Password Encryption, on page 547 chapter.

Keychain Management

Keychain management allows you to create and maintain keychains, which are sequences of keys (sometimes called shared secrets). You can use keychains with features that secure communications with other devices by using key-based authentication. The device allows you to configure multiple keychains.

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication.

For more information, see the Configuring Keychain Management, on page 557 chapter.

Control Plane Policing

The Cisco NX-OS device provides control plane policing to prevent denial-of-service (DoS) attacks from impacting performance. The supervisor module of the Cisco NX-OS device has both the management plane and control plane and is critical to the operation of the network. Any disruption to the supervisor module would result in serious network outages. Excessive traffic to the supervisor module could overload it and slow down the performance of the entire Cisco NX-OS device. Attacks on the supervisor module can include different types, such as denial-of-service (DoS) attacks that generate IP traffic streams to the control plane at a very high rate. These attacks cause the control plane to spend significant time handling these packets, making it unable to process genuine traffic.

For more information, see the Configure Control Plane Policing, on page 599 chapter.

Rate Limits

Rate limits can prevent redirected packets for egress exceptions from overwhelming the supervisor module on a Cisco NX-OS device.

For more information, see the Configuring Rate Limits, on page 639 chapter.

Software Image

The Cisco NX-OS software consists of one NXOS software image.

Virtual Device Contexts

Cisco NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Cisco Nexus 9000 Series switches currently do not support multiple VDCs. All switch resources are managed in the default VDC.

SGT Tagged Packet

Beginning with Cisco NX-OS Release 10.2(2)F, the Cisco Nexus 9300- EX/ FX/FX2/FX3/GX platform switches can be used as a L2 or L3 transit device to forward a SGT tagged packet. Cisco NX-OS N9K can pass the packet without stripping or altering the SGT.

Beginning with Cisco NX-OS Release 10.3(3)F, SGT forwarding is supported on Cisco Nexus 9500 switches with N9K-X97160YC-EX line card.

Note NX-OS N9K does not do any SGT enforcement. It also does not generate or rewrite SGT/DGT info in the packets. Packets are received and transmitted out with the same tags.

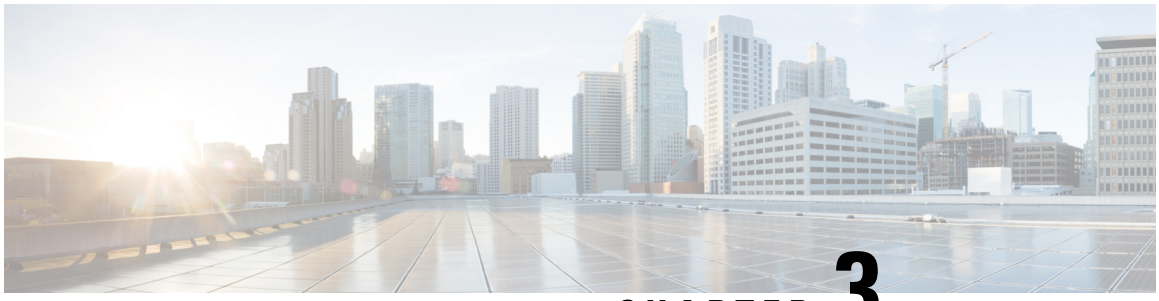
TLS Protocol Support

The Transport Layer Security (TLS) protocol is a cryptographic protocol designed to provide secure communication over a computer network.

Beginning with Cisco NX-OS Release 10.4(3)F, Cisco Nexus application supports Transport Layer Security (TLS) version 1.3 by default.

The following applications supports TLSv1.3 for faster and secure communication:

- **Programmability and Management Interfaces:**
  - NX-API: For more details, see NX-API Management Commands .
  - gNMI: For more details, see gNMI - Management Interface .
  - gRPC agent: For more details, see gRPC Agent .
- **System Operations and Monitoring**
  - Secure syslog: For more details, see Configuring System Message Logging.
  - Callhome: For more details, see Configuring Smart Call Home.
  - Smart Licensing: For more details, see Smart Licensing Using Policy.
  - Copy utilities (HTTPS option): For more details, see Using the Device File Systems, Directories, and Files.
- **Security and Authentication**
  - RadSec: For more details, see Guidelines and limitations for RadSec, on page 64.
  - 802.1x: For more details, see 802.1X guidelines and limitations, on page 265.
  - LDAP: For more details, see Guidelines and limitations for LDAP, on page 124.



CHAPTER 3

Configuring FIPS

This chapter describes how to configure the Federal Information Processing Standards (FIPS) mode on Cisco NX-OS devices.

This chapter includes the following sections:

- About FIPS, on page 15
- Prerequisites for FIPS, on page 16
- Guidelines and limitations for FIPS, on page 17
- Default settings for FIPS, on page 17
- Configuring FIPS, on page 17
- FIPS configuration verification commands, on page 19
- Create 2048 bit RSA key, on page 19
- Configuration example for FIPS, on page 20
- Additional references for FIPS, on page 20

About FIPS

The FIPS 140–2 Publication, *Security Requirements for Cryptographic Modules*, describes U.S. government requirements for cryptographic modules. According to FIPS 140–2, a cryptographic module is a set of hardware, software, firmware, or a combination of these components that implements cryptographic functions and is contained within a defined boundary. These modules provide cryptographic algorithms and, optionally, key generation.

FIPS specifies certain cryptographic algorithms as secure, and it identifies which algorithms should be used if a cryptographic module is to be called FIPS compliant.

FIPS Self-Tests

A cryptographic module must perform power-up self-tests and conditional self-tests to ensure that it is functioning properly.

Power-up self-tests run automatically after the device powers up. A device goes into FIPS mode only after all self-tests are successfully completed. If any self-test fails, the device logs a system message and moves into an error state.

Generate the "crypto key generate rsa label test exportable modulus" with a minimum modulus size of 2048. A key size less than 2048 is categorized as a “FIPS Self-Test” or results in a "FIPS Error State".

The device uses a cryptographic algorithm known-answer test (KAT) for each FIPS 140- 3 -approved cryptographic function (encryption, decryption, authentication, and random number generation) implemented on the device. The device applies the algorithm to data for which the correct output is already known, and then compares the calculated output to this known answer. If the calculated output does not match the known answer, the KAT fails.

Conditional self-tests include these tests:

- Pair-wise consistency test: This test runs when a public or private key-pair is generated.
- Continuous random number generator test: This test runs when a random number is generated.

  The Cisco TrustSec manager also runs a bypass test to ensure that encrypted text is never sent as plain text.



Note A bypass test failure on CTS-enabled ports causes only those corresponding ports to be shut down. The bypass test might fail because of packet drops caused by data path congestion. In such cases, we recommend that you try bringing up the port again.

FIPS error state

When the system is booted up in FIPS mode, the FIPS power-up self-tests run on the supervisor and line card modules. If any of these bootup tests fail, the whole system is moved to the FIPS error state. In this state, the system deletes all cryptographic keys and shuts down all line cards as required by FIPS. This mode is exclusively meant for debugging purposes.

After the switch is in the FIPS error state, any reload of a line card moves it to the failure state. To move the switch back to FIPS mode, it has to be rebooted. However, once the switch is in FIPS mode, a power-up self-test failure during a line card reload or insertion affects only that line card, and only the corresponding line card is moved to the failure state.

Prerequisites for FIPS

FIPS has the following prerequisites:

- Disable Telnet. Users should log in using Secure Shell (SSH) only.
- Any existing user accounts on the device that have been configured for SNMPv3 should be configured only with SHA for authentication and AES/3DES for privacy.
- Delete all SSH server RSA1 key-pairs.
- Enable HMAC-SHA1 message integrity checking (MIC) for Cisco TrustSec Security Association Protocol (SAP) negotiation. Enter the **sap hash-algorithm HMAC-SHA-1** command in `cts-manual` or `cts-dot1x` mode.

Guidelines and limitations for FIPS

Use this reference to review the guidelines and limitations for FIPS.

FIPS has these configuration guidelines and limitations:

- The user authentication mechanisms supported for SSH are usernames and passwords, public keys, and X.509 certificates.
- Passwords must consist of at least eight alphanumeric characters.
- You must disable Radius and TACACS when FIPS mode is on, because OpenSSL enforces this requirement in FIPS mode.
- When RadSec is enabled, Radius doesn't need to be disabled when FIPS is enabled.
- When the **fips mode enable** command is executed after an ASCII reload, you need to reload the Cisco NX-OS switch after executing the **copy running-config startup-config** command.

Default settings for FIPS

Use this reference to review the default settings for FIPS.

This table lists the default settings for FIPS parameters.

Table 2: Default FIPS parameters

Parameters	Default
FIPS mode	Disabled

Configuring FIPS

This section describes how to configure FIPS mode on Cisco NX-OS devices.

Enable FIPS mode

Before you begin

Ensure that you are in the default VDC.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Enable FIPS mode.

Example:

```
switch(config)# fips mode enable
```

Note
fips mode enable can be entered only when all LCs are online or else it leads to LC failure.

Step 3 Exit configuration mode.

Example:

```
switch(config)# exit
switch#
```

Step 4 (Optional) Display the status of FIPS mode.

Example:

```
switch# show fips status
FIPS mode is enabled
```

Step 5 Required: Copy the running configuration to the startup configuration.

Example:

```
switch# copy running-config startup-config
```

Step 6 Required: Reload the Cisco NX-OS device.

Example:

```
switch# reload
```

Note
After you enable FIPS, a reboot is required for the system to operate in FIPS mode.

Disable FIPS

You can disable FIPS mode on the device using the CLI.

Before you begin

Ensure that you are in the default VDC.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Disable FIPS mode.

Example:

```
switch(config)# no fips mode enable
```

Step 3 Exit the configuration mode.

Example:

```
switch(config)# exit
switch#
```

Step 4 (Optional) Display the status of FIPS mode.

Example:

```
switch# show fips status
FIPS mode is disabled
```

Step 5 Copy the running configuration to the startup configuration.

Example:

```
switch# copy running-config startup-config
```

Step 6 Reload the Cisco NX-OS device.

Example:

```
switch# reload
```

FIPS configuration verification commands

Use this reference to verify the FIPS configuration.

To display FIPS configuration information, perform one of the following tasks:

Table 3: Table 1

Command	Purpose
show fips status	Displays the status of the FIPS feature.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 9000 Series NX-OS Security Command Reference* .

Create 2048 bit RSA key

Steps to create a 2048 bit RSA key:

- Enter global configuration mode.

```
N9k-Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

- Disable the SSH feature to allow removal or regeneration of SSH keys safely.

```
N9k-Switch(config)#no feature ssh
XML interface to system may become unavailable since ssh is disabled
```

- Remove the existing RSA SSH key from the device configuration and generate a new RSA SSH key with a 2048-bit length to enhance security.

```
N9k-Switch(config)#no ssh key rsa
N9k-Switch(config)#ssh key rsa 2048
New SSH Key has a bitcount of 2048:
```

- Verify the configured SSH keys and their details.

```
N9k-Switch#show ssh key
**************************************
rsa Keys generated:Wed Apr 28 13:05:18 2021
ssh-rsa  AAAAB3NzaC1yc2EAAAADAQABAAABAQDHpxEgZ9LwmbOEpJeJtLwqedmTLkZV7Setxb9D4xgO
rsa Keys generated:Wed Apr 28 13:05:18 2021
ssh-rsa  AAAAB3NzaC1yc2EAAAADAQABAAABAQDHpxEgZ9LwmbOEpJeJtLwqedmTLkZV7Setxb9D4xgO
p2o2f6wt/48bPp/vLDGsxTF2PtLRtRSSDFNSQmkw9bg+MXvTpgNivdxWLjxtwo3YpYwPkBiReVmyrFgE
UuBmV/sDfhJpHXLoH9lR2+y0L5w1OG3cJxMe30TI37O3M8fZPjrAtHgkUubfEpiTbcyEw+aIHf+chyoR
eDJxcEdnlboiTDFR0/+jMUUM/vMtxd5x5DH3AO7htA/i8lvskrReR1CpX1sOOdcshmS57EEuEzR9cs+w
KSftQh6vLD802207T6+J7/+cXMVNQEbq0mCSzeTmOsuIQe8u9ZC24pgYzZ19
bitcount:2048
fingerprint:
SHA256:Am9861AIq5MzfSPQr4ZXGe0f5M9crnhk7HVZBXhMVBo
**************************************
could not retrieve dsa key information
**************************************
could not retrieve ecdsa key information
**************************************
```

Configuration example for FIPS

The following example shows how to enable FIPS mode:

```
config terminal
fips mode enable
show fips status
exit
copy running-config startup-config
reload
```

Additional references for FIPS

This section includes additional information related to implementing FIPS.

Related Documents

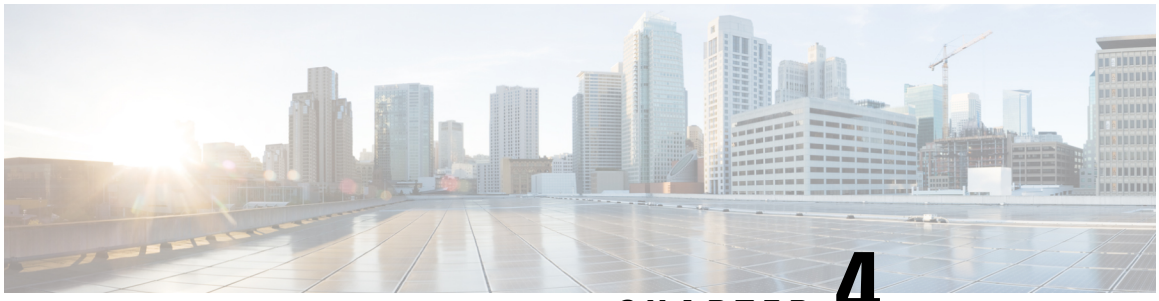
Table 4: Table 1

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 9000 Series NX-OS Security Command Reference*

Standards

Table 5: Table 3

Standards	Title
FIPS 140-2	Security Requirements for Cryptographic Modules



CHAPTER 4

Configuring AAA

This chapter describes how to configure authentication, authorization, and accounting (AAA) on Cisco NX-OS devices.

This chapter includes the following sections:

- About AAA, on page 23
- Prerequisites for AAA, on page 29
- Guidelines and limitations for AAA, on page 29
- Default settings for AAA, on page 30
- Configuring AAA, on page 31
- Monitor and clear the local AAA accounting log, on page 54
- AAA configuration verification commands, on page 55
- Configuration examples for AAA, on page 56
- Configuration examples for login parameters, on page 56
- Configuration examples for the password prompt feature, on page 57
- Additional references for AAA, on page 58

About AAA

This section includes information about AAA on Cisco NX-OS devices.

AAA security services

The AAA feature allows you to verify the identity of, grant access to, and track the actions of users managing a Cisco NX-OS device. Cisco NX-OS devices support Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control System Plus (TACACS+) protocols.

Based on the user ID and password combination that you provide, Cisco NX-OS devices perform local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the Cisco NX-OS device and AAA servers. You can configure a common secret key for all AAA servers or for only a specific AAA server.

AAA security provides the following services:

- **Authentication**

Identifies users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption.

Authentication is the process of verifying the identity of the person or device accessing the Cisco NX-OS device, which is based on the user ID and password combination provided by the entity trying to access the Cisco NX-OS device. Cisco NX-OS devices allow you to perform local authentication (using the local lookup database) or remote authentication (using one or more RADIUS or TACACS+ servers).

- **Authorization**

  Provides access control.AAA authorization is the process of assembling a set of attributes that describe what the user is authorized to perform. Authorization in the Cisco NX-OS software is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

- **Accounting**

  Provides the method for collecting information, logging the information locally, and sending the information to the AAA server for billing, auditing, and reporting.

  The accounting feature tracks and maintains a log of every management session used to access the Cisco NX-OS device. You can use this information to generate reports for troubleshooting and auditing purposes. You can store accounting logs locally or send them to remote AAA servers.



Note The Cisco NX-OS software supports authentication, authorization, and accounting independently. For example, you can configure authentication and authorization without configuring accounting.

Benefits of using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS and TACACS+
- Multiple backup devices

Remote AAA services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- It is easier to manage user password lists for each Cisco NX-OS device in the fabric.
- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.
- You can centrally manage the accounting log for all Cisco NX-OS devices in the fabric.

- It is easier to manage user attributes for each Cisco NX-OS device in the fabric than using the local databases on the Cisco NX-OS devices.

AAA server groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers that implements the same AAA protocol. The purpose of a server group is to provide for failover servers in case a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, then that server group option is considered a failure. If required, you can specify multiple server groups. If the Cisco NX-OS device encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA service configuration options

The AAA configuration in Cisco NX-OS devices is service based, which means that you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell (SSH) login authentication
- Console login authentication
- User management session accounting

Table 6: AAA service configuration commands

AAA Service Configuration Option	Related Command
Telnet or SSH login	**aaa authentication login default**
Console login	**aaa authentication login console**
User session accounting	**aaa accounting default**

You can specify the following authentication methods for the AAA services:

- **All RADIUS servers**

  Uses the global pool of RADIUS servers for authentication.

- **Specified server groups**

  Uses specified RADIUS, TACACS+, or LDAP server groups you have configured for authentication.

- **Local**

  Uses the local username or password database for authentication.

- **None**

  Specifies that no AAA authentication be used.

Note If you specify the all RADIUS servers method, rather than a specified server group method, the Cisco NX-OS device chooses the RADIUS server from the global pool of configured RADIUS servers, in the order of configuration. Servers from this global pool are the servers that can be selectively configured in a RADIUS server group on the Cisco NX-OS device.

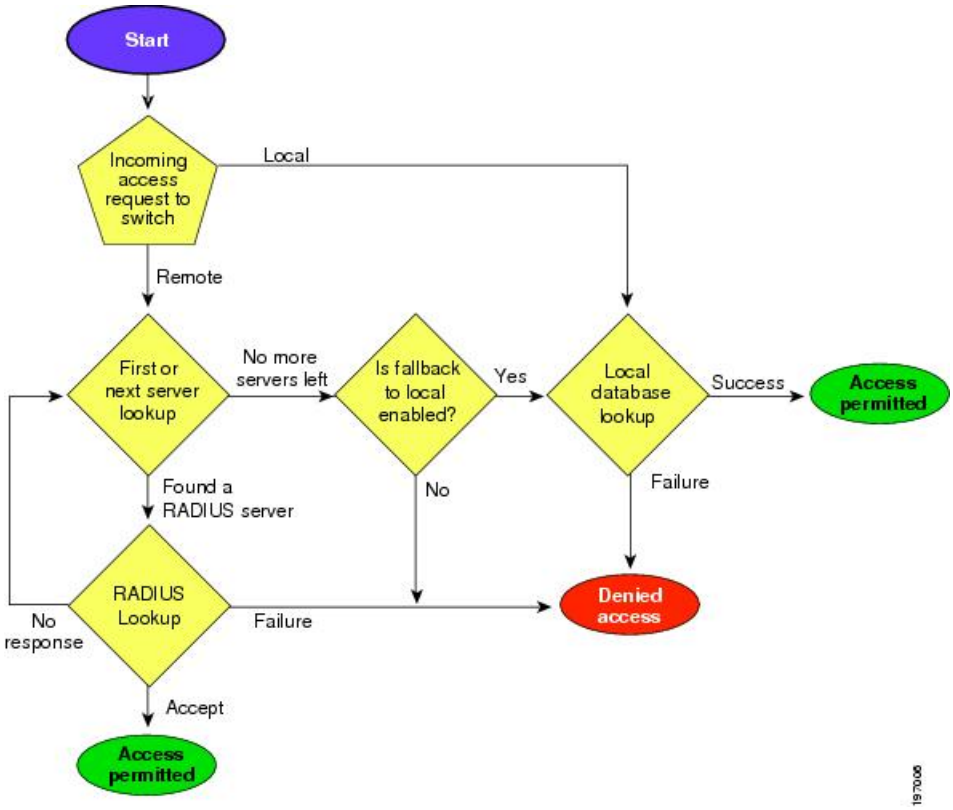
Table 7: AAA authentication methods for AAA services

AAA Service	AAA Methods
Console login authentication	Server groups, local, and none
User login authentication	Server groups, local, and none
User management session accounting	Server groups and local

Note For console login authentication, user login authentication, and user management session accounting, the Cisco NX-OS device tries each option in the order specified. The local option is the default method when other configured options fail. You can disable the local option for the console or default login by using the **no aaa authentication login** { **console** | **default** } **fallback error local** command.

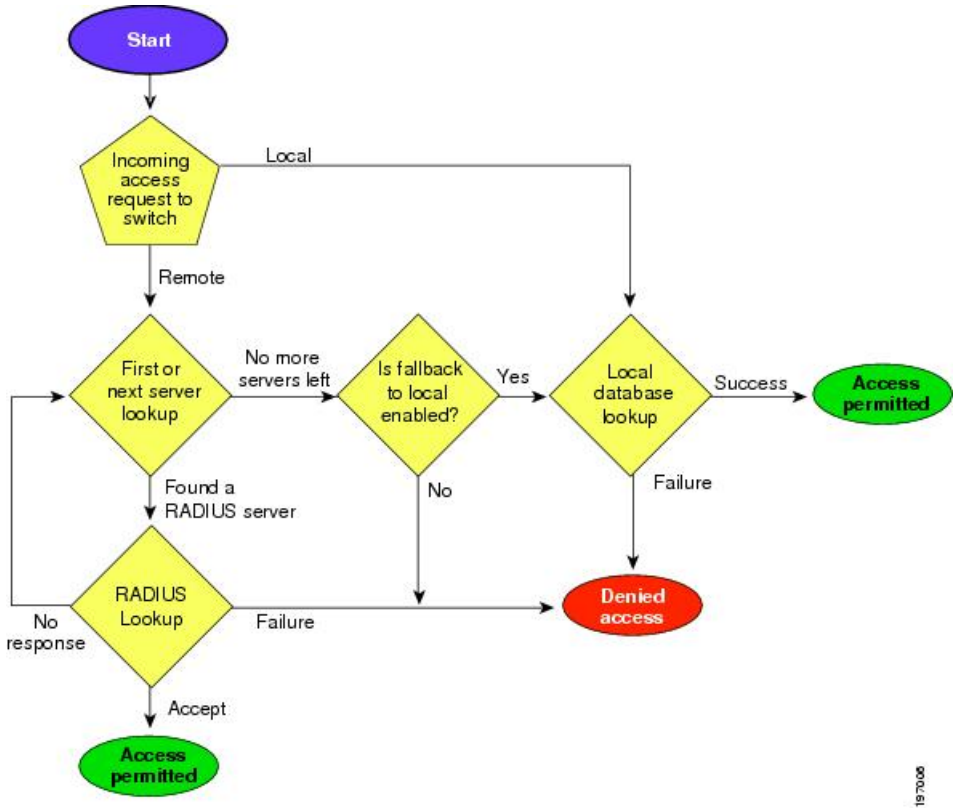
Authentication and authorization process for user login

Figure 1: Authorization and authentication flow for user login



Workflow

Figure 2: Authorization and authentication flow for user login



Here is how the process works:

- Log in to the Cisco NX-OS device using Telnet, SSH, or console.
- Configure AAA server groups using the server group authentication method, then the device sends the request to the first AAA server.
  - If the AAA server fails to respond, the next AAA server is tried, continuing until the remote server responds to the authentication request.
  - If all AAA servers in the server group fail to respond, the servers in the next server group are tried.
  - If all configured methods fail, the local database is used for authentication, if console login fallback is disabled.
- If the Cisco NX-OS device successfully authenticates you through a remote AAA server, these possibilities apply:
  - If the AAA server protocol is RADIUS, user roles specified in the cisco-av-pair attribute are downloaded.
  - If the AAA server protocol is TACACS+, then another request is sent to the same server to get the user roles specified as custom attributes for the shell.

- The device logs you in and assigns roles configured in the local database when your username and password are successfully authenticated locally.

Note "No more server groups left" means that there is no response from any server in all server groups. "No more servers left" means that there is no response from any server within this server group.

AES password encryption and primary encryption keys

You can enable strong, reversible 128-bit Advanced Encryption Standard (AES) password encryption, also known as type-6 encryption. To start using type-6 encryption, you must enable the AES password encryption feature and configure a primary encryption key, which is used to encrypt and decrypt passwords.

After you enable AES password encryption and configure a primary key, all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) are stored in type-6 encrypted format, unless you disable type-6 password encryption. You can also configure Cisco NX-OS to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

Prerequisites for AAA

Remote AAA servers have the following prerequisites:

- Ensure that at least one RADIUS, TACACS+, or LDAP server is reachable through IP.
- Ensure that the Cisco NX-OS device is configured as a client of the AAA servers.
- Ensure that the secret key is configured on the Cisco NX-OS device and the remote AAA servers.
- Ensure that the remote server responds to AAA requests from the Cisco NX-OS device.

Guidelines and limitations for AAA

Use this reference to review the guidelines and limitations for AAA.

AAA has the following guidelines and limitations:

- If you have a user account that is configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- Cisco Nexus 9000 Series switches support the **aaa authentication login ascii-authentication** command only for TACACS+ (and not for RADIUS).
- If you modify the default login authentication method (without using the **local** keyword), the configuration overrides the console login authentication method. To explicitly configure the console authentication method, use the **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**} command.
- The **login block-for** and **login quiet-mode** configuration mode commands are renamed to **system login block-for** and **system login quiet-mode** , respectively.

- When you use the **system login quiet-mode access-class QUIET_LIST** command, you must ensure that the access list is correctly defined to only block the specified traffic. For example, if you need to block only the user logins from untrusted hosts, then the access list should specify ports 22, 23, 80, and 443 corresponding to SSH, telnet, and HTTP-based access from those hosts.
- Beginning with Cisco NX-OS release 10.4(3)F, support for SSH based authorization of X.509 certificates using TACACS+ server is being provided on the Cisco Nexus 9000 Series platform switches. This feature can be enabled using **aaa authorization ssh-certificate default group** *tac-group-name* command. For more information, see Configure AAA SSH-Cert-Authorization on TACACS servers, on page 46.

SNMP synchronization guidelines and limitations

- Beginning with Cisco NX-OS Release 10.2(1)F, SNMPV3 attributes can be mentioned before the shell:roles attribute in cisco-av-pair.
- LDAP does not support ‘snmpv3’ attributes.
- The security users created using DCNM (also called as Nexus Dashboard Fabric Controller from Release 12.0.1a) will not have a corresponding SNMPv3 profile when the desynchronization CLI is enabled. When the synchronization is disabled, the users created on the security component can log in to the switch, but the switches will not be discovered by the controller, as the controller uses the SNMP configuration created for the security user to discover the switch. Furthermore, the SNMP does not recognize the security users created due to the desynchronized state of the userDB, resulting in failure to discover the switch. Therefore, to have the switches discovered by the controller, the SNMP user must be explicitly created. It is not recommended to use the desynchronization CLI along with DCNM functionality. For more information, refer to the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.
- Beginning with Cisco NX-OS Release 10.2(2)F, a new desynchronization CLI is introduced to provide you an option to disable the user synchronization between the SNMP and the security components. For more information, refer to the *Configuring SNMP* chapter in the *System Management Configuration Guide*.

  For more information about the Cisco Nexus 9000 switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.
- When the desynchronization CLI is enabled, remote users will not be synced to SNMP database.

Guidelines and limitations for AAA on Cisco N9800 Series switches

- Beginning with Cisco NX-OS Release 10.3(1)F, AAA is supported on the Cisco Nexus 9808 switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, AAA is supported on Cisco Nexus X98900CD-A, and X9836DM-A line cards with 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, AAA is supported on the Cisco Nexus 9804 switches, X98900CD-A, and X9836DM-A line cards.

Default settings for AAA

Use this reference to review the default settings for AAA.

Table 8: Default AAA parameter settings

Parameters	Default
Console authentication method	local
Default authentication method	local
Login authentication failure messages	Disabled
CHAP authentication	Disabled
MSCHAP authentication	Disabled
Default accounting method	local
Accounting log display length	250 KB

Configuring AAA

This section describes the tasks for configuring AAA on Cisco NX-OS devices.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Note Cisco Nexus 9K Series switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

Process for configuring AAA

Follow these steps to configure AAA authentication and accounting:

Procedure

Step 1 If you want to use remote RADIUS, TACACS+, or LDAP servers for authentication, configure the hosts on your Cisco NX-OS device.

Step 2 Configure console login authentication methods.

Step 3 Configure default login authentication methods for user logins.

Step 4 Configure default AAA accounting default methods.

Configure console login authentication methods

This section describes how to configure the authentication methods for the console login.

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS, TACACS+, or LDAP servers
- Local database on the Cisco NX-OS device
- Username only (none)

The default method is local, but you have the option to disable it.

Note The **group radius** and **group** *server-name* forms of the **aaa authentication** command refer to a set of previously defined RADIUS servers. Use the **radius-server host** command to configure the host servers. Use the **aaa group server radius** command to create a named group of servers.

Note If you perform a password recovery when remote authentication is enabled, local authentication becomes enabled for console login as soon as the password recovery is done. As a result, you can log into the Cisco NX-OS device through the console port using the new password. After login, you can continue to use local authentication, or you can enable remote authentication after resetting the admin password configured at the AAA servers. For more information about the password recovery process, see the *Cisco Nexus 9000 Series NX-OS Troubleshooting Guide.*

Before you begin

Configure RADIUS, TACACS+, or LDAP server groups, as needed.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters configuration mode.

Step 2 **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**}

Example:

```
switch(config)#aaa authentication login console group radius
```

Configures login authentication methods for the console.

The *group-list* argument consists of a space-delimited list of group names. The group names are the following:

- **radius**

  Uses the global pool of RADIUS servers for authentication.

- *named-group*

  Uses a named subset of RADIUS, TACACS+, or LDAP servers for authentication.

The **local** method uses the local database for authentication, and the **none** method specifies that no AAA authentication be used.

The default console login method is **local** , which is used when no methods are configured or when all the configured methods fail to respond, unless fallback to local is disabled for the console login.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show aaa authentication**

Example:

```
switch#show aaa authentication
```

Displays the configuration of the console login authentication methods.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure default login authentication methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS, TACACS+, or LDAP servers
- Local database on the Cisco NX-OS device
- Username only

The default method is local, but you have the option to disable it.

Before you begin

Configure RADIUS, TACACS+, or LDAP server groups, as needed.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters configuration mode.

Step 2 **aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**}

Example:

```
switch(config)#aaa authentication login default group radius
```

Configures the default authentication methods.

The *group-list* argument consists of a space-delimited list of group names. The group names are the following:

- **radius** —Uses the global pool of RADIUS servers for authentication.
- *named-group* —Uses a named subset of RADIUS, TACACS+, or LDAP servers for authentication.

The **local** method uses the local database for authentication, and the **none** method specifies that no AAA authentication be used. The default login method is **local** , which is used when no methods are configured or when all the configured methods fail to respond, unless fallback to local is disabled for the console login.

You can configure one of the following:

- AAA authentication groups
- AAA authentication groups with no authentication
- Local authentication
- No authentication

Note
The **local** keyword is not supported (and is not required) when configuring AAA authentication groups because local authentication is the default if remote servers are unreachable. For example, if you configure **aaa authentication login default group g1** , local authentication is tried if you are unable to authenticate using AAA group g1. In contrast, if you configure **aaa authentication login default group g1 none** , no authentication is performed if you are unable to authenticate using AAA group g1.

Note
Cisco NX-OS AAA authentication does not support hashed key and only supports Type 6/7 keys.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show aaa authentication**

Example:

```
switch#show aaa authentication
```

Displays the configuration of the default login authentication methods.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Disable fallback to local authentication

By default, if remote authentication is configured for console or default login and all AAA servers are unreachable (resulting in an authentication error), the Cisco NX-OS device falls back to local authentication to ensure that users aren't locked out of the device. However, you can disable fallback to local authentication in order to increase security.



Caution Disabling fallback to local authentication can lock your Cisco NX-OS device, forcing you to perform a password recovery in order to gain access. To prevent being locked out of the device, we recommend that you disable fallback to local authentication for only the default login or the console login, not both.

Before you begin

Configure remote authentication for the console or default login.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters configuration mode.

Step 2 **no aaa authentication login** {**console** | **default**} **fallback error local**

Example:

```
switch(config)#no aaa authentication login console fallback error local
```

Disables fallback to local authentication for the console or default login if remote authentication is configured and all AAA servers are unreachable.

The following message appears when you disable fallback to local authentication:

```
"WARNING!!! Disabling fallback can lock your switch."
```

Step 3 (Optional) **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show aaa authentication**

Example:

```
switch#show aaa authentication
```

Displays the configuration of the console and default login authentication methods.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable the default user role for AAA authentication

You can allow remote users who do not have a user role to log in to the Cisco NX-OS device through a RADIUS or TACACS+ remote authentication server using a default user role. When you disable the AAA default user role feature, remote users who do not have a user role cannot log in to the device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters configuration mode.

Step 2 **aaa user default-role**

Example:

```
switch(config)#aaa user default-role
```

Enables the default user role for AAA authentication. The default is enabled.

You can disable the default user role feature by using the **no** form of this command.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show aaa user default-role**

Example:

```
switch#show aaa user default-role
```

Displays the AAA default user role configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable login authentication failure messages

When you log in, the login is processed by rolling over to the local user database if the remote AAA servers do not respond. In such cases, the following messages display on the user's terminal if you have enabled login failure messages:

```
Remote AAA servers unreachable; local authentication done.
Remote AAA servers unreachable; local authentication failed.
```

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters configuration mode.

Step 2 **aaa authentication login error-enable**

Example:

```
switch(config)#aaa authentication login error-enable
```

Enables login authentication failure messages. The default is disabled.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show aaa authentication**

Example:

```
switch#show aaa authentication
```

Displays the login failure message configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Log successful and failed login attempts

You can configure the switch to log all successful and failed login attempts to the configured syslog server.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 Required: [**no**] **login on-failure log**

Example:

```
switch(config)#login on-failure log
```

Logs all failed authentication messages to the configured syslog server only if the logging level is set to 6. With this configuration, the following syslog message appears after the failed login:

AUTHPRIV-3-SYSTEM_MSG: pam_aaa:Authentication failed for user admin from 172.22.00.00

Note
When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message. If these additional messages need to be ignored, the authpriv value should be set to 3.

Step 3 Required: [**no**] **login on-success log** | **logging level** [**authpriv** *value* | **daemon** *value*]

Example:

```
switch(config)#login on-success log
switch(config)#logging level authpriv 6
switch(config)#logging level daemon 6
```

Logs all successful authentication messages to the configured syslog server only if the logging level is set to 6. With this configuration, the following syslog message appears after the successful login:

AUTHPRIV-6-SYSTEM_MSG: pam_aaa:Authentication success for user admin from 172.22.00.00

Note
When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message. If these additional messages need to be ignored, the authpriv value should be set to 3.

Step 4 (Optional) **show login on-failure log**

Example:

```
switch#show login on-failure log
```

Displays whether the switch is configured to log failed authentication messages to the syslog server.

Step 5 (Optional) **show login on-successful log**

Example:

```
switch#show login on-successful log
```

Displays whether the switch is configured to log successful authentication messages to the syslog server.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure login block per user

Ensure that the switch is in global configuration mode.

The Login Block Per User feature helps detect suspected Denial of Service (DoS) attacks and to slow down dictionary attacks. This feature is applicable for local users and remote users. Use this task to configure login parameters to block a user after failed login attempts.

Note From Release 9.3(7), you can configure login block for remote users.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 **aaa authentication rejected** *attempts* **in** *seconds* **ban** *seconds*

Example:

```
switch(config)#aaa authentication rejected 3 in 20 ban 300
```

Configures login parameters to block a user.

Note
Use **no aaa authentication rejected** command to revert to the default login parameters.

Step 3 **exit**

Example:

```
switch(config)#exit
```

Exits to privileged EXEC mode.

Step 4 (Optional) **show running-config**

Example:

```
switch#show running-config
```

Displays the login parameters.

Step 5 **show aaa local user blocked**

Example:

```
switch#show aaa local user blocked
```

Displays the blocked local users.

Step 6 **clear aaa local user blocked {username user| all}**

Example:

```
switch#clear aaa local user blocked username testuser
```

Clears the blocked local users.

all –Clears all the blocked local users.

Step 7 **show aaa user blocked**

Example:

```
switch#show aaa user blocked
```

Displays all blocked local and remote users.

Step 8 (Optional) **clear aaa user blocked {username user| all}**

Example:

```
switch#clear aaa user blocked username testuser
```

Clears all blocked local and remote users.

all – Clears all the blocked local and remote users.

Note Only network-admin, and vdc-admin have privileges to run the show and clear commands.

The following example shows how to configure the login parameters to block a user for 300 seconds when three login attempts fail within a period of 20 seconds:

```
switch(config)#aaa authentication rejected 3 in 20 ban 300
switch#show running-config | include rejected
aaa authentication rejected 3 in 20 ban 300
switch#show aaa local user blocked
Local-user                State
testuser                  Watched (till 11:34:42 IST Nov 12 2020)
switch#clear aaa local user blocked username testuser
switch#show aaa user blocked
```

```
Local-user                State
testuser                  Watched (till 11:34:42 IST Nov 12 2020)
switch#clear aaa user blocked username testuser
```

Create AAA bypass user

Exclude specific users from remote AAA checks and accounting to enable internal processes or automation to function without standard user authentication restrictions.

Use this procedure when select users or system processes require bypassing remote AAA controls to operate successfully on the switch.

Follow these steps to create an AAA bypass user.

Before you begin

- Verify you have administrator access to the switch.
- Ensure you understand which users require AAA bypassing for automation or integration

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Use the **aaa bypass user** command to exempt the specified user from remote accounting and authorization.

When configuring user bypass with the **aaa bypass user** command, you can choose to bypass either accounting, authorization, or both for a specified user. This allows for granular control over which AAA processes are skipped for each user.

- **accounting** : Only bypasses AAA accounting for the specified user. Authorization and authentication still apply.

  ```
  switch(config)#aaa bypass-user user1 accounting
  ```

- **authorization** : Only bypasses AAA authorization for the specified user. Accounting and authentication still apply.

  ```
  switch(config)#aaa bypass-user user1 authorization
  ```

- **No keyword** (just pressing Enter): Bypasses both authorization and accounting for the user.

  ```
  switch(config)#aaa bypass-user user1
  ```

Step 3 Exit global configuration mode.

Example:

```
switch(config)#exit
switch#
```

Step 4 (Optional) Verify user activity in locally captured accounting logs.

Example:

```
switch#show accounting log
```

The specified user is now exempt from remote AAA checks and accounting; verification can be performed using local accounting logs.

Enable CHAP authentication

The Cisco NX-OS software supports the Challenge Handshake Authentication Protocol (CHAP), a challenge-response authentication protocol that uses the industry-standard Message Digest (MD5) hashing scheme to encrypt responses. You can use CHAP for user logins to a Cisco NX-OS device through a remote authentication server (RADIUS or TACACS+).

By default, the Cisco NX-OS device uses Password Authentication Protocol (PAP) authentication between the Cisco NX-OS device and the remote server. If you enable CHAP, you need to configure your RADIUS or TACACS+ server to recognize the CHAP vendor-specific attributes (VSAs).

Note Cisco Nexus 9K Series switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors. For example:

```
2017 Jun 14 16:14:15 N9K-1 %RADIUS-2-RADIUS_NO_AUTHEN_INFO: ASCII authentication not supported
2017 Jun 14 16:14:16 N9K-1 %AUTHPRIV-3-SYSTEM_MSG: pam_aaa:Authentication failed from
192.168.12.34 - dcos_sshd[16804]
```

Table 9: CHAP RADIUS and TACACS+ VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	CHAP-Challenge	Contains the challenge sent by an AAA server to a CHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	CHAP-Response	Contains the response value provided by a CHAP user in response to the challenge. It is used only in Access-Request packets.

Before you begin

Disable AAA ASCII authentication for logins.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters configuration mode.

Step 2 **no aaa authentication login ascii-authentication**

Example:

```
switch(config)#no aaa authentication login ascii-authentication
```

Disables ASCII authentication.

Step 3 **aaa authentication login chap enable**

Example:

```
switch(config)#aaa authentication login chap enable
```

Enables CHAP authentication. The default is disabled.

Note
You cannot enable both CHAP and MSCHAP or MSCHAP V2 on your Cisco NX-OS device.

Step 4 (Optional) **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 5 (Optional) **show aaa authentication login chap**

Example:

```
switch#show aaa authentication login chap
```

Displays the CHAP configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable MSCHAP or MSCHAP V2 authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. The Cisco NX-OS software also supports MSCHAP Version 2 (MSCHAP V2). You can use MSCHAP for user logins to a Cisco NX-OS device through a remote authentication server (RADIUS or TACACS+). MSCHAP V2 only supports user logins to a Cisco NX-OS device through remote authentication RADIUS servers. If you configure a TACACS+ group with MSCHAP V2, the AAA default login authentication uses the next configured method, or the local method, if no other server group is configured.

Note The Cisco NX-OS software may display the following message:

" Warning: MSCHAP V2 is supported only with Radius."

This warning message is informational only and does not affect MSCHAP V2 operation with RADIUS.

By default, the Cisco NX-OS device uses Password Authentication Protocol (PAP) authentication between the Cisco NX-OS device and the remote server. If you enable MSCHAP or MSCHAP V2, you need to configure your RADIUS server to recognize the MSCHAP and MSCHAP V2 vendor-specific attributes (VSAs).

Table 10: MSCHAP and MSCHAP V2 RADIUS VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by an AAA server to an MSCHAP or MSCHAP V2 user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by an MSCHAP or MSCHAP V2 user in response to the challenge. It is only used in Access-Request packets.

Before you begin

Disable AAA ASCII authentication for logins.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters configuration mode.

Step 2 **no aaa authentication login ascii-authentication**

Example:

```
switch(config)#no aaa authentication login ascii-authentication
```

Disables ASCII authentication.

Step 3 **aaa authentication login** {**mschap** | **mschapv2**} **enable**

Example:

```
switch(config)#aaa authentication login mschap enable
```

Enables MSCHAP or MSCHAP V2 authentication. The default is disabled.

Note
You cannot enable both MSCHAP and MSCHAP V2 on your Cisco NX-OS device.

Step 4 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 5 (Optional) **show aaa authentication login** {**mschap** | **mschapv2**}

Example:

```
switch#show aaa authentication login mschap
```

Displays the MSCHAP or MSCHAP V2 configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure AAA authorization on LDAP servers

You can configure the default AAA authorization method for LDAP servers.

Before you begin

Enable LDAP.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **aaa authorization ssh-certificate default** {**group** *group-list* [**none**] | **local** | **none**}

Example:

```
switch(config)#aaa authorization ssh-certificate default group ldap1 ldap2
```

Configures the default AAA authorization method for the LDAP servers.

The **ssh-certificate** keyword configures LDAP or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.

The *group-list* argument consists of a space-delimited list of LDAP server group names. Servers belonging to this group are contacted for AAA authorization. The **local** method uses the local database for authorization, and the **none** method specifies that no AAA authorization be used.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 4 (Optional) **show aaa authorization** [**all**]

Example:

```
switch#show aaa authorization
```

Displays the AAA authorization configuration. The **all** keyword displays the default values.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Enable TACACS+, on page 97

Configure AAA SSH-Cert-Authorization on TACACS servers

To configure AAA SSH-Cert-Authorization on TACACS Servers, follow these steps:

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **aaa authorization ssh-certificate default** {**group** *group-list* [**none**] | **local** | **none**}

Example:

```
switch(config)#aaa authorization ssh-certificate default group tac1
```

Configures the default AAA authorization-method for SSH request having X509 certificate as TACACS server-group(s).

The **ssh-certificate** keyword configures TACACS or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.

The *group-list* argument consists of a space-delimited list of TACACS server group names. Servers belonging to this group are contacted for AAA authorization. The **local** method uses the local database for authorization, and the **none** method specifies that no AAA authorization be used.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 4 (Optional) **show aaa authorization** [**all**]

Example:

```
switch#show aaa authorization
```

Displays the AAA authorization configuration. The **all** keyword displays the default values.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config
startup-config
```

Copies the running configuration to the startup configuration.

Configure AAA accounting default methods

Cisco NX-OS software supports TACACS+ and RADIUS methods for accounting. Cisco NX-OS devices report user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server.

When you activate AAA accounting, the Cisco NX-OS device reports these attributes as accounting records, which are then stored in an accounting log on the security server.

You can create default method lists defining specific accounting methods, which include the following:

- **RADIUS server group**

  Uses the global pool of RADIUS servers for accounting.

- **Specified server group**

  Uses a specified RADIUS or TACACS+ server group for accounting.

- **Local**

  Uses the local username or password database for accounting.

Note If you have configured server groups and the server groups do not respond, by default, the local database is used for authentication.

Before you begin

Configure RADIUS or TACACS+ server groups, as needed.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters configuration mode.

Step 2 **aaa accounting default** {**group** *group-list* | **local**}

Example:

```
switch(config)#aaa accounting default group radius
```

Configures the default accounting method.

The *group-list* argument consists of a space-delimited list of group names. The group names are the following:

- **radius** —Uses the global pool of RADIUS servers for accounting.
- *named-group* —Uses a named subset of TACACS+ or RADIUS servers for accounting.

The **local** method uses the local database for accounting.

The default method is **local** , which is used when no server groups are configured or when all the configured server groups fail to respond.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show aaa accounting**

Example:

```
switch#show aaa accounting
```

Displays the configuration AAA accounting default methods.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Using AAA server VSAs with Cisco NX-OS devices

You can use vendor-specific attributes (VSAs) to specify Cisco NX-OS user roles and SNMPv3 parameters on AAA servers.

About VSAs

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use RADIUS servers for authentication on a Cisco NX-OS device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

VSA format

The following VSA protocol options are supported by the Cisco NX-OS software:

- **Shell**

  Protocol used in access-accept packets to provide user profile information.

- **Accounting**

  Protocol used in accounting-request packets. If a value contains any white spaces, put it within double quotation marks.

The following attributes are supported by the Cisco NX-OS software:

- **roles**

  Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space. For example, if you belong to role network-operator and network-admin, the value field would be network-operator network-admin. This subattribute is sent in the VSA portion of the Access-Accept frames from the RADIUS server, and it can only be used with the shell protocol value. These examples use the roles attribute:

  ```
  shell:roles=network-operator network-admin
  ```

  ```
  shell:roles*network-operator network-admin
  ```

  The following examples show the roles attribute as supported by FreeRADIUS:

  ```
  Cisco-AVPair = shell:roles=\network-operator network-admin\
  ```

  ```
  Cisco-AVPair = shell:roles*\network-operator network-admin\
  ```

Note When you specify a VSA as shell:roles*"network-operator network-admin" or "shell:roles*\"network-operator network-admin\"", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

- **accountinginfo**

Stores accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

Specifying Cisco NX-OS user roles and SNMPv3 parameters on AAA servers

You can use the VSA cisco-av-pair on AAA servers to specify user role mapping for the Cisco NX-OS device using this format:

```
shell:roles="roleA roleB …"
```

If you do not specify the role option in the cisco-av-pair attribute, the default user role is network-operator.

The SNMPv3 attributes should come together, either before the shell attributes or after. You can also specify your SNMPv3 authentication and privacy protocol attributes as follows:

```
snmpv3:auth=”SHA” priv=”AES-128” shell:roles="network-admin" shell:priv-lvl=15

shell:roles="network-admin" shell:priv-lvl=15 snmpv3:auth=”SHA” priv=”AES-128”
```

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If you do not specify these options in the cisco-av-pair attribute, MD5 and DES are the default authentication protocols.

Configuring secure login features

Configure login parameters

You can configure login parameters to automatically block further login attempts when a possible denial-of-service (DoS) attack is detected and slow down dictionary attacks by enforcing a quiet period if multiple failed connection attempts are detected.

Note This feature restarts if a system switchover occurs or the AAA process restarts.

Note The **login block-for** and **login quiet-mode** configuration mode commands have been renamed to **system login block-for** and **system login quiet-mode** , respectively.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 [**no**] **system login block-for** *seconds* **attempts** *tries* **within** *seconds*

Example:

```
switch(config)#system login block-for 100 attempts 2 within 60
```

Configures the quiet mode time period. The range for all arguments is from 1 to 65535.

The example shows how to configure the switch to enter a 100-second quiet period if 2 failed login attempts are exceeded within 60 seconds.

After you enter this command, all login attempts made through Telnet or SSH are denied during the quiet period. Access control lists (ACLs) are not exempt from the quiet period until the **system** command is entered.

Note
You must enter this command before any other login command can be used.

Step 3 (Optional) [**no**] **system login quiet-mode access-class** *acl-name*

Example:

```
switch(config)#system login quiet-mode access-class myacl
```

Specifies an ACL that is to be applied to the switch when it changes to quiet mode. When the switch is in quiet mode, all login requests are denied, and the only available connection is through the console.

Step 4 (Optional) **show system login** [**failures**]

Example:

```
switch#show system login
```

Displays the login parameters. The **failures** option displays information related only to failed login attempts.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Restrict user login sessions

You can restrict the maximum number of simultaneous login sessions per user. Doing so prevents users from having multiple unwanted sessions and solves the potential security issue of unauthorized users accessing a valid SSH or Telnet session.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 [**no**] **user max-logins** *max-logins*

Example:

```
switch(config)#user max-logins 1
```

Restricts the maximum number of simultaneous login sessions per user. The range is from 1 to 7. If you set the maximum login limit as 1, only one Telnet or SSH session is allowed per user.

Note
The configured login limit applies to all users. You cannot set a different limit for individual users.

Step 3 (Optional) **show running-config all | include max-login**

Example:

```
switch#show running-config all | include max-login
```

Displays the maximum number of login sessions allowed per user.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Restrict the password length

You can restrict the minimum and maximum length of the user password. This feature enables you to increase system security by forcing the user to provide a strong password.

Before you begin

You must enable password strength checking using the **password strength-check** command. If you restrict the password length but do not enable password strength checking and the user enters a password that is not within the restricted length, an error appears, but a user account is created. To enforce the password length and prevent a user account from being created, you must enable password strength checking and restrict the password length.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 [**no**] **userpassphrase** { **min-length** *min-length* | **max-length** *max-length* }

Example:

```
switch(config)#userpassphrase min-length 8 max-length 80
```

Restricts the minimum and/or maximum length of the user password. The minimum password length is from 4 to 127 characters, and the maximum password length is from 80 to 127 characters.

Step 3 (Optional) **show userpassphrase** {**length** | **max-length** | **min-length**}

Example:

```
switch#show userpassphrase length
```

Displays the minimum and maximum length of the user password.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable the password prompt for the username

You can configure the switch to prompt the user to enter a password after entering the username.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 **password prompt username**

Example:

```
switch(config)#password prompt username
Password prompt username is enabled.
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.
```

Configures the switch to prompt the user to enter a password after she enters the **username** command without the **password** option or the **snmp-server user** command. The password that the user enters will be hidden. You can use the **no** form of this command to disable this feature.

Step 3 (Optional) **copy running-config startup-config**

Example:

```
switch(config)#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure the shared secret for RADIUS or TACACS+

The shared secret that you configure for remote authentication and accounting between the switch and the RADIUS or TACACS+ server should be hidden because it is sensitive information. You can use a separate

command to generate an encrypted shared secret for the **radius-server** [**host**] **key** and **tacacs-server** [**host**] **key** commands. The SHA256 hashing method is used to store the encrypted shared secret.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **generate type7_encrypted_secret**

Example:

```
switch(config)#generate type7_encrypted_secret
Type-7 (Vigenere) Encryption,
 Use this encrypted secret to configure radius and tacacs shared secret with key type 7.
 Copy complete secret with double quotes.

Enter plain text secret:
Confirm plain text secret:
Type 7 Encrypted secret is : "fewhg"
```

Configures the RADIUS or TACACS+ shared secret with key type 7. You are prompted to enter the shared secret in plain text twice. The secret is hidden as you enter it. Then an encrypted version of the secret appears.

Note
You can generate the encrypted equivalent of a plain-text secret separately and configure the encrypted shared secret later using the **radius-server** [**host**] **key** and **tacacs-server** [**host**] **key** commands.

Step 3 (Optional) **copy running-config startup-config**

Example:

```
switch(config)#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Monitor and clear the local AAA accounting log

The Cisco NX-OS device maintains a local log for the AAA accounting activity. You can monitor this log and clear it.

Procedure

Step 1 **show accounting log** [*size* | **last-index** | **start-seqnum** *number* | **start-time** *year month day hh* **:** *mm* **:** *ss*]

Example:

```
switch#show accounting log
```

Displays the accounting log contents. By default, the command output contains up to 250,000 bytes of the accounting log. You can use the *size* argument to limit command output. The range is from 0 to 250000 bytes. You can also specify a starting sequence number or a starting time for the log output.The range of the starting index is from 1 to 1000000. Use the **last-index** keyword to display the value of the last index number in the accounting log file.

Step 2 (Optional) **clear accounting log** [**logflash**]

Example:

```
switch#clear aaa accounting log
```

Clears the accounting log contents. The **logflash** keyword clears the accounting log stored in the logflash.

AAA configuration verification commands

Use this reference to verify the AAA configuration.

To display AAA configuration information, perform one of the following tasks:

Table 11: AAA configuration verification commands

Command	Purpose
show aaa accounting	Displays AAA accounting configuration.
show aaa authentication [**login** {**ascii-authentication** | **chap** | **error-enable** | **mschap** | **mschapv2**}]	Displays AAA authentication login configuration information.
show aaa groups	Displays the AAA server group configuration.
show login [**failures**]	Displays the login parameters. The **failures** option displays information related only to failed login attempts. **Note** The **clear login failures** command clears the login failures in the current watch period.
show login on-failure log	Displays whether the switch is configured to log failed authentication messages to the syslog server.

Command	Purpose
show login on-successful log	Displays whether the switch is configured to log successful authentication messages to the syslog server.
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration.
show running-config all | include max-login	Displays the maximum number of login sessions allowed per user.
show startup-config aaa	Displays the AAA configuration in the startup configuration.
show userpassphrase {**length** | **max-length** | **min-length**}	Displays the minimum and maximum length of the user password.
show userpassphrase sequence alphabet length	Displays the maximum alphabet sequence length of the user password.
show userpassphrase sequence keyboard length	Displays the maximum sequence keyboard length of the user password.

Configuration examples for AAA

The following example shows how to configure AAA:

```
aaa authentication login default group radius
aaa authentication login console group radius
aaa accounting default group radius
```

Configuration examples for login parameters

The following example shows how to configure the switch to enter a 100-second quiet period if 3 failed login attempts is exceeded within 60 seconds. This example shows no login failures.

Login failure command examples

```
switch#configure terminal
switch(config)#system login block-for 100 attempts 3 within 60
switch#show login

No Quiet-Mode access list has been configured, default ACL will be applied.

 Switch is enabled to watch for login Attacks.
 If more than 3 login failures occur in 60 seconds or less,
 logins will be disabled for 100 seconds.
```

```
 Switch presently in Normal-Mode.
 Current Watch Window remaining time 45 seconds.
 Present login failure count 0.

switch#show login failures
*** No logged failed login attempts with the device.***
```

Quiet-mode ACL configuration example

The following example shows how to configure a quiet-mode ACL. All login requests are denied during the quiet period except hosts from the myacl ACL. This example also shows a login failure.

Quiet-mode ACL command examples

```
switch#configure terminal
switch(config)#system login block-for 100 attempts 3 within 60
switch(config)#system login quiet-mode access-class myacl

switch#show login

 Switch is enabled to watch for login Attacks.
 If more than 3 login failures occur in 60 seconds or less,
 logins will be disabled for 100 seconds.

 Switch presently in Quiet-Mode.
 Will remain in Quiet-Mode for 98 seconds.
 Denying logins from all sources.

switch#show login failures
Information about last 20 login failure's with the device.
--------------------------------------------------------------------------------
Username      Line                            SourceIPAddr    Appname    TimeStamp
--------------------------------------------------------------------------------
asd                /dev/pts/0         192.0.2.1   login      Mon Aug  3 18:18:54 2015
qweq               /dev/pts/0         192.0.2.1   login      Mon Aug  3 18:19:02 2015
qwe                /dev/pts/0         192.0.2.1   login      Mon Aug  3 18:19:08 2015
--------------------------------------------------------------------------------
```

Configuration examples for the password prompt feature

Explains configuration examples for the password prompt feature.

Username password prompt

The following example shows how to configure the switch to prompt the user to enter a password after entering the **username** command and the error message that displays if user does not enter a password.

```
switch#configure terminal
switch(config)#password prompt username
Password prompt username is enabled.
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.

switch(config)#username user1
Enter password:
```

```
Confirm password:
warning: password for user:user1 not set. S/he may not be able to login
```

SNMP server user password prompt

The following example shows how to configure the switch to prompt the user to enter a password after she enters the **snmp-server user** command and the prompts that then display to the user.

```
switch#configure terminal
switch(config)#password prompt username
Password prompt username is enabled.
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.

N9K-1(config)#snmp-server user user1
Enter auth md5 password (Press Enter to Skip):
Enter auth sha password (Press Enter to Skip):
```

Additional references for AAA

Use this reference to find additional references for AAA.

This section includes additional information related to implementing AAA.

Related Documents

Table 12: Related documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*

Standards

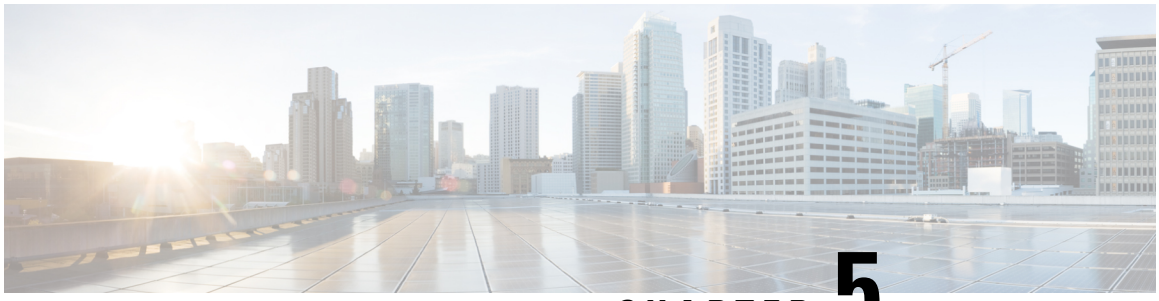
Table 13: Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

Table 14: MIBs

MIBs	MIBs Link
MIBs related to AAA	To locate and download supported MIBs, go to the following URL: https://example.com/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 5

Configuring RADIUS

This chapter describes how to configure the Remote Access Dial-In User Service (RADIUS) protocol on Cisco NX-OS devices.

This chapter includes the following sections:

- About RADIUS, on page 59
- About RADIUS change of authorization, on page 62
- Prerequisites for RADIUS, on page 63
- Guidelines and limitations for RADIUS, on page 63
- Guidelines and limitations for RadSec, on page 64
- Guidelines and limitations for RADIUS change of authorization, on page 65
- Default settings for RADIUS, on page 65
- Configuring RADIUS servers, on page 66
- Enable or disable the dynamic author server, on page 84
- Configure RADIUS change of authorization, on page 85
- RADIUS configuration verification commands, on page 86
- RADIUS change of authorization configuration verification commands, on page 86
- Monitor RADIUS servers, on page 87
- Clear the RADIUS server statistics, on page 87
- Configuration example for RADIUS, on page 88
- Configuration examples of RADIUS change of authorization, on page 89
- Where to go next, on page 89
- Additional references for RADIUS, on page 89

About RADIUS

The RADIUS distributed client/server system allows you to secure networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco NX-OS devices and send authentication and accounting requests to a central RADIUS server that contains all user authentication and network service access information.

RADIUS network environments

RADIUS can be implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

You can use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor network devices, each supporting RADIUS. For example, network devices from several vendors can use a single RADIUS server-based security database.
- Networks already using RADIUS. You can add a Cisco NX-OS device with RADIUS to the network. This action might be the first step when you make a transition to a AAA server.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.
- Networks that support authentication profiles. Using the RADIUS server in your network, you can configure AAA authentication and set up per-user profiles. Per-user profiles enable the Cisco NX-OS device to better manage ports using their existing RADIUS solutions and to efficiently manage shared resources to offer different service-level agreements.

RADIUS operation

When a user attempts to log in and authenticate to a Cisco NX-OS device using RADIUS, the following process occurs:

- The user is prompted for and enters a username and password.
- The username and encrypted password are sent over the network to the RADIUS server.
- The user receives one of the following responses from the RADIUS server:
  - **ACCEPT**

    The user is authenticated.
  - **REJECT**

    The user is not authenticated and is prompted to reenter the username and password, or access is denied.
  - **CHALLENGE**

    A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
  - **CHANGE PASSWORD**

    A request is issued by the RADIUS server, asking the user to select a new password.

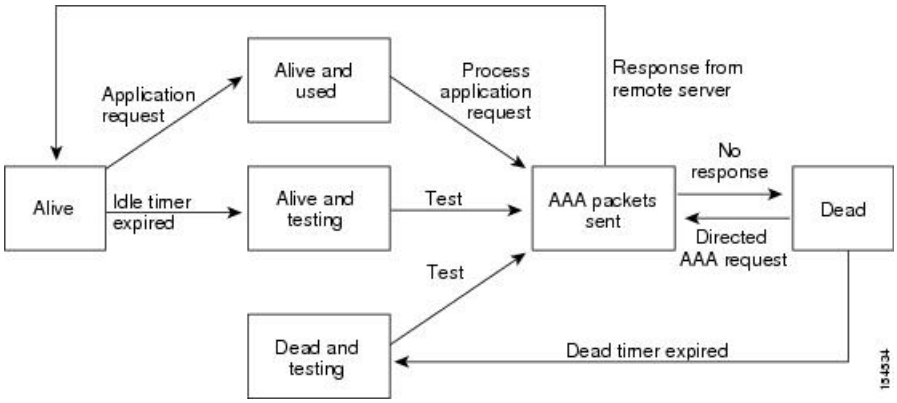
The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 or IPv6 address, access list, and user timeouts.

RADIUS server monitoring

An unresponsive RADIUS server can cause a delay in processing AAA requests. You can configure the Cisco NX-OS device to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The Cisco NX-OS device periodically monitors the dead RADIUS servers and brings them to the alive state once they respond. This monitoring process verifies that a RADIUS server is in a working state before real AAA requests are sent its way. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco NX-OS device displays an error message that a failure is taking place.

Figure 3: RADIUS server states



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The RADIUS server monitoring is performed by sending a test authentication request to the RADIUS server.

Vendor-specific attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use RADIUS servers for authentication on a Cisco NX-OS device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the Cisco NX-OS software:

- **Shell**

Protocol used in access-accept packets to provide user profile information.

- **Accounting**

  Protocol used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The Cisco NX-OS software supports the following attributes:

- **roles**

  Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and network-admin, the value field would be network-operator network-admin. This subattribute, which the RADIUS server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute that is supported by the Cisco Access Control Server (ACS):

  ```
  shell:roles=network-operator network-admin
  shell:roles*"network-operator network-admin
  ```

  The following examples show the roles attribute that is supported by FreeRADIUS:

  ```
  Cisco-AVPair = shell:roles=\network-operator network-admin\
  Cisco-AVPair = shell:roles*\network-operator network-admin\
  ```



Note When you specify a VSA as shell:roles*"network-operator network-admin" or "shell:roles*\"network-operator network-admin\"", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

- **accountinginfo**

  Stores accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch. It can be used only with the accounting protocol data units (PDUs).

About RADIUS change of authorization

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. Cisco NX-OS sofware supports the RADIUS Change of Authorization (CoA) request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

When Dot1x is enabled, the network device acts as the authenticator and is responsible for processing dynamic COA per session.

The following requests are supported:

- Session reauthentication
- Session termination

Session reauthentication

To initiate session reauthentication, the authentication, authorization, and accounting (AAA) server sends a standard CoA-Request message that contains a Cisco VSA and one or more session identification attributes. The Cisco VSA is in the form of Cisco:Avpair=“subscriber:command=reauthenticate”.

The current session state determines the response of the device to the message in the following scenarios:

- If the session is currently authenticated by IEEE 802.1x, the device responds by sending an Extensible Authentication Protocol over LAN (EAPOL)-RequestId message to the server.
- If the session is currently authenticated by MAC authentication bypass (MAB), the device sends an access request to the server, passing the same identity attributes used for the initial successful authentication.
- If session authentication is in progress when the device receives the command, the device terminates the process and restarts the authentication sequence, starting with the method configured to be attempted first.

Session termination

A CoA Disconnect-Request terminates the session without disabling the host port. CoA Disconnect-Request termination causes reinitialization of the authenticator state machine for the specified host, but does not restrict the host’s access to the network.

If the session cannot be located, the device returns a Disconnect-NAK message with the “Session Context Not Found” error-code attribute.

If the session is located, but the NAS was unable to remove the session due to some internal error, the device returns a Disconnect-NAK message with the *Session Context Not Removable* error-code attribute.

If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK message.

Prerequisites for RADIUS

RADIUS has the following prerequisites:

- Obtain IPv4 or IPv6 addresses or hostnames for the RADIUS servers.
- Obtain keys from the RADIUS servers.
- Ensure that the Cisco NX-OS device is configured as a RADIUS client of the AAA servers.

Guidelines and limitations for RADIUS

RADIUS has the following guidelines and limitations for Cisco Nexus switches, including platform-specific features and command restrictions.

General guidelines and limitations

- You can configure a maximum of 64 RADIUS servers on the Cisco NX-OS device.
- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- Only the RADIUS protocol supports one-time passwords.
- For N9K-X9636C-R and N9K-X9636Q-R line cards and the N9K-C9508-FM-R fabric module, RADIUS authentication fails for usernames with special characters.
- Cisco Nexus 9K Series switches support the CLI command, **aaa authentication login ascii-authentication** , only for TACAAS+, but not for RADIUS. Ensure that you have disabled **aaa authentication login ascii-authentication** switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.
- Beginning with Cisco NX-OS Release 10.4(4), the radius-server CLI allows a value of 0 for the timeout and retransmit parameters.

  Starting with Cisco NX-OS Release 10.4(4), a value of 0 for timeout and retransmit is also shown in the **show running-config** output.

  During a downgrade between releases 10.4(4) and 9.3(11) or later, without these fixes, any RADIUS server configuration that uses a timeout value of 0 or a retransmit value of 0 could be lost or may not work. To ensure configuration consistency, avoid using 0 as a value for these parameters when migrating between affected releases, or verify support for these values in both the source and target releases.

Guidelines and limitations for RADIUS on Cisco Nexus 9800 series switches

- Beginning with Cisco NX-OS Release 10.3(1)F , RADIUS is supported on the Cisco Nexus 9808 platform switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, RADIUS is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, RADIUS is supported on the Cisco Nexus 9804 switches, X98900CD-A, and X9836DM-A line cards.

Guidelines and limitations for RadSec

Use this reference to review the guidelines and limitations for RadSec.

RadSec has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.3(1)F, the RADIUS Secure (RadSec) support is provided on Cisco Nexus switches to secure the communication between RADIUS/TCP peers at the transport layer.
- RadSec must be enabled/disabled at the switch level, as the combination of servers having different transport protocols (i.e. UDP and TCP-with-TLS) is not possible.
- **radius-server directed-request** command is not supported along with the RadSec feature.

- **test aaa server radius** command is not supported for the RadSec servers, only **test aaa group** command is supported with the RadSec.
- Dot1x is not officially supported with RadSec.
- RADIUS server monitoring is not supported along with the RadSec servers.
- RADIUS server re-transmit and timeout are applicable to UDP based RADIUS mode and not supported for RadSec servers.
- Beginning with Cisco NX-OS Release 10.4(3)F, TLS version 1.3 and 1.2 is supported on Cisco Nexus switches. TLS v1.1 is deprecated.

Guidelines and limitations for RADIUS change of authorization

Use this reference to review the guidelines and limitations for RADIUS change of authorization.

RADIUS Change of Authorization has the following guidelines and limitations:

- RADIUS Change of Authorization is supported on FEX.
- RADIUS change of Authorization is supported for VXLAN EVPN.

Default settings for RADIUS

Use this reference to review the default settings for RADIUS.

The following table lists the default settings for RADIUS parameters, including server roles, timers, ports, and monitoring credentials that are applied unless explicitly modified by the user.

Table 15: Default RADIUS parameter settings

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1
Retransmission timer interval	5 seconds
Authentication port	1812
Accounting port	1813
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring RADIUS servers

This section describes how to configure RADIUS servers on a Cisco NX-OS device.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Note Cisco Nexus 9K Series switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

RADIUS server configuration process

1. Establish the RADIUS server connections to the Cisco NX-OS device.
2. Configure the RADIUS secret keys for the RADIUS servers.
3. If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.
4. If needed, configure any of the following optional parameters:
   - Dead-time interval
   - RADIUS server specification allowed at user login
   - Timeout interval
   - TCP port
5. (Optional) If RADIUS distribution is enabled, commit the RADIUS configuration to the fabric.

Related Topics

Configure RADIUS server hosts, on page 66
Configure global RADIUS keys, on page 67

Configure RADIUS server hosts

To access a remote RADIUS server, you must configure the IP address or hostname of a RADIUS server. You can configure up to 64 RADIUS servers.

Note By default, when you configure a RADIUS server IP address or hostname of the Cisco NX-OS device, the RADIUS server is added to the default RADIUS server group. You can also add the RADIUS server to another RADIUS server group.

Before you begin

Ensure that the server is already configured as a member of the server group.

Ensure that the server is configured to authenticate RADIUS traffic.

Ensure that the Cisco NX-OS device is configured as a RADIUS client of the AAA servers.

Procedure

Step 1 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} command to specify the IPv4 or IPv6 address or hostname for a RADIUS server to use for authentication.

Example:

```
switch#configure terminal
switch(config)#radius-server host 10.10.1.1
```

Step 2 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure a key for a specific RADIUS server, on page 68

Configure global RADIUS keys

Configure global RADIUS keys to establish a shared secret between your device and remote RADIUS servers for secure authentication.

You can configure RADIUS keys for all servers used by the Cisco NX-OS device. A RADIUS key is a shared secret text string between the Cisco NX-OS device and the RADIUS server hosts.

Before you begin

Obtain the RADIUS key values for the remote RADIUS servers.

Configure the RADIUS key on the remote RADIUS servers.

Procedure

Step 1 Use the **radius-server key** [**0** | **6** | **7**] *key-value* command to specify a RADIUS key for all RADIUS servers.

Example:

```
switch#configure terminal
switch(config)#radius-server key 0 QsEfThUkO
switch(config)#exit
```

Example:

```
switch(config)#radius-server key 7 "fewhg"
```

You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters.

By default, no RADIUS key is configured.

Note
If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configure the shared secret for RADIUS or TACACS+, on page 53 .

Step 2 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Note
The RADIUS keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys.

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure a RADIUS server group, on page 74
AES password encryption and primary encryption keys, on page 547

Configure a key for a specific RADIUS server

Configure a key for a specific RADIUS server to ensure secure communication between your device and the RADIUS server.

You can configure a key on the Cisco NX-OS device for a specific RADIUS server. A RADIUS key is a secret text string shared between the Cisco NX-OS device and a specific RADIUS server.

Before you begin

Configure one or more RADIUS server hosts.

Obtain the key value for the remote RADIUS server.

Configure the key on the RADIUS server.

Procedure

Step 1 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **key** [**0** | **6** | **7**] *key-value* command to specify a RADIUS key for a specific RADIUS server.

Example:

```
switch#configure terminal
switch(config)#radius-server host 10.10.1.1 key 0 PlIjUhYg
switch(config)#exit
```

Example:

```
switch(config)#radius-server host 10.10.1.1 key 7 "fewhg"
```

You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters.

This RADIUS key is used instead of the global RADIUS key.

Note
If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configure the shared secret for RADIUS or TACACS+, on page 53 .

Step 2 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Note
The RADIUS keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys.

Step 3 (Optional) Use the **copy running-config startup-config** copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure RADIUS server hosts, on page 66
AES password encryption and primary encryption keys, on page 547

Configure the RADIUS attribute message authenticator

You can configure a RADIUS attribute message authenticator for all servers that use Cisco NX-OS switches. The RADIUS attribute encapsulates Extended Access Protocol (EAP) packets to allow the switch to authenticate dial-in users through EAP using HMAC-MD5.

Note Cisco Fabric Services (CFS) does *not* distribute RADIUS attribute message authenticators.

Beginning with Cisco NX-OS Release 10.2(9)M , the **radius-server attribute message-authenticator** command is introduced on the Cisco Nexus 9000 switches.

Procedure

Step 1 Use the **radius-server attribute message-authenticator** specify a RADIUS attribute message-authenticaton for all RADIUS servers.

Example:

```
switch#configure terminal
switch(config)#radius-server attribute message-authenticator
switch(config)#exit
```

By default, the RADIUS attribute message-authenticator is disabled.

Step 2 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
etransmission count:1
timeout value:5
deadtime value:0

message-authenticator attribute:enabled

source interface:any available
total number of servers:4
following RADIUS servers are configured:
10.10.1.1:
available for authentication on port:1812
available for accounting on port:1813
RADIUS shared secret:********
timeout:60
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure RadSec

RadSec is a protocol for transporting RADIUS datagrams over TLS.

This procedure describes how to enable or disable RadSec on a switch.

Before you begin

- Ensure that the client identity certificate and CA certificate of the server are installed on the switch.
- Ensure that the subject name in the server certificate is matching with the server host name/IP address that is configured on the switch.
- Before configuring AAA authentication and accounting to use RadSec servers, use **test aaa group** command and ensure RadSec authentication is success.
- Configure TLS idle-timeout to maximum value on RadSec server to avoid frequent TLS sessions retries from switch.

Procedure

Step 1 Use the **radius-server secure tls** command to enable the RadSec at global level.

Example:

```
switch#configure terminal
switch(config)#radius-server secure tls
```

Note
This CLI will not change or affect the port numbers that is used for RadSec.

Step 2 Use the **radius-server host t** {*ipv4-address* | *ipv6-address* | *hostname*} **key** {*key*} **auth-port 2083 acct-port 2083 authentication accounting** command to configure the RadSec server with shared secret key along with the authentication and accounting ports.

Example:

```
switch(config)#radius-server host 10.105.222.161 key radsec auth-port 2083 acct-port 2083
authentication accounting
```

Note
For server, the default RadSec port for authentication and accounting is "2083" and the key is “radsec”. For switch, there is no default configuration for RadSec port and key, please add this configuration explicitly as defined on server.

Step 3 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **tls client-trustpoint** *trustpoint* command to configures the TLS client trustpoint where the client identity certificate is installed.

Example:

```
switch(config)#radius-server host 10.105.222.161 tls client-trustpoint rad1
```

Step 4 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **tls idle-timeout** *value* command to configure the TLS idle-timeout.

Example:

```
switch(config)#radius-server host 10.105.222.161 tls idle-timeout 80
```

The default value is 600 seconds.

Note
If there are no transactions from the RadSec client, server can close the connection based on its timeout value. The TLS idle-timeout on the client is not supported in this release. Client does not close connections on its own.



Note When remote user logs-in, you can notice delay in login for approximately 20 seconds i.e when TLS session establishment is happening for the first time between switch and RadSec server, Once TLS sessions are up no delay will be seen for consecutive remote log-ins.



Note When a RadSec client is facing certificate related issues such as no certificate or invalid certificates are being exchanged with the server, you may experience delay in `show run` commands.

About RADIUS with DTLS

RADIUS with DTLS is a protocol introduced in Cisco NX-OS Release 10.4(1)F for transporting RADIUS datagrams over a secure channel using UDP.

- Provides secure communication between RADIUS peers at the transport layer.
- Helps secure RADIUS packet transfer through different administrative domains.
- Protects RADIUS traffic over suspicious and unsafe networks.

Configure RADIUS with DTLS

Before you begin

- Ensure that you create client identity certificate with subject and alternative name same as the IP address/DNS hostname of the switch. Install the client identity certificate on the switch using a trustpoint.
- Ensure that the server certificate of ISE server used for DTLS/RADIUS is installed on the switch.
- Make sure that the CA certificate used to sign client identity certificate is installed in trusted certificate store of ISE server.
- Ensure that the subject name in the server certificate is same as the server hostname/IP address that is configured on the switch.
- Before configuring AAA authentication and accounting groups to use RADIUS servers, check with test aaa group command and ensure that the RADIUS authentication is successful.
- You must enable RADIUS with DTLS protocol at the switch level.
- Configuring combination of RADIUS servers to use different transports protocols such as DTLS and TLS is not supported. You can configure one protocol at an instant.

Procedure

Step 1 Use the **radius-server secure dtls** command to enable the RADIUS with DTLS protocol on the switch.

Example:

```
switch#configure terminal
 switch(config)#radius-server secure dtls
```

Step 2 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **key** {*radius/dtls*} **auth-port 2083 acct-port 2083 authentication accounting** command to configures the RADIUS server with shared secret key along with the authentication and accounting ports.

Example:

```
switch(config)#radius-server host 10.105.222.161 key radius/dtls auth-port 2083 acct-port
2083 authentication accounting
```

Note
The default destination DTLS port for authentication and accounting is **UDP/2083** . There is no default server key for DTLS as per RFC. Ensure that you add this configuration explicitly as defined on server. The ISE server must be pre-set with the "radius/dtls" key at that instant. Check and add the key on the Nexus switch while configuring DTLS with an ISE server.

Step 3 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **dtls client-trustpoint** *trustpoint* command to configures the DTLS client-trustpoint parameter with a trustpoint where the switch identity certificate is installed.

Example:

```
switch(config)#radius-server host 10.105.222.161 dtls client-trustpoint rad1
```

The *rad1* is a trustpoint on the switch which must have the client identity certificate.

Step 4 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **dtls idle-timeout** *value* command to onfigures the DTLS idle-timeout.

Example:

```
switch(config)#radius-server host 10.105.222.161 dtls idle-timeout 80
```

The default value is 600 seconds.

Note
If there are no transactions from the RADIUS client, server can close the connection as per defined timeout value. The DTLS idle-timeout on the client is not supported in this release. Client does not close connections on its own.

Note

- When a remote user logs in, you may notice a delay of approximately 20 seconds, which occurs when the TLS session is being established for the first time between the switch and the RADIUS server. Once the TLS sessions are up, no delay will be seen for consecutive remote logins.
- When a RADIUS client is facing certificate related issues such as no certificate or invalid certificates are being exchanged with the server, you may experience delay in the `show run` commands.

Configure a RADIUS server group

Configure RADIUS server groups to specify one or more remote AAA servers for authentication. All servers in the group must be RADIUS servers.

You can specify one or more remote AAA servers for authentication using server groups. All members of a group must belong to the RADIUS protocol. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service.

Before you begin

Ensure that all servers in the group are RADIUS servers.

Procedure

Step 1 Use the **aaa group server radius** *group-name* command to create a RADIUS server group and enters the RADIUS server group configuration submode for that group.

Example:

```
switch#configure terminal
switch(config)#aaa group server radius RadServer
switch(config-radius)#
```

The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.

To delete a RADIUS server group, use the **no** form of this command.

Note
You are not allowed to delete the default system generated default group (RADIUS).

Step 2 Use the **server** {*ipv4-address* | *ipv6-address* | *hostname*} command to configure the RADIUS server as a member of the RADIUS server group.

Example:

```
switch(config-radius)#server 10.10.1.1
```

If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.

Step 3 (Optional) Use the **deadtime** *minutes* command to configure the monitoring dead time.

Example:

```
switch(config-radius)#deadtime 30
```

The default is 0 minutes. The range is from 1 through 1440.

Note
If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.

Step 4 (Optional) Use the **server** {*ipv4-address* | *ipv6-address* | *hostname*} command to configure the RADIUS server as a member of the RADIUS server group.

Example:

```
switch(config-radius)#server 10.10.1.1
```

Tip
If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.

Step 5 (Optional) Use the **use-vrf** *vrf-name* command to specifies the VRF to use to contact the servers in the server group.

Example:

```
switch(config-radius)#use-vrf vrf1
switch(config-radius)#exit
```

Step 6 (Optional) Use the **show radius-server groups** [*group-name*] command to display the RADIUS server group configuration.

Example:

```
switch#show radius-server groups
```

Step 7 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure the RADIUS Dead-Time interval, on page 82

Configure the global source interface for a RADIUS server group

You can configure a global source interface for RADIUS server groups to use when accessing RADIUS servers. You can also configure a different source interface for a specific RADIUS server group. By default, the Cisco NX-OS software uses any available interface.

Procedure

Step 1 Use the **ip radius source-interface** *interface* command to configure the global source interface for all RADIUS server groups configured on the device.

Example:

```
switch#configure terminal
switch(config)#ip radius source-interface mgmt 0
switch(config)#exit
```

Step 2 (Optional) Use the **show radius-server** command to display the RADIUS server configuration information.

Example:

```
switch#show radius-server
```

Step 3 (Optional) Use the **copy running-config startup config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure a RADIUS server group, on page 74

Allow a user to specify a RADIUS server at login

By default, the Cisco NX-OS device forwards an authentication request based on the default AAA authentication method. You can configure the Cisco NX-OS device to allow the user to specify a VRF and RADIUS server to send the authentication request by enabling the directed-request option. If you enable this option, the user can log in as *username* **@** *vrfname* **:** *hostname* , where *vrfname* is the VRF to use and **hostname** is the name of a configured RADIUS server.

Note If you enable the directed-request option, the Cisco NX-OS device uses only the RADIUS method for authentication and not the default local method.

Note User-specified logins are supported only for Telnet sessions.

Procedure

Step 1 Use the **radius-server directed-request** command to allow users to specify a RADIUS server to send the authentication request when logging in. The default is disabled.

Example:

```
switch#configure terminal
switch(config)#radius-server directed-request
```

Step 2 (Optional) Use the **show radius-server directed-request** command to display the directed request configuration.

Example:

```
switch#show radius-server directed-request
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure the global RADIUS transmission retry count and timeout interval

You can configure a global retransmission retry count and timeout interval for all RADIUS servers. By default, a Cisco NX-OS device retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. The timeout interval determines how long the Cisco NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

Procedure

Step 1 Use the **radius-server retransmit** *count* command to specify the retransmission count for all RADIUS servers.

Example:

```
switch#configure terminal
switch(config)#radius-server retransmit 3
```

The default retransmission count is 1 and the range is from 0 to 5.

Step 2 Use the **radius-server timeout** *seconds* command to specify the transmission timeout interval for RADIUS servers.

Example:

```
switch(config)#radius-server timeout 10
```

The default timeout interval is 5 seconds and the range is from 1 to 60 seconds.

Step 3 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure the RADIUS transmission retry count and timeout interval for a server

By default, a Cisco NX-OS device retries a transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. You can also set a timeout interval that the Cisco NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

Before you begin

Configure one or more RADIUS server hosts.

Procedure

Step 1 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **retransmit** *count* command to specify the retransmission count for a specific server. The default is the global value.

Example:

```
switch#configure terminal
switch(config)#radius-server host server1 retransmit 3
```

Note
The retransmission count value specified for a RADIUS server overrides the count specified for all RADIUS servers.

Step 2 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds* command to specify the transmission timeout interval for a specific server. The default is the global value.

Example:

```
switch(config)#radius-server host server1 timeout 10
```

Note
The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers.

Step 3 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure RADIUS server hosts, on page 66

Configure accounting and authentication attributes for a RADIUS server

You can specify that a RADIUS server is to be used only for accounting purposes or only for authentication purposes. By default, RADIUS servers are used for both accounting and authentication. You can also specify the destination UDP port numbers where RADIUS accounting and authentication messages should be sent if there is a conflict with the default port.

Before you begin

Configure one or more RADIUS server hosts.

Procedure

Step 1 (Optional) Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **acct-port** *udp-port* command to specify a UDP port to use for RADIUS accounting messages.

Example:

```
switch#configure terminal
switch(config)#radius-server host 10.10.1.1 acct-port 2004
```

The range is from 0 to 65535.

Step 2 (Optional) Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **accounting** command to specify to use the RADIUS server only for accounting purposes. The default is both accounting and authentication

Example:

```
switch(config)#radius-server host 10.10.1.1 accounting
```

Step 3 (Optional) Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **auth-port** *udp-port* command to specify a UDP port to use for RADIUS authentication messages.

Example:

```
switch(config)#radius-server host 10.10.2.2 auth-port 2005
```

The default UDP port is 1812. The range is from 0 to 65535.

Step 4 (Optional) Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **authentication** command to specify to use the RADIUS server only for authentication purposes.

Example:

```
switch(config)#radius-server host 10.10.2.2 authentication
switch(config)#exit
```

The default is both accounting and authentication.

Step 5 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Step 6 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure RADIUS server hosts, on page 66

Configure global periodic RADIUS server monitoring

You can monitor the availability of all RADIUS servers without having to configure the test parameters for each server individually. Any servers for which test parameters are not configured are monitored using the global level parameters.

Note Test parameters that are configured for individual servers take precedence over global test parameters.

The global configuration parameters include the username and password to use for the servers and an idle timer. The idle timer specifies the interval in which a RADIUS server receives no requests before the Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the RADIUS database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic RADIUS server monitoring is not performed.

Before you begin

Enable RADIUS.

Procedure

Step 1 Use the **radius-server test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} command to specify parameters for global server monitoring.

Example:

```
switch#configure terminal
switch(config)#radius-server test username user1 password Ur2Gd2BH idle-time 3
```

The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.

Note
For periodic RADIUS server monitoring, the idle timer value must be greater than 0.

Step 2 Use the **radius-server deadtime** *minutes* command to specify the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive.

Example:

```
switch(config)#radius-server deadtime 5
switch(config)#exit
```

The default value is 0 minutes, and the valid range is from 0 to 1440 minutes.

Step 3 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure periodic RADIUS server monitoring on individual servers, on page 81

Configure periodic RADIUS server monitoring on individual servers

Configure periodic monitoring of individual RADIUS servers to proactively check server availability and performance in NX-OS environments.

You can monitor the availability of individual RADIUS servers. The configuration parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval during which a RADIUS server receives no requests before the Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note Test parameters that are configured for individual servers take precedence over global test parameters.

Note For security reasons, we recommend that you do not configure a test username that is the same as an existing user in the RADIUS database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the Cisco NX-OS device does not perform periodic RADIUS server monitoring.

Before you begin

Enable RADIUS.

Add one or more RADIUS server hosts.

Procedure

Step 1 Use the **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} command to specify parameters for individual server monitoring.

Example:

```
switch#configure terminal
switch(config)#radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time
 3
```

The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.

Note
For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.

Step 2 Use the **radius-server deadtime** *minutes* command to specify the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive.

Example:

```
switch(config)#radius-server deadtime 5
switch(config)#exit
```

The default value is 0 minutes, and the valid range is from 1 to 1440 minutes.

Step 3 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure RADIUS server hosts, on page 66
Configure global periodic RADIUS server monitoring, on page 80

Configure the RADIUS Dead-Time interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the Cisco NX-OS device waits after declaring a RADIUS server is dead, before sending out a test packet to determine if the server is now alive. The default value is 0 minutes.



Note When the dead-time interval is 0 minutes, RADIUS servers are not marked as dead even if they are not responding. You can configure the dead-time interval for a RADIUS server group.

Procedure

Step 1 Use the **radius-server deadtime** *minutes* command to configure the dead-time interval.

Example:

```
switch#configure terminal
switch(config)#radius-server deadtime 5
```

The default value is 0 minutes. The range is from 1 to 1440 minutes.

Step 2 (Optional) Use the **show radius-server** command to display the RADIUS server configuration.

Example:

```
switch#show radius-server
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure a RADIUS server group, on page 74

Configuring One-Time passwords

- One-time password (OTP) support is available for Cisco devices through the use of RSA SecurID token servers.
- Users authenticate by entering a personal identification number or one-time password and the token code displayed on their RSA SecurID token.
- The token code changes every 60 seconds to enhance security.

Prerequisites and configuration steps for One-Time passwords

Before configuring one-time passwords, ensure the following prerequisites are met and follow the configuration steps.

- Configure a RADIUS server host and remote default login authentication on the device.
- Install Cisco Secure Access Control Server (ACS) version 4.2.
- Install RSA Authentication Manager version 7.1 (the RSA SecurID token server).
- Install RSA ACE Agent/Client.

1. Enable RSA SecurID token server authentication on Cisco Secure ACS.
2. Add the RSA SecurID token server to the Unknown User Policy database.

Note The token code used for logging into the Cisco device changes every 60 seconds. To prevent problems with device discovery, it is recommended to use different usernames that are present on the Cisco Secure ACS internal database.

Configuring One-Time passwords

For example, after installing the required software and configuring the RADIUS server, enable RSA SecurID authentication and add the token server to the Unknown User Policy database in Cisco Secure ACS to support OTP logins.

Manually monitor a RADIUS server or group

You can manually issue a test message to a RADIUS server or to a server group.

Procedure

Step 1 Use the **test aaa server radius** {*ipv4-address* | *ipv6-address* | *hostname*} [**vrf** *vrf-name*] *username password* command to send a test message to a RADIUS server to confirm availability.

Example:

```
switch#test aaa server radius 10.10.1.1 user1 Ur2Gd2BH
```

Step 2 Use the **test aaa group** *group-name username password* command to send a test message to a RADIUS server group to confirm availability.

Example:

```
switch#test aaa group RadGroup user2 As3He3CI
```

Enable or disable the dynamic author server

You can enable or disable the dynamic author server.

Procedure

Use the **aaa server radius dynamic-author** command to enable the RADIUS dynamic author server.

Example:

```
switch#configure terminal
switch(config)#aaa server radius dynamic-author
```

You can disable the RADIUS dynamic author server using the *no* form of this command.

Configure RADIUS change of authorization

You can configure RADIUS Change of Authorization (CoA) on your switch to enable dynamic interaction with an external policy server and manage AAA server clients, ports, and server keys.

Follow these steps to configure RADIUS CoA.

Procedure

Step 1 Use the [**no**] **aaa server radius dynamic-author** command to configure the switch as an AAA server to facilitate interaction with an external policy server.

Example:

```
switch#configure terminal
switch(config)#aaa server radius dynamic-author
```

You can disable the RADIUS dynamic author and the associated clients using the no form of this command.

Step 2 Use the [**no**] **client** {*ip-address* | **hostname**} [**server-key** [**0** | **7**] *string*] command to configure the IP address or the hostname of the AAA server client.

Example:

```
switch(config-locsvr-da-radius)#client 192.168.0.5 server-key cisco1
```

Use the optional server-key keyword and string argument to configure the server key at the client level. You can remove the client server using the no form of this command.

Note
Configuring the server key at the client level overrides the server key that is configured at the global level.

Step 3 Use the [**no**] **port** *port-number* command to specify the port on which a device listens to the RADIUS requests from the configured RADIUS clients.

Example:

```
switch(config-locsvr-da-radius)#port 3799
```

The port range is 1 - 65535. You can revert to the default port using the no form of this command.

Note
The default port for a packet of disconnect is 1700.

Step 4 Use the [**no**] **server-key** [**0** | **7**] *string* command to configure the global RADIUS key to be shared between a device and the RADIUS clients.

You can remove the server-key using the no form of this command.

RADIUS configuration verification commands

Use this reference to verify the RADIUS configuration.

- RADIUS configuration verification involves using show commands to display current settings and server parameters.
- Key commands include **show radius** , **show running-config radius** , **show startup-config radius** , and **show radius-server** .
- These commands help administrators confirm the status and details of RADIUS services and servers.

RADIUS configuration verification reference

To display RADIUS configuration information, perform one of the following tasks:

Table 16: RADIUS verification commands

Command	Purpose
show radius {**status** | **pending** | **pending-diff**}	Displays the RADIUS Cisco Fabric Services distribution status and other details.
show running-config radius [**all**]	Displays the RADIUS configuration in the running configuration.
show startup-config radius	Displays the RADIUS configuration in the startup configuration.
show radius-server [*hostname* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters.

RADIUS change of authorization configuration verification commands

Use this reference to verify the RADIUS change of authorization configuration.

- Verification of RADIUS Change of Authorization configuration involves using specific show and clear commands.
- These commands display or clear configuration and statistics for dot1x, AAA, and RADIUS components.
- Commands can be used to monitor both server and client RADIUS statistics.

RADIUS change of authorization verification commands

To display RADIUS Change of Authorization configuration information, perform one of the following tasks:

Table 17: RADIUS change of authorization configuration verification commands

Command	Purpose
show running-config dot1x	Displays the dot1x configuration in the running configuration.
show running-config aaa	Displays the AAA configuration in the running configuration.
show running-config radius	Displays the RADIUS configuration in the running configuration.
show aaa server radius statistics	Displays the local RADIUS server statistics.
show aaa client radius statistics {*ip address* | *hostname*}	Displays the local RADIUS client statistics.
clear aaa server radius statistics	Clears the local RADIUS server statistics.
clear aaa client radius statistics {*ip address* | *hostname*}	Clears the local RADIUS client statistics.

Monitor RADIUS servers

You can monitor the statistics that the Cisco NX-OS device maintains for RADIUS server activity.

Before you begin

Configure one or more RADIUS server hosts.

Procedure

	Command or Action	Purpose
Step 1	**show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch#show radius-server statistics 10.10.1.1`	Displays the RADIUS statistics.

Related Topics

Configure RADIUS server hosts, on page 66
Clear the RADIUS server statistics, on page 87

Clear the RADIUS server statistics

You can display the statistics that the Cisco NX-OS device maintains for RADIUS server activity.

Before you begin

Configure RADIUS servers on the Cisco NX-OS device.

Procedure

Step 1 (Optional) Use the **show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} command to displays the RADIUS server statistics on the Cisco NX-OS device.

Example:

```
switch#show radius-server statistics 10.10.1.1
```

Step 2 Use the **clear radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} command to clears the RADIUS server statistics.

Example:

```
switch#clear radius-server statistics 10.10.1.1
```

Related Topics

Configure RADIUS server hosts, on page 66

Configuration example for RADIUS

The configuration example demonstrates how to set up RADIUS authentication and accounting on NX-OS devices.

- Specifies the RADIUS server key and host information.
- Defines a RADIUS server group for authentication and accounting.
- Associates the server with the RADIUS group.

RADIUS configuration on NX-OS

The following example shows how to configure RADIUS:

- Use the **radius-server** commands to specify server keys and hosts.
- Configure the **aaa group server radius** to define server groups.

The following configuration sets up a RADIUS server and associates it with a server group for authentication and accounting:

```
radius-server key 7 "ToIkLhPpG"
radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
aaa group server radius RadServer
server 10.10.1.1
```

Configuration examples of RADIUS change of authorization

Use this reference to review configuration examples of RADIUS change of authorization.

RADIUS Change of Authorization (CoA) configuration involves setting up a RADIUS server and enabling dynamic authorization for clients.

- Defines the RADIUS server host and authentication key.
- Enables dynamic authorization for the RADIUS server.
- Specifies the client IP, VRF, and server key for dynamic authorization.

RADIUS change of authorization configuration

The following example shows how to configure RADIUS Change of Authorization:

- Configure the RADIUS server host and key.
- Enable dynamic authorization for the RADIUS server.
- Specify the client IP, VRF, and server key for dynamic authorization.

```
radius-server host 10.77.143.170 key 7 "fewhg123" authentication accounting
aaa server radius dynamic-author
    client 10.77.143.170 vrf management server-key 7 "fewhg123"
```

Where to go next

You can now configure AAA authentication methods to include the server groups.

Additional references for RADIUS

This section describes additional information related to implementing RADIUS.

Related Documents

Table 18: Related documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards

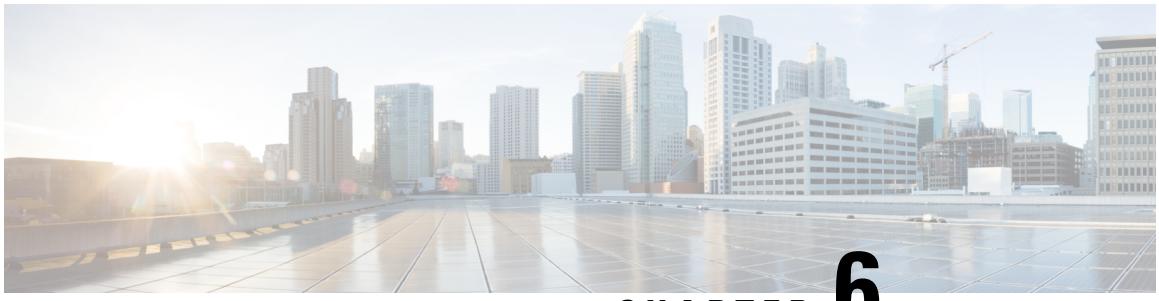
Table 19: Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

Table 20: MIBs

MIBs	MIBs Link
MIBs related to RADIUS	To locate and download supported MIBs, go to the following URL: https://example.com/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 6

Configuring TACACS+

This chapter describes how to configure the Terminal Access Controller Access Control System Plus (TACACS+) protocol on Cisco NX-OS devices.

This chapter includes the following sections:

- TACACS+, on page 91
- Prerequisites for TACACS+, on page 95
- Guidelines and limitations for TACACS+, on page 95
- Default settings for TACACS+, on page 96
- One-time password support, on page 96
- Configuring TACACS+, on page 97
- Monitor TACACS+ servers, on page 116
- Clear TACACS+ server statistics, on page 117
- TACACS+ configuration verification commands, on page 117
- Configuration examples for TACACS+, on page 118
- Where to go next, on page 119
- Additional references for TACACS+, on page 119

TACACS+

TACACS+ is a security protocol that provides centralized authentication, authorization, and accounting for users attempting Cisco NX-OS device access. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to TACACS+ server and must configure it before the configured TACACS+ features on your NX-OS device are available.

TACACS+ provides for separate authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The TACACS+ client/server protocol uses TCP (TCP port 49) for transport. NX-OS devices provide centralized authentication using the TACACS+ protocol.

Beginning with NX-OS release 10.4(3)F, SSH based authorization of X.509 certificates using TACACS+ server can be done using the **aaa authorization ssh-certificate default group** command on the Cisco Nexus 9000 Series platform switches. For configuration details, see Configure X.509 Certificate-Based SSH authorization using TACACS server, on page 114

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the NX-OS device can authorize access without authenticating.
- Uses the TCP transport protocol to send data between the AAA client and server, making reliable transfers with a connection-oriented protocol.
- Obfuscates the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. The RADIUS protocol only obfuscates passwords.

TACACS+ Operation for User Login

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco NX-OS device using TACACS+, the following actions occur:



Note TACACS+ allows an arbitrary conversation between the daemon and the user until the daemon receives enough information to authenticate the user. This action is usually done by prompting for a username and password combination, but may include prompts for other items, such as your mother's maiden name.

1. When the Cisco NX-OS device establishes a connection, it contacts the TACACS+ daemon to obtain the username and password.
2. The Cisco NX-OS device will eventually receive one of the following responses from the TACACS+ daemon:

   ACCEPT
   User authentication succeeds and service begins. If the Cisco NX-OS device requires user authorization, authorization begins.
   REJECT
   User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.
   ERROR
   An error occurred at some time during authentication either at the daemon or in the network connection between the daemon and the Cisco NX-OS device. If the Cisco NX-OS device receives an ERROR response, the Cisco NX-OS device tries to use an alternative method for authenticating the user.

   After authentication, the user also undergoes an additional authorization phase if authorization has been enabled on the Cisco NX-OS device. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.
3. If TACACS+ authorization is required, the Cisco NX-OS device again contacts the TACACS+ daemon and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access.

   Services include the following:

   - Telnet, remote login (rlogin), Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services

- Connection parameters, including the host or client IP address (IPv4 or IPv6), access list, and user timeouts

Default TACACS+ server obfuscation type and secret key

You must configure the TACACS+ secret key to authenticate the switch to the TACACS+ server. A secret key is a secret text string shared between the Cisco NX-OS device and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white spaces are not allowed). You can configure a global secret key for all TACACS+ server configurations on the Cisco NX-OS device to use.

You can override the global secret key assignment by explicitly using the **key** option when configuring an individual TACACS+ server.

Command authorization support for TACACS+ servers

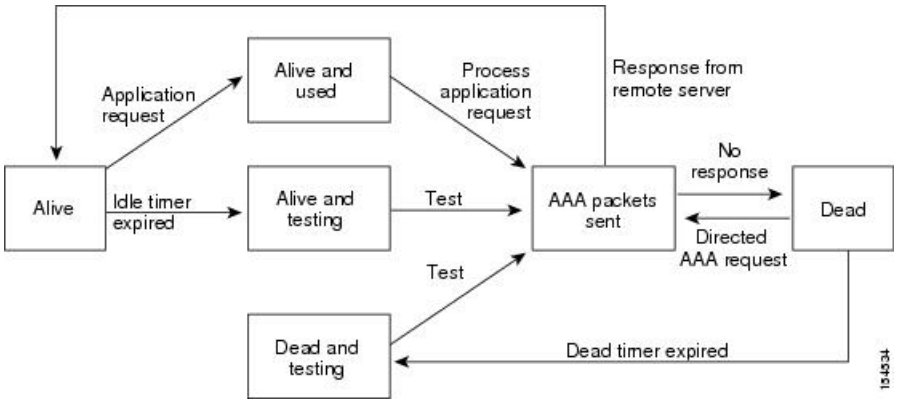
By default, command authorization is done against a local database in the Cisco NX-OS software when an authenticated user enters a command at the command-line interface (CLI). You can also verify authorized commands for authenticated users using TACACS+.

TACACS+ server monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. A Cisco NX-OS device can periodically monitor a TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests.

The Cisco NX-OS device marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. A Cisco NX-OS device periodically monitors dead TACACS+ servers and brings them to the alive state once they are responding. This process verifies that a TACACS+ server is in a working state before real AAA requests are sent its way. Whenever a TACACS+ server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco NX-OS device displays an error message that a failure is taking place before it can impact performance.

Figure 4: TACACS+ server states



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The TACACS+ server monitoring is performed by sending a test authentication request to the TACACS+ server.

Vendor-specific attributes for TACACS+

The draft standard from the Internet Engineering Task Force (IETF) describes how to communicate vendor-specific attributes (VSAs) between a network access server and a TACACS+ server. Attribute 26 is reserved for vendor-specific attributes. With VSAs, vendors can provide extended attributes intended for unique requirements and not for general use.

Cisco VSA format for TACACS+

The Cisco TACACS+ implementation supports one vendor-specific option using the format recommended in the IETF specification.

The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use TACACS+ servers for authentication on a Cisco NX-OS device, the TACACS+ protocol directs the TACACS+ server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the Cisco NX-OS software:

- Shell

  Protocol used in access-accept packets to provide user profile information.

- Accounting

  Protocol used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The Cisco NX-OS software supports the following attributes:

- roles

  Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and network-admin, the value field would be network-operator network-admin. This subattribute, which the TACACS+ server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute as supported by Cisco ACS:

  ```
  shell:roles=network-operator network-admin
  shell:roles*network-operator network-admin
  ```

Note When you specify a VSA as shell:roles* *network-operator network-admin*, this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

- accountinginfo

  Stores accounting information in addition to the attributes covered by a standard TACACS+ accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the TACACS+ client on the switch. It can be used only with the accounting protocol data units (PDUs).

Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the TACACS+ servers.
- Obtain the secret keys from the TACACS+ servers, if any.
- Ensure that the Cisco NX-OS device is configured as a TACACS+ client of the AAA servers.

Guidelines and limitations for TACACS+

Use this reference to review the guidelines and limitations for TACACS+.

TACACS+ has the following guidelines and limitations:

- TACACS+ authentication fails for usernames containing special characters on N9K-X9636C-R and N9K-X9636Q-R line cards and the N9K-C9508-FM-R fabric module.
- Beginning with Cisco NX-OS Release 10.3(1)F , TACACS+ is supported on the Cisco Nexus 9808 switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, TACACS+ is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, TACACS+ is supported on the Cisco Nexus 9804 switches,X98900CD-A and X9836DM-A line cards.
- Beginning with Cisco NX-OS release 10.4(3)F, SSH based authorization of X.509 certificates using TACACS+ server can be done using the aaa authorization ssh-certificate default group command on the Cisco Nexus 9000 Series platform switches.
- The Cisco NX-OS switches do not support custom username/password prompts. If custom prompts are provided to the switch, they will be ignored.

Configuration limitations

- You can configure a maximum of 64 TACACS+ servers on the Cisco NX-OS device.

- If a user account on the local Cisco NX-OS device shares its name with a remote user account on an AAA server, the Cisco NX-OS software applies the user roles from the local account to the remote user, not those configured on the AAA server.
- Cisco recommends that you configure the dead-time interval if more than six servers are configured in a group. If you must configure more than six servers, make sure to set the dead-time interval to a value greater than 0 and enable dead server monitoring by configuring the test username and test password.

Default settings for TACACS+

Use this reference to review the default settings for TACACS+.

The default settings for TACACS+ are:

Default settings for TACACS+ information

Table 21: Default TACACS+ parameters settings

Parameters	Default
TACACS+	Disabled
Dead timer interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test
Privilege level support for TACACS+ authorization	Disabled

One-time password support

A one-time password (OTP) is a password that is valid for a single login session or a transaction. OTPs avoid multiple disadvantages that are associated with the static passwords. OTPs are not at risk to replay attacks. If an intruder manages to record an OTP that was already used to log into a service or to conduct an operation, it cannot be misused because it is no longer valid.

OTPs are applicable only to the RADIUS and TACACS+ protocol daemons. For a RADIUS protocol daemon, you must ensure that you disable the ASCII authentication mode. For a TACACS+ protocol daemon, you must enable the ASCII authentication mode. To enable the ASCII authentication mode, use the **aaa authentication login ascii-authentication** command.

Configuring TACACS+

This section describes how to configure TACACS+ on a Cisco NX-OS device.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

TACACS+ server configuration process

Enable centralized AAA on Cisco NX-OS using TACACS+, so authentication, command authorization, and accounting are handled by external servers with secure shared keys and optional server groups.

Procedure

Step 1 Enable TACACS+.

Step 2 Establish the TACACS+ server connections to the Cisco NX-OS device.

Step 3 Configure the secret keys for the TACACS+ servers.

Step 4 If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.

Step 5 (Optional) Configure the TCP port.

Step 6 (Optional) If needed, configure periodic TACACS+ server monitoring.

Step 7 (Optional) If TACACS+ distribution is enabled, commit the TACACS+ configuration to the fabric.

Related Topics

Enable TACACS+, on page 97

Enable TACACS+

By default, the TACACS+ feature is disabled on the Cisco NX-OS device. You must explicitly enable the TACACS+ feature to access the configuration and verification commands for authentication.

Procedure

Step 1 Use the **feature tacacs+** command to enable TACACS.

Example:

```
switch#configure terminal
switch(config)#feature tacacs+
```

Step 2 Use the **exit** command to exit configuration mode.

Example:

```
switch(config)#exit
switch#
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure TACACS+ server hosts

To access a remote TACACS+ server, you must configure the IP address or the hostname for the TACACS+ server on the Cisco NX-OS device. You can configure as many as 64 TACACS+ servers.

Note By default, when you configure a TACACS+ server IP address or hostname on the Cisco NX-OS device, the TACACS+ server is added to the default TACACS+ server group. You can add the TACACS+ server to another TACACS+ server group.

Before you begin

Enable TACACS+ to allow remote authentication.

Obtain the IPv4 addresses, IPv6 addresses, or hostnames for the remote TACACS+

Procedure

Step 1 Use the **tacacs-server host** {*ipv4-address* | *ipv6-address* | *hostname*} command to specify the IPv4 or IPv6 address or hostname for a TACACS+ server.

Example:

```
switch#configure terminal
switch(config)#tacacs-server host 10.10.2.2
```

Step 2 (Optional) Use the **show tacacs+** {**pending** | **pending-diff**} command to display the TACACS+ configuration pending for distribution.

Example:

```
switch#show tacacs+ pending
```

Step 3 (Optional) Use the **tacacs+ commit** to apply the TACACS+ configuration changes in the temporary database to the running configuration.

Example:

```
switch(config)#tacacs+ commit
```

Step 4 Use the **exit** command to exit configuration mode.

Example:

```
switch(config)#exit
switch#
```

Step 5 (Optional) Use the **show tacacs-server** command to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Step 6 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable TACACS+, on page 97
Configure TACACS+ server groups, on page 101

Configure global TACACS+ keys

You can configure secret TACACS+ keys at the global level for all servers used by the Cisco NX-OS device. A secret key is a shared secret text string between the Cisco NX-OS device and the TACACS+ server hosts.

Before you begin

Enable TACACS+.

Obtain the secret key values for the remote TACACS+ servers.

Procedure

Step 1 Use the **tacacs-server key** [**0** | **6** | **7**] *key-value* command to specify a TACACS+ key for all TACACS+ server.

Example:

```
switch#configure terminal
switch(config)#tacacs-server key 0 QsEfThUkO
```

Example:

```
switch(config)#tacacs-server key 7 "fewhg"
```

You can specify that the *key-value* is in clear text format (**0**) , is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters.

By default, no secret key is configured.

Note
If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configure the shared secret for RADIUS or TACACS+, on page 53 .

Step 2 Use the **exit** command to exit configuration mode.

Example:

```
switch(config)#exit
switch#
```

Step 3 (Optional) Use the **show tacacs-server** to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Note
The secret keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted secret keys.

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable TACACS+, on page 97
AES password encryption and primary encryption keys, on page 547

Configure a key for a specific TACACS+ server

You can configure secret keys for a TACACS+ server. A secret key is a shared secret text string between the Cisco NX-OS device and the TACACS+ server host.

Before you begin

Enable TACACS+.

Obtain the secret key values for the remote TACACS+ servers.

Procedure

Step 1 Use the **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **6** | **7**] *key-value* command to specify a secret key for a specific TACACS+ server

Example:

```
switch#configure terminal
switch(config)#tacacs-server host 10.10.1.1 key 0 PlIjUhYg
switch(config)#exit
```

Example:

```
switch(config)#tacacs-server host 10.10.1.1 key 7 "fewhg"
```

You can specify that the *key-value* is in clear text format (**0**) , is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters.

This secret key is used instead of the global secret key.

Note
If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configure the shared secret for RADIUS or TACACS+, on page 53 .

Step 2 (Optional) Use the **show tacacs-server** command to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Note
The secret keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted secret keys.

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

AES password encryption and primary encryption keys, on page 547

Configure TACACS+ server groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must belong to the TACACS+ protocol. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **aaa group server tacacs+** *group-name* command to create a TACACS+ server group and enters the TACACS+ server group configuration mode for that group.

Example:

```
switch#configure terminal
switch(config)#aaa group server tacacs+ TacServer
switch(config-tacacs+)#
```

Step 2 Use the **server** {*ipv4-address* | *ipv6-address* | *hostname*} command to configure the TACACS+ server as a member of the TACACS+ server group.

Example:

```
switch(config-tacacs+)#server 10.10.2.2
switch(config-tacacs+)#exit
```

If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command.

Step 3 (Optional) Use the **show tacacs-server groups** command to display the TACACS+ server group configuration.

Example:

```
switch#show tacacs-server groups
```

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable TACACS+, on page 97
Remote AAA services, on page 24
Configure TACACS+ server hosts, on page 98
Configure the TACACS+ Dead-Time interval, on page 109

Configure the global source interface for TACACS+ server groups

You can configure a global source interface for TACACS+ server groups to use when accessing TACACS+ servers. You can also configure a different source interface for a specific TACACS+ server group. By default, the Cisco NX-OS software uses any available interface.

Procedure

Step 1 Use the **ip tacacs source-interface** *interface* command to configure the global source interface for all TACACS+ server groups configured on the device.

Example:

```
switch#configure terminal
switch(config)#ip tacacs source-interface mgmt 0
switch(config)#exit
```

Step 2 (Optional) Use the **show tacacs-server** command to display the TACACS+ server configuration information.

Example:

```
switch#show tacacs-server
```

Step 3 (Optional) Use the **copy running-config startup config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable TACACS+, on page 97
Configure TACACS+ server groups, on page 101

Allowing users to specify a TACACS+ server at login

You can configure the switch to allow the user to specify which TACACS+ server to send the authentication request by enabling the directed-request option. By default, a Cisco NX-OS device forwards an authentication request based on the default AAA authentication method. If you enable this option, the user can log in as *username* **@** *vrfname* **:** *hostname* , where *vrfname* is the VRF to use and *hostname* is the name of a configured TACACS+ server.

Note If you enable the directed-request option, the Cisco NX-OS device uses only the TACACS+ method for authentication and not the default local method.

Note User-specified logins are supported only for Telnet sessions.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **tacacs-server directed-request** command for allowing users to specify a TACACS+ server to send the authentication request when logging in.

Example:

```
switch#configure terminal
switch(config)#tacacs-server directed-request
```

By default this command is disabled.

Step 2 (Optional) Use the **show tacacs+** {**pending** | **pending-diff**} command to display pending TACACS+ configuration.

Example:

```
switch#show tacacs+ pending
```

Step 3 (Optional) Use the **tacacs+ commit** command to apply the TACACS+ configuration changes in the temporary database to the running configuration.

Example:

```
switch(config)#tacacs+ commit
switch(config)#exit
```

Step 4 (Optional) Use the **show tacacs-server directed-request** command to display the TACACS+ directed request configuration.

Example:

```
switch#show tacacs-server directed-request
```

Step 5 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable TACACS+, on page 97

Configure the timeout interval for a TACACS+ server

You can set a timeout interval that the Cisco NX-OS device waits for responses from a TACACS+ server before declaring a timeout failure. The timeout interval determines how long the Cisco NX-OS device waits for responses from a TACACS+ server before declaring a timeout failure.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **tacacs-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds* command to specify the timeout interval for a specific server. The default is the global value.

Example:

```
switch#configure terminal
switch(config)#tacacs-server host server1 timeout 10
```

Note
The timeout interval value specified for a TACACS+ server overrides the global timeout interval value specified for all TACACS+ servers.

Step 2 (Optional) Use the **show tacacs+** {**pending** | **pending-diff**} command to display the TACACS+ configuration pending for distribution.

Example:

```
switch#show tacacs+ pending
```

Step 3 (Optional) Use the **tacacs+ commit** command to apply the TACACS+ configuration changes in the temporary database to the running configuration.

Example:

```
switch(config)#tacacs+ commit
switch(config)#exit
```

Step 4 (Optional) Use the **show tacacs-server** command to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Step 5 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable TACACS+, on page 97

Configure TCP ports

You can configure another TCP port for the TACACS+ servers if there are conflicts with another application. By default, Cisco NX-OS devices use port 49 for all TACACS+ requests.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **tacacs-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **port** *tcp-port* command to specify the TCP port to use for TACACS+ messages to the server. The default TCP port is 49. The range is from 1 to 65535.

Example:

```
switch#configure terminal
switch(config)#tacacs-server host 10.10.1.1 port 2
```

Step 2 (Optional) Use the **show tacacs+** {**pending** | **pending-diff**} command to display the TACACS+ configuration pending for distribution.

Example:

```
switch#show tacacs+ distribution pending
```

Step 3 (Optional) Use the **tacacs+ commit** command to apply the TACACS+ configuration changes in the temporary database to the running configuration.

Example:

```
switch(config)#tacacs+ commit
switch(config)#exit
```

Step 4 (Optional) Use the **show tacacs-server** command to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Step 5 (Optional) Use the **copy running-config startup-config** commad to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable TACACS+, on page 97

Configure global periodic TACACS+ server monitoring

You can monitor the availability of all TACACS+ servers without having to configure the test parameters for each server individually. Any servers for which test parameters are not configured are monitored using the global level parameters.

Note Test parameters that are configured for individual servers take precedence over global test parameters.

The global configuration parameters include the username and password to use for the servers and an idle timer. The idle timer specifies the interval in which a TACACS+ server receives no requests before the Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note The test parameters are distributed across switches. If even one switch in the fabric is running an older release, the test parameters are not distributed to any switch in the fabric.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the TACACS+ database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not performed.

Note The username **test** on a Nexus switch cannot be disabled but can be changed if needed; to avoid the TACACS server being marked DEAD and ALIVE periodically, ensure the TACACS server always sends a response (Access-Accept or Access-Reject) to the **test** user request packet.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **tacacs-server test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} command to specify parameters for global server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.

Example:

```
switch#configure terminal
switch(config)#tacacs-server test username user1 password Ur2Gd2BH idle-time 3
```

Note
For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.

Step 2 Use the **tacacs-server dead-time** *minutes* command to specify the number of minutes before the Cisco NX-OS device checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 0 to 1440 minutes.

Example:

```
switch(config)#tacacs-server dead-time 5
switch(config)#exit
```

Step 3 (Optional) Use the **show tacacs-server** command to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure periodic TACACS+ server monitoring on individual servers, on page 107

Configure periodic TACACS+ server monitoring on individual servers

You can monitor the availability of individual TACACS+ servers. The configuration parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval in which a TACACS+ server receives no requests before the Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.



Note Test parameters that are configured for individual servers take precedence over global test parameters.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the TACACS+ database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not performed.

Note The test parameters are distributed across switches. The test parameters are not distributed to any switch in the fabric.

Before you begin

Enable TACACS+.

Add one or more TACACS+ server hosts.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Use the **tacacs-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} command to specify parameters for individual server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.

Example:

```
switch(config)#tacacs-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time
 3
```

Note
For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.

Step 3 Use the **tacacs-server dead-time** *minutes* command to specify the number of minutes before the Cisco NX-OS device checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 0 to 1440 minutes.

Example:

```
switch(config)#tacacs-server dead-time 5
switch(config)#exit
```

Step 4 (Optional) Use the **show tacacs-server** to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Step 5 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure TACACS+ server hosts, on page 98
Configure global periodic TACACS+ server monitoring, on page 106

Configure the TACACS+ Dead-Time interval

You can configure the dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the Cisco NX-OS device waits, after declaring a TACACS+ server is dead, before sending out a test packet to determine if the server is now alive.

Note When the dead-timer interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-timer per group.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **tacacs-server deadtime** *minutes* command to configure the global dead-time interval. The default value is 0 minutes. The range is from 1 to 1,440 minutes.

Example:

```
switch#configure terminal
switch(config)#tacacs-server deadtime 5
```

Step 2 (Optional) Use the **show tacacs+** {**pending** | **pending-diff**} command to display the pending TACACS+ configuration.

Example:

```
switch#show tacacs+ pending
```

Step 3 (Optional) Use the **tacacs+ commit** command to apply the TACACS+ configuration changes in the temporary database to the running configuration.

Example:

```
switch(config)#tacacs+ commit
switch(config)#exit
```

Step 4 (Optional) Use the **show tacacs-server** command to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Step 5 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure ASCII authentication

You can enable ASCII authentication on the TACACS+ server.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **aaa authentication login ascii-authentication** command to enable ASCII authentication. The default is disabled.

Example:

```
switch#configure terminal
switch(config)#aaa authentication login ascii-authentication
```

Step 2 (Optional) Use the **show tacacs+** {**pending** | **pending-diff**} command to display the pending TACACS+ configuration.

Example:

```
switch#show tacacs+ pending
```

Step 3 (Optional) Use the **tacacs+ commit** command to apply the TACACS+ configuration changes in the temporary database to the running configuration.

Example:

```
switch(config)#tacacs+ commit
switch(config)#exit
```

Step 4 (Optional) Use the **show tacacs-server** command to display the TACACS+ server configuration.

Example:

```
switch#show tacacs-server
```

Step 5 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure command authorization on TACACS+ servers

You can configure authorization for commands on TACACS+ servers.

Caution Command authorization disables user role-based authorization control (RBAC), including the default roles.

Note If you use a console to log in to the server, command authorization is disabled. Authorization is available for both non-console and console sessions. By default, command authorization is disabled for console sessions even if it is configured for default (non-console) sessions. You must explicitly configure a AAA group for the console to enable command authorization for console sessions.

Note By default, context-sensitive help and command tab completion show only the commands supported for a user as defined by the assigned roles. When you enable command authorization, the Cisco NX-OS software displays all commands in the context sensitive help and in tab completion, regardless of the role assigned to the user.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **aaa authorization** {**commands** | **config-commands**} {**console** | **default**} {**group** *group-list* [**local**] | **local**} command to configure the command authorization method for specific roles on a TACACS+ server.

Example:

```
switch#configure terminal
switch(config)#aaa authorization commands
default group TacGroup
Per command authorization will disable RBAC for all
users. Proceed (y/n)?
```

The **commands** keyword configures authorization sources for all EXEC commands, and the **config-commands** keyword configures authorization sources for all configuration commands.

The **console** keyword configures command authorization for a console session, and the **default** keyword configures command authorization for a non-console session.

The *group-list* argument consists of a space-delimited list of TACACS+ server group names. Servers belonging to this group are contacted for command authorization. The **local** method uses the local role-based database for authorization.

The **local** method is used only if all the configured server groups fail to respond and you have configured **local** as the fallback method. The default method is **local** .

If you have not configured a fallback method after the TACACS+ server group method, authorization fails if all server groups fail to respond.

If you press **Enter** at the confirmation prompt, the default action is *n* .

Step 2 (Optional) Use the **show tacacs+** {**pending** | **pending-diff**} command to display the pending TACACS+ configuration.

Example:

```
switch#show tacacs+ pending
```

Step 3 (Optional) Use the **tacacs+ commit** command to apply the TACACS+ configuration changes in the temporary database to the running configuration.

Example:

```
switch(config)#tacacs+ commit
switch(config)#exit
```

Step 4 (Optional) Use the **show aaa authorization** [**all**] command to display the AAA authorization configuration. The **all** keyword displays the default values.

Example:

```
switch#show aaa authorization
```

Step 5 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable TACACS+, on page 97
Testing command authorization on TACACS+ servers, on page 112

Testing command authorization on TACACS+ servers

You can test the command authorization for a user on the TACACS+ servers.

Note You must send correct commands for authorization or else the results may not be reliable.

Note The **test** command uses the default (non-console) method for authorization, not the console method.

Before you begin

Enable TACACS+.

Ensure that you have configured command authorization for the TACACS+ servers.

Procedure

Use the **test aaa authorization command-type** {**commands** | **config-commands**} **user** *username* **command** *command-string* command to test a user's authorization for a command on the TACACS+ servers.

Example:

```
switch#test aaa authorization command-type commands
user TestUser command reload
```

The **commands** keyword specifies only EXEC commands and the **config-commands** keyword specifies only configuration commands.

Note
Put double quotes (") before and after the *command-string* argument if it contains spaces.

Related Topics

Enable TACACS+, on page 97
Configure command authorization on TACACS+ servers, on page 111
Configuring User Accounts and RBAC, on page 233

Enable and disable command authorization verification

You can enable and disable command authorization verification on the command-line interface (CLI) for the default user session or for another username.

Note The commands do not execute when you enable authorization verification.

Procedure

Step 1 Use the **terminal verify-only** [**username** *username*] command to enable command authorization verification. After you enter this command, the Cisco NX-OS software indicates whether the commands you enter are authorized or not.

Example:

```
switch#terminal verify-only
```

Step 2 Use the **terminal no verify-only** [**username** *username*] command to disable command authorization verification.

Example:

```
switch#terminal no verify-only
```

Configure X.509 Certificate-Based SSH authorization using TACACS server

Beginning with Cisco NX-OS release 10.4(3)F, you can configure SSH-based authorization of x509v3-certificates using a TACAC+ server on the Cisco Nexus switches.

To configure X.509 certificate-based SSH-authorization using a TACAC+ server, follow these steps:

Procedure

Step 1 Use the **aaa authorization ssh-certificate default group** *tacacs-group-name* command to configure the default AAA authorization method for the TACAC+ servers.

Example:

```
switch#configure terminal
switch(config)#aaa authorization ssh-certificate default group tac
switch(config)#exit
switch#
```

The **ssh-certificate** keyword configures TACACS or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.

Note

- Ensure that the *tacacs-group-name* is configured under the TACACS-server configuration using the **aaa group server tacacs+** *tacacs-group-name* command.
- To support SSH certificate-based authentication, configure a crypto trustpoint and install the root CA. For more details, see the Configuring PKI, on page 179 section.

Step 2 (Optional) Use the **show aaa authorization** [**all**] command to display the AAA authorization configuration. The **all** keyword displays the default values.

Example:

```
switch#show aaa authorization
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Permitting or denying commands for users of privilege roles

As a network administrator, you can modify the privilege roles to permit users to execute specific commands or to prevent users from running those commands.

You must follow these guidelines when changing the rules of privilege roles:

- You cannot modify the priv-14 and priv-15 roles.
- You can add deny rules only to the priv-0 role.

- These commands are always permitted for the priv-0 role: **configure** , **copy** , **dir** , **enable** , **ping** , **show** , **ssh** , **telnet** , **terminal** , **traceroute** , **end** , and **exit** .

Procedure

Step 1 Use the [**no**] **role name priv-** *n* command to enable or disable a privilege role and enters role configuration mode. The *n* argument specifies the privilege level and is a number between 0 and 13.

Example:

```
switch#configure terminal
switch(config)#role name priv-5
switch(config-role)#
```

Step 2 Use the **rule** *number* {**deny** | **permit**} **command** *command-string* command to configure a command rule for users of privilege roles. These rules permit or deny users to execute specific commands. You can configure up to 256 rules for each role. The rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

Example:

```
switch(config-role)# rule 2 permit command pwd
switch(config-role)#exit
```

The *command-string* argument can contain spaces.

Note
Repeat this command for as many rules as needed.

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure user roles and rules, on page 242

Manually monitoring TACACS+ servers or groups

You can manually issue a test message to a TACACS+ server or to a server group.

Before you begin

Enable TACACS+.

Procedure

Step 1 Use the **test aaa server tacacs+** {*ipv4-address* | *ipv6-address* | *hostname*} [**vrf** *vrf-name*] *username password* command to send a test message to a TACACS+ server to confirm availability.

Example:

```
switch#test aaa server tacacs+ 10.10.1.1 user1 Ur2Gd2BH
```

Step 2 Use the **test aaa group** *group-name username password* command to send a test message to a TACACS+ server group to confirm availability.

Example:

```
switch#test aaa group TacGroup user2 As3He3CI
```

Related Topics

Configure TACACS+ server hosts, on page 98
Configure TACACS+ server groups, on page 101

Disable TACACS+

You can disable TACACS+.

Caution When you disable TACACS+, all related configurations are automatically discarded.

Procedure

Step 1 Use the **no feature tacacs+** command to disable TACACS+.

Example:

```
switch#configure terminal
switch(config)#no feature tacacs+
switch(config)#exit
```

Step 2 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Monitor TACACS+ servers

You can monitor the statistics that the Cisco NX-OS device maintains for TACACS+ server activity.

Before you begin

Configure TACACS+ servers on the Cisco NX-OS device.

Procedure

Use the **show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} command to display the TACACS+ statistics.

Example:

```
switch#show tacacs-server statistics 10.10.1.1
```

Related Topics

Configure TACACS+ server hosts, on page 98
Clear TACACS+ server statistics, on page 117

Clear TACACS+ server statistics

You can display the statistics that the Cisco NX-OS device maintains for TACACS+ server activity.

Before you begin

Configure TACACS+ servers on the Cisco NX-OS device.

Procedure

Step 1 (Optional) Use the **show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} command to display the TACACS+ server statistics on the Cisco NX-OS device.

Example:

```
switch#show tacacs-server statistics 10.10.1.1
```

Step 2 Use the **clear tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} command to clear the TACACS+ server statistics.

Example:

```
switch#clear tacacs-server statistics 10.10.1.1
```

Related Topics

Configure TACACS+ server hosts, on page 98

TACACS+ configuration verification commands

Use this reference to verify the TACACS+ configuration.

To display the TACACS+ configuration, perform one of the following tasks:

TACACS+ configuration verification commands information

Table 22: TACACS+ configuration verification commands

Command	Purpose
show tacacs+ {**status** | **pending** | **pending-diff**}	Displays the TACACS+ Cisco Fabric Services distribution status and other details.
show running-config tacacs [**all**]	Displays the TACACS+ configuration in the running configuration.
show startup-config tacacs	Displays the TACACS+ configuration in the startup configuration.
show tacacs-server [*host-name* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured TACACS+ server parameters.

Configuration examples for TACACS+

The following example shows how to configure a TACACS+ server host and server group:

```
feature tacacs+
tacacs-server key 7 "ToIkLhPpG"
tacacs-server host 10.10.2.2 key 7 "ShMoMhTl"
aaa group server tacacs+ TacServer
    server 10.10.2.2
```

The following example shows how to configure and use command authorization verification:

```
switch#terminal verify-only
switch#show interface ethernet 7/2 brief
%Success
switch#terminal no verify-only
switch#show interface ethernet 7/2 brief
--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                   Speed     Port
Interface                                                                  Ch #
--------------------------------------------------------------------------------
Eth7/2        1      eth  access down    SFP not inserted           auto(D) --
```

The following example shows how to permit all users with roles priv-5 and above to execute the **pwd** command:

```
switch#configure terminal
switch(config)#role name priv-5
switch(config-role)#rule 1 permit command pwd
```

The following example shows how to deny the **show running-config** command to all users with roles below priv-5. First, you must remove the permission to execute this command from the priv-0 role; then you must permit the command at role priv-5 so that users with roles priv-5 and above have permission to run the command.

```
switch#configure terminal
switch(config)#role name priv-0
switch(config-role)#rule 2 deny command show running-config
switch(config-role)#exit
```

```
switch(config)#role name priv-5
switch(config-role)#rule 3 permit command show running-config
switch(config-role)#exit
```

Where to go next

You can now configure AAA authentication methods to include the server groups.

Additional references for TACACS+

Use this reference to find additional references for TACACS+.

This section includes additional information related to implementing TACACS+.

Related Documents

Table 23: Related documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco NX-OS 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards

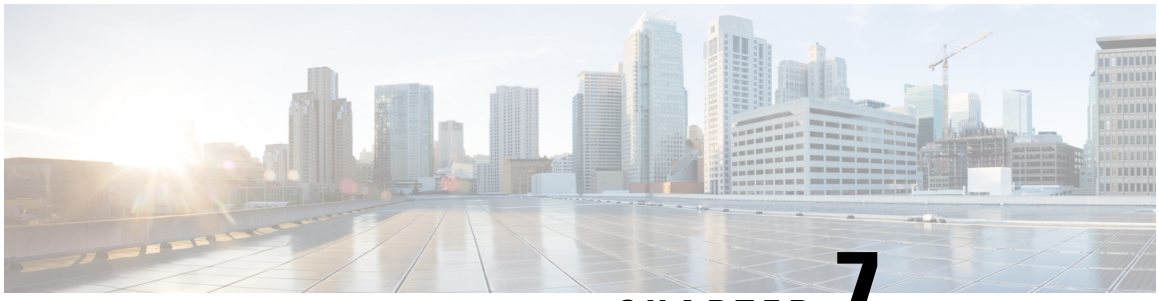
Table 24: Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

Table 25: MIBs

MIBs	MIBs Link
MIBs related to TACACS+	To locate and download supported MIBs, go to the following URL: https://example.com/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 7

Configuring LDAP

This chapter describes how to configure the Lightweight Directory Access Protocol (LDAP) on Cisco NX-OS devices and includes the following sections:

- About LDAP, on page 121
- Prerequisites for LDAP, on page 124
- Guidelines and limitations for LDAP, on page 124
- Default settings for LDAP, on page 125
- Configuring LDAP, on page 126
- Monitor LDAP servers, on page 140
- Clear LDAP server statistics, on page 141
- LDAP configuration verification commands, on page 142
- LDAP configuration examples, on page 142
- Where to go next, on page 143
- Additional references for LDAP, on page 143

About LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to an NX-OS device. An LDAP daemon (service process), which typically runs on a UNIX or Windows NT workstation, maintains LDAP services in a database. To make the LDAP features on your Cisco NX-OS device available, you must access and configure an LDAP server first.

LDAP provides for separate authentication and authorization facilities. LDAP allows a single access control server (the LDAP daemon) to provide authentication and authorization for each service independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The LDAP client/server protocol uses TCP (port 389) for transport requirements. NX-OS devices provide centralized authentication using the LDAP protocol.

LDAP authentication and authorization

Clients establish a TCP connection and authentication session with an LDAP server through a simple bind (username and password). As part of the authorization process, the LDAP server searches its database to retrieve the user profile and other information.

You can configure the bind operation to authenticate first, then authorize, or configure the operation to search first and then bind.

When you search first and bind later, you can use the distinguished name (DN) from the search result as the user DN for binding. You avoid creating a DN by adding the username (cn attribute) to the baseDN. This is useful if the user DN differs from the username and the baseDN together. For user binding, construct the bind DN using the base DN and append-with-baseDN. By default, append-with-baseDN is cn=$userid.

Note As an alternative to the bind method, you can establish LDAP authentication using the compare method, which compares the attribute values of a user entry at the server. For example, the user password attribute that can be compared for authentication. The default password attribute type is userPassword.

LDAP operations for user logins

When you attempt to log in to an NX-OS device using Password Authentication Protocol (PAP) over LDAP, these actions occur:

- When the NX-OS device establishes a connection, it contacts the LDAP daemon to obtain your username and password.
- The Cisco NX-OS device receives one of these responses from the LDAP daemon:
  - ACCEPT—User authentication succeeds and service begins. If the NX-OS device requires user authorization, authorization begins.
  - REJECT—User authentication fails. The LDAP daemon either denies further access to the user or prompts the user to retry the login sequence.
  - ERROR—An error may occur in the daemon during authentication or in the network connection between the daemon and the Cisco NX-OS device. If the NX-OS device receives an ERROR response, it uses an alternative authentication method for your account.
- After authentication, you also undergo an additional authorization phase if authorization is enabled on the NX-OS device. You must successfully complete LDAP authentication before proceeding to LDAP authorization.
- If LDAP authorization is required, the NX-OS device contacts the LDAP daemon again. The daemon then returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that determine which services you can access. It also directs the EXEC or NETWORK session for your account. Examples of these services include:
  - Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), and EXEC services
  - Connection parameters, including host IP addresses or a client IP address, such as IPv4 or IPv6 (Internet Protocol version 4 or 6), access lists, and user timeouts

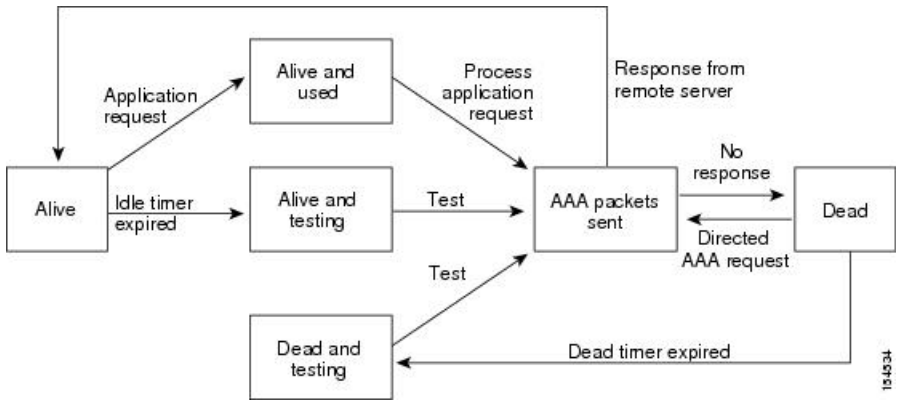
Note

- LDAP allows an arbitrary conversation between the daemon and you until the daemon receives enough information to authenticate your identity. This usually involves prompts for your username and password and may include prompts for other information.
- In LDAP, authorization can occur before authentication.

LDAP server monitoring

An unresponsive LDAP server can delay the processing of AAA requests. A Cisco NX-OS device can periodically monitor an LDAP server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive LDAP servers as dead and does not send AAA requests to any dead LDAP servers. A Cisco NX-OS device periodically monitors dead LDAP servers and brings them to the alive state once they are responding. This process verifies that an LDAP server is in a working state before real AAA requests are sent its way. Whenever an LDAP server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated, and the Cisco NX-OS device displays an error message that a failure is taking place before it can impact performance. The following figure shows the server states for LDAP server monitoring.

Figure 5: LDAP server states



Note The monitoring interval for alive servers and dead servers is different and can be configured by the user. The LDAP server monitoring is performed by sending a test authentication request to the LDAP server.

Vendor-specific attributes for LDAP

You can use the Internet Engineering Task Force (IETF) draft standard method to communicate vendor-specific attributes between your network access server and the LDAP server. Attribute 26 serves this purpose. You can use vendor-specific attributes to enable extended features that are not intended for general use.

Cisco VSA format for LDAP

The Cisco LDAP implementation supports one vendor-specific option using the format recommended in the IETF specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is an = (equal sign) for mandatory attributes, and an * (asterisk) indicates optional attributes. When you use LDAP servers for authentication on a Cisco NX-OS device, LDAP directs the LDAP server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

The following VSA protocol option is supported by the Cisco NX-OS software:

- **Shell**—Protocol used in access-accept packets to provide user profile information.

The Cisco NX-OS software supports the following attribute:

- **roles**—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space.

Virtualization support for LDAP

The Cisco NX-OS device uses virtual routing and forwarding instances (VRFs) to access the LDAP servers. For more information on VRFs, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.*

Prerequisites for LDAP

Use this reference to review prerequisites for LDAP.

LDAP has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the LDAP servers.
- Ensure that the Cisco NX-OS device is configured as an LDAP client of the AAA servers.

Guidelines and limitations for LDAP

Use this reference to review the guidelines and limitations for LDAP.

LDAP has the following guidelines and limitations:

Supported features

- You can configure a maximum of 64 LDAP servers on the Cisco NX-OS device.
- Cisco NX-OS supports only LDAP version 3.
- Cisco NX-OS supports only these LDAP servers:

- OpenLDAP
- Microsoft Active Directory

- LDAP over Secure Sockets Layer (SSL) supports only SSL version 3 and Transport Layer Security (TLS) version 1.2 .
- Beginning with Cisco NX-OS Release 10.4(3)F, LDAP over Secure Sockets Layer (SSL) supports TLS version 1.3 and 1.2 on Cisco Nexus switches. TLS v1.1 is deprecated.
- For LDAP over SSL, the LDAP client configuration must include the hostname as a subject in the LDAP server certificate.
- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on a AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

Supported platforms

- Beginning with Cisco NX-OS Release 10.3(1)F , LDAP is supported on the Cisco Nexus 9808 switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, LDAP is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, LDAP is supported on the Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.
- Beginning from Cisco NX-OS Release 10.4(2)F, the configuration replace feature is supported for LDAP on Cisco NX-OS devices.

Default settings for LDAP

This table lists the default settings for LDAP parameters.

Table 26: LDAP default parameters and value

Parameters	Default
LDAP	Disabled
LDAP authentication method	First search and then bind
LDAP authentication mechanism	Plain
Dead-time interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	60 minutes
Periodic server monitoring username	test
Periodic server monitoring password	Cisco

Configuring LDAP

This section describes how to configure LDAP on a Cisco NX-OS device.

LDAP server configuration process

Summary

You can configure LDAP servers by following this configuration process.

Workflow

You can configure LDAP servers by following this configuration process.

1. Enable LDAP.
2. Establish the LDAP server connections to the Cisco NX-OS device.
3. If needed, configure LDAP server groups with subsets of the LDAP servers for AAA authentication methods.
4. (Optional) Configure the TCP port.
5. (Optional) Configure the default AAA authorization method for the LDAP server.
6. (Optional) Configure an LDAP search map.
7. (Optional) If needed, configure periodic LDAP server monitoring.

Related Topics

Enable the LDAP feature, on page 126
Configure LDAP server hosts, on page 127
Configure the RootDN for an LDAP server, on page 128
Configure LDAP server groups, on page 129
Configure the TCP ports, on page 132
Configure LDAP search maps, on page 134
Configure periodic LDAP server monitoring, on page 135

Enable the LDAP feature

LDAP is initially disabled on NX-OS devices. You need to enable the LDAP feature to access authentication configuration and verification commands.

Follow these steps to enable the LDAP feature:

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
```

Step 2 Enable LDAP by using the **feature ldap** command. To disable it, enter the no form of the command.

Example:

```
switch(config)#feature ldap
```

Note
Disabling LDAP removes all related configurations.

Step 3 Save the running configuration to the startup configuration with **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

The LDAP feature is enabled, and authentication configuration commands are now accessible.

Related Topics

LDAP server configuration process, on page 126
Configure LDAP server hosts, on page 127
Configure the RootDN for an LDAP server, on page 128
Configure LDAP server groups, on page 129
Configure the global LDAP timeout interval, on page 131
Configure the timeout interval for an LDAP server, on page 132
Configure the TCP ports, on page 132
Configure LDAP search maps, on page 134
Configure periodic LDAP server monitoring, on page 135
Configure the LDAP dead-time interval, on page 136
Configure AAA authorization on LDAP servers, on page 137

Configure LDAP server hosts

Configure LDAP server hosts to allow Cisco NX-OS devices to access remote LDAP servers for user authentication. Use this procedure to add LDAP server IPs or hostnames and optionally enable SSL for secure communication.

Note By default, when you configure an LDAP server IP address or hostname on the Cisco NX-OS device, the LDAP server is added to the default LDAP server group. You can also add the LDAP server to another LDAP server group.

Follow these steps to configure LDAP server hosts:

Before you begin

- Enable LDAP on the device.
- Obtain the IPv4 or IPv6 addresses or the hostnames for the LDAP servers.

- If enabling SSL, ensure the LDAP server certificate is manually configured on your device.

Procedure

Step 1 Enter the global configuration mode using the **configure terminal** command..

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify the IP address or hostname for an LDAP server using the command **ldap-server host** .

- Optionally, use the `enable-ssl` keyword to encrypt communication.
- Optionally, use the `referral-disable` keyword to disable referral links.

Example:

```
switch(config)#ldap-server host 10.10.2.2 enable-ssl
```

Step 3 (Optional) Verify LDAP server configuration using the **show ldap-server** command .

Example:

```
switch#show ldap-server
```

Step 4 Save the configuration with **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

LDAP server hosts are configured and ready for authentication operations.

Related Topics

LDAP server configuration process, on page 126
Enable the LDAP feature, on page 126
Configure LDAP server groups, on page 129
Configure the RootDN for an LDAP server, on page 128
Configure LDAP server groups, on page 129
Configure periodic LDAP server monitoring, on page 135
Monitor LDAP servers, on page 140
Clear LDAP server statistics, on page 141

Configure the RootDN for an LDAP server

You can configure the root designated name (DN) for the LDAP server database. The rootDN is used to bind to the LDAP server to verify its state.

Before you begin

- Enable LDAP.

- Obtain the IPv4 or IPv6 addresses or the hostnames for the remote LDAP servers.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify the rootDN for the LDAP server database and the bind password for the root using the [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **rootDN** *root-name* [**password** *password* [**port** *tcp-port* [**timeout** *seconds*] | **timeout** *seconds*]]

Example:

```
switch(config)#ldap-server host 10.10.1.1 rootDN cn=manager,dc=acme,dc=com password Ur2Gd2BH
 timeout 60
```

Optionally specifies the TCP port to use for LDAP messages to the server. The range is from 1 to 65535, and the default TCP port is the global value or 389 if a global value is not configured. Also specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured.

Step 3 (Optional) Display the LDAP server configuration. **show ldap-server**

Example:

```
switch#show ldap-server
```

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Related Topics

LDAP server configuration process, on page 126
Enable the LDAP feature, on page 126
Configure LDAP server hosts, on page 127

Configure LDAP server groups

Specify one or more remote authentication, authorization, and accounting (AAA) servers in server groups to authenticate your users.

After you configure the servers, the system uses the servers in the order you specify. You can configure these server groups at any time, but they take effect only when you apply them to an AAA service.

Before you begin

Enable LDAP.

Procedure

Step 1 In global configuration mode, create an LDAP server group.

Example:

```
switch#configure terminal
switch(config)#aaa group server ldap LDAPServer1
```

Enters LDAP server group configuration mode.

Step 2 Configure the LDAP server as a member of the LDAP server group.

Example:

```
switch(config-ldap)#server 192.0.2.2
```

If the specified LDAP server is not found, configure it using the **ldap-server host** command and retry this command.

Step 3 (Optional) Perform LDAP authentication using the bind or compare method.

Example:

```
switch(config-ldap)#authentication compare password-attribute TyuL8r
```

The default LDAP authentication method is the bind method using first search and then bind.

Step 4 (Optional) Configure the group name in the LDAP server.

Example:

```
switch(config-ldap)#enable user-server-group
```

Users can log in with public-key authentication if their username appears as a member of this configured group in the LDAP server.

Step 5 (Optional) Allow users to log in if a user profile lists the subject-DN for the user certificate as authorized.

Example:

```
switch(config-ldap)#enable Cert-DN-match
```

Step 6 (Optional) Specify the VRF that contacts the servers in the server group.

Example:

```
switch(config-ldap)#use-vrf vrf1
```

Step 7 Exit LDAP server group configuration mode.

Example:

```
switch(config-ldap)#exit
switch(config)#
```

Step 8 (Optional) Display the LDAP server group configuration.

Example:

```
switch#show ldap-server groups
```

Step 9 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

LDAP server configuration process, on page 126
Configure LDAP server hosts, on page 127
Enable the LDAP feature, on page 126
Configure LDAP server hosts, on page 127

Configure the global LDAP timeout interval

You can set a global timeout interval that determines how long the NX-OS device waits for responses from all LDAP servers before declaring a timeout failure.

Before you begin

Enable LDAP.

Procedure

Step 1 Enter global configuration mode with the **configure terminal** command

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Set the timeout interval for LDAP servers by entering the [**no**] **ldap-server timeout** *seconds* command.

Example:

```
switch(config)#ldap-server timeout 10
```

The default timeout interval is five seconds. Specify a value from one to sixty seconds.

Step 3 (Optional) Display the LDAP server configuration by using the **show ldap-server** command.

Example:

```
switch#show ldap-server
```

Step 4 (Optional) Copy the running configuration to the startup configuration with the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Enable the LDAP feature, on page 126
Configure the timeout interval for an LDAP server, on page 132
Configure the timeout interval for an LDAP server, on page 132

Configure the timeout interval for an LDAP server

You can set a timeout interval that determines how long the NX-OS device waits for responses from an LDAP server before declaring a timeout failure.

Complete these steps to configure the timeout interval for an LDAP server.

Before you begin

Enable LDAP.

Procedure

Step 1 Enter global configuration mode with the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify the timeout interval for a specific server using the [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds* command.

Example:

```
switch(config)#ldap-server host server1 timeout 10
```

The default is the global value.

Note
The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.

Step 3 (Optional) Display the LDAP server configuration with the **show ldap-server** command.

Example:

```
switch#show ldap-server
```

Step 4 (Optional) Copy the running configuration to the startup configuration with the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure the global LDAP timeout interval, on page 131
Enable the LDAP feature, on page 126
Configure the global LDAP timeout interval, on page 131

Configure the TCP ports

Configure an alternate TCP port for LDAP servers if another application conflicts with the default port 389 on NX-OS devices.

To configure the TCP port for LDAP servers, complete these steps.

Before you begin

Enable LDAP.

Procedure

Step 1 Enter global configuration mode with the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify the TCP port to use for LDAP messages to the server using the [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **port** *tcp-port* [**timeout** *seconds*] command.

Example:

```
switch(config)#ldap-server host 10.0.0.1 port 200 timeout 5
```

- The default TCP port is 389. You can specify any port from 1 to 65,535.
- You can specify a timeout interval for the server from 1 to 60 seconds. If you do not set a value, the system uses the global timeout value, or 5 seconds if none is set.

Note
The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.

Step 3 (Optional) Display the LDAP server configuration using the **show ldap-server** command.

Example:

```
switch#show ldap-server
```

Step 4 (Optional) Copy the running configuration to the startup configuration with the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

The specified LDAP server uses the configured TCP port and timeout for LDAP communication. The system saves your configuration changes permanently after you save them.

What to do next

Test connectivity to the LDAP server using the new TCP port to ensure successful configuration.

Related Topics

LDAP server configuration process, on page 126
Enable the LDAP feature, on page 126

Configure LDAP search maps

Configure LDAP search maps on a Nexus switch. This enables LDAP queries based on attributes, filters, and search bases.

Performing this configuration allows the switch to send queries to an LDAP server for purposes such as user profiles or role definitions. The switch uses the specified criteria in search maps for these queries.

Proceed with the configuration steps for LDAP search maps.

Before you begin

You must enable LDAP.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Create a new LDAP search map using the **ldap search-map** *map-name* command.

Example:

```
switch(config)#ldap search-map map1
switch(config-ldap-search-map)#
```

Step 3 (Optional) Configure the attribute name, search filter, and base-DN for the required search operation, such as user profile, trusted certificate, certificate revocation list, certificate DN match, public key match, or user-switchgroup lookup. Use the corresponding command: [**userprofile** | **trustedCert** | **CRLLookup** | **user-certdn-match** | **user-pubkey-match** | **user-switch-bind**] **attribute-name** *attribute-name* **search-filter** *filter* **base-DN** *base-DN-name* .

Example:

```
switch(config-ldap-search-map)#userprofile attribute-name att-name search-filter
(&(objectClass=inetOrgPerson)(cn=$userid)) base-DN dc=acme,dc=com
```

- The *attribute-name* refers to the LDAP server attribute containing the Nexus role definition.
- The values specified are used to generate the LDAP query.

Step 4 (Optional) Exit LDAP search map configuration mode. Use the **exit** command.

Example:

```
switch(config-ldap-search-map)#exit
switch(config)#
```

Step 5 (Optional) Display the configured LDAP search maps using the **show ldap-search-map** command.

Example:

```
switch#show ldap-search-map
```

Step 6 (Optional) Save the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

You have configured the LDAP search maps. You can search the LDAP server when needed.

Related Topics

LDAP server configuration process, on page 126
Enable the LDAP feature, on page 126

Configure periodic LDAP server monitoring

You can monitor the availability of LDAP servers. The configuration parameters include the username, password, rootDN (used to bind and verify the state of the server), and an idle timer of the server. The idle timer is the period the LDAP server receives no requests before the NX-OS device sends a test packet. Configure this option to test servers periodically or perform a single test.

Monitor the availability of LDAP servers using these steps.

Note To protect network security, use a username that does not exist in the LDAP server database.

Before you begin

Enable LDAP.

Procedure

Step 1 Enter global configuration mode with the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Required: Specify the parameters for server monitoring using the [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **test rootDN** *root-name* [**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]] command.

Example:

```
switch(config)#ldap-server host 10.10.1.1 test rootDN root1 username user1 password Ur2Gd2BH
 idle-time 3
```

The default username is test, and the default password is Cisco. The default value for the idle timer is sixty minutes (1 hour), and the valid range is from one to one thousand four hundred forty minutes (24 hours).

Note
Use a username that does not exist in the LDAP server database.

Step 3 Check the unresponsive connection time, in minutes, between the NX-OS device and the LDAP server using the [**no**] **ldap-server deadtime** *minutes* command.

Example:

```
switch(config)#ldap-server deadtime 5
```

The default value is zero minutes, and the valid range is from one to sixty minutes (1 hour).

Step 4 (Optional) Display the LDAP server configuration using the **show ldap-server** command.

Example:

```
switch#show ldap-server
```

Step 5 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command

Example:

```
switch#copy running-config startup-config
```

You have configured the LDAP server monitoring frequency.

Related Topics

LDAP server configuration process, on page 126
Enable the LDAP feature, on page 126
Configure LDAP server hosts, on page 127

Configure the LDAP dead-time interval

You can configure the dead-time interval for all LDAP servers. This interval is the period the NX-OS device waits after declaring an LDAP server dead. After this period, the device sends a test packet to determine whether the server is alive.

Note When the dead-time interval is zero minutes (0 min), LDAP servers are not marked as dead even if they are not responding. You can configure the dead-time interval per group.

To configure the dead-time interval for the LDAP server, use these steps.

Before you begin

Enable LDAP.

Procedure

Step 1 Enters global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Configure the global dead-time interval with the [**no**] **ldap-server deadtime** *minutes* command.

Example:

```
switch(config)#ldap-server deadtime 5
```

The default value is zero minutes (0 min). You can configure a range from one to sixty minutes (1 hour, 60 min).

Step 3 (Optional) Display the LDAP server configuration using the **show ldap-server** command.

Example:

```
switch#show ldap-server
```

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

You have configured the LDAP dead-time interval.

Related Topics

Enable the LDAP feature, on page 126

Configure AAA authorization on LDAP servers

Configure the default Authentication, Authorization, and Accounting (AAA) authorization method for Lightweight Directory Access Protocol (LDAP) server.

Before you begin

Enable Lightweight Directory Access Protocol (LDAP).

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Configure the default Authentication, Authorization, and Accounting (AAA) authorization method for the Lightweight Directory Access Protocol (LDAP) servers using the **aaa authorization** {**ssh-certificate** | **ssh-publickey**} **default** {**group** *group-list* | **local**} command.

Example:

```
switch(config)#aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2
```

The **ssh-certificate** keyword enables LDAP or local authorization with certificate authentication. The **ssh-publickey** keyword enables LDAP or local authorization with SSH public key authentication. By default, authorization is local, which means the authorized commands are determined by the assigned user role.

The *group-list* argument consists of a space-delimited list of LDAP server group names. Servers that belong to this group are contacted for AAA authorization. The **local** method uses the local database for authorization.

Step 3 (Optional) Display the Authentication, Authorization, and Accounting (AAA) authorization configuration using the **show aaa authorization** [**all**] command.

Example:

```
switch#show aaa authorization
```

The **all** keyword displays the default values.

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

The default AAA authorization method for LDAP servers is now configured.

Related Topics

Enable the LDAP feature, on page 126

Configure LDAP SSH public key authorization

Enable AAA SSH public key authorization through LDAP.

To configure LDAP SSH public key authorization, use these steps.

Before you begin

Before configuring LDAP SSH public key authorization, complete these prerequisites:

- Save the user public key under the attribute defined in the
- Sign in using the user private key from an SSH client to confirm access.

Note The private key that is presented during SSH sign-in is verified with the public key which is saved in the LDAP server.

Procedure

Step 1 Save the public key of the user under the attribute defined in the **user-pubkey-match** configuration in the LDAP server.

Example:

```
switch#configure terminal
switch(config)#ldap-server host ldap.example.com rootDN
"CN=ucsadmin1,CN=Users,DC=PI-Sec-DT,DC=com" password 7 password1
switch(config)#ldap search-map Map1
switch(config-ldap-search-map)#userprofile attribute-name "description" search-filter
"(cn=$userid)" base-DN "DC=PI-Sec-DT,DC=com"
switch(config-ldap-search-map)#user-pubkey-match attribute-name "sshPublicKeys" search-filter
 "(cn=$userid)" base-DN "DC=PI-Sec-DT,DC=com"
```

```
switch(config-ldap-search-map)#exit
switch(config)#aaa group server ldap ldap1
switch(config-ldap)#server ldap.example.com
switch(config-ldap)#use-vrf management
switch(config-ldap)#ldap-search-map Map1
switch(config-ldap)#exit
switch(config)#aaa authorization ssh-publickey default group ldap1
```

The **sshPublicKeys** attribute is added in the **user-pubkey-match**.

Step 2 Use the SSH client with the user's private key to sign in to the switch management IP address.

Example:

```
ssh ldapuser@10.0.0.1 -i ldap_pub_key_test
```

The LDAP server now enables SSH public key AAA authorization.

Configure LDAP SSH certificate authorization

Authorize SSH access using LDAP server certificates and distinguished names.

Authentication, authorization, and accounting (AAA) authorization is performed through an LDAP server with a certificate, and the distinguished name (DN) of the certificate is saved in the user attribute of the LDAP server.

Before you begin

During LDAP SSH certificate authorization, the following tasks are addressed:

- Validation of the user certificate presented through the Secure Shell (SSH) client using the certificate authority certificate installed in the switch.
- By default, the **enable cert-dn-match** configuration is enabled, so cert-DN-match automatically validates the certificate using the distinguished name (DN) stored in the LDAP server.

Procedure

Step 1 Save the certificate DN in an LDAP server under any specific attribute that is mentioned in the **user-certdn-match** configuration.

Example:

```
switch#configure terminal
switch(config)#ldap-server host ldap.example.com rootDN
"CN=ucsadmin1,CN=Users,DC=PI-Sec-DT,DC=com" password 7 password1
switch(config)#ldap search-map Map24
switch(config-ldap-search-map)#userprofile attribute-name "description" search-filter
"(cn=$userid)" base-DN "DC=PI-Sec-DT,DC=com"
switch(config-ldap-search-map)#user-certdn-match attribute-name <attribute> search-filter
"(cn=$userid)" base-DN "DC=PI-Sec-DT,DC=com"
switch(config-ldap-search-map)#exit
switch(config)#aaa group server ldap ldap24
switch(config-ldap)#server ldap.example.com
switch(config-ldap)#enable cert-dn-match
switch(config-ldap)#use-vrf management
```

```
switch(config-ldap)#ldap-search-map Map24
switch(config-ldap)#exit
switch(config)#aaa authorization ssh-certificate default group ldap24
```

The format is "x509v3-sign-rsa DN /DC=com, DC=PI-Sec-DT, CN=Users, CN=username1".

Step 2 Display the details of the root certificate authority certificate installed on the switch.

Example:

```
switch#show crypto ca certificates
Trustpoint: ldap
CA certificate 0:
subject=C = IN, ST = KAR, L = BGL, O = Cisco, OU = DCBG-Cert, CN = RootCA
issuer=C = IN, ST = KAR, L = BGL, O = Cisco, OU = DCBG-Cert, CN = RootCA
serial=82EE7603BF7E74A9
notBefore=May 29 07:12:30 2023 GMT
notAfter=May 26 07:12:30 2033 GMT
SHA1 Fingerprint=00:00:5E:00:53:01:00:00:5E:00:53:02:00:00:5E:00:53:03:F7:1D
purposes: sslserver sslclient
```

Step 3 Sign in from the Secure Shell (SSH) client

Example:

```
ssh username1@10.0.0.1 -i username1.crt -vvv -oCACertificateFile=rootCA.crt
```

- In the Secure Shell (SSH) client, the input certificate contains both private key and user certificate concatenated in a single file "<user>.crt".
- The rootCA.crt file contains the root certificate authority certificate.
- The IP address is the switch management IP address.

Users can authenticate to the switch via SSH using LDAP certificates. The switch validates the certificate DN against the LDAP server attribute.

What to do next

- Verify successful Secure Shell (SSH) authentication and access control.
- Check the logs to confirm that certificate validation events occurred.

Monitor LDAP servers

You can monitor the statistics that the Cisco NX-OS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch#show ldap-server statistics 10.10.1.1`	Displays the LDAP server statistics.

Related Topics

Configure LDAP server hosts, on page 127
Clear LDAP server statistics, on page 141
Clear LDAP server statistics, on page 141

Clear LDAP server statistics

You can display the statistics that the Cisco NX-OS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Cisco NX-OS device.

Procedure

Step 1 (Optional) Display the LDAP server statistics using the **show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} command.

Example:

```
switch#show ldap-server statistics 10.10.1.1
```

Step 2 Clear the LDAP server statistics using the **clear ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} command.

Example:

```
switch#clear ldap-server statistics 10.10.1.1
```

Related Topics

Monitor LDAP servers, on page 140
Configure LDAP server hosts, on page 127
Monitor LDAP servers, on page 140

LDAP configuration verification commands

To display Lightweight Directory Access Protocol (LDAP) configuration information, perform one of the following tasks.

Table 27: Verification commands

Command	Purpose
show running-config ldap [**all**]	Displays the LDAP configuration in the running configuration.
show startup-config ldap	Displays the LDAP configuration in the startup configuration.
show ldap-server	Displays LDAP configuration information.
show ldap-server groups	Displays LDAP server group configuration information.
show ldap-server statistics {*hostname* | *ipv4-address* | *ipv6-address*}	Displays LDAP statistics.
show ldap-search-map	Displays information about the configured LDAP attribute maps.

LDAP configuration examples

These configuration examples help you set up LDAP server hosts and groups, create LDAP search maps, configure AAA authorization with certificate authentication, and validate authentication in your operating system.

Configure an LDAP server host and group:

```
feature ldap
ldap-server host 10.10.2.2 enable-ssl
aaa group server ldap LdapServer
server 10.10.2.2
exit
show ldap-server
show ldap-server groups
```

Configure an LDAP search map:

```
ldap search-map s0
userprofile attribute-name att-name search-filter "
(&(objectClass=Person)(sAMAccountName=$userid))" base-DN dc=acme,dc=com
exit
show ldap-search-map
```

Configure AAA authorization with certificate authentication for LDAP server:

```
aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2
 exit
show aaa authorization
```

Validate authentication:

```
failing
test aaa group LdapServer user <user-password>
user has failed authentication

! working
test aaa group LdapServer user <user-password>
user has been authenticated
```

Where to go next

You can now configure AAA authentication methods to include the server groups.

Additional references for LDAP

Use this reference to find additional references for LDAP.

Related Documents

Table 28: Related documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards

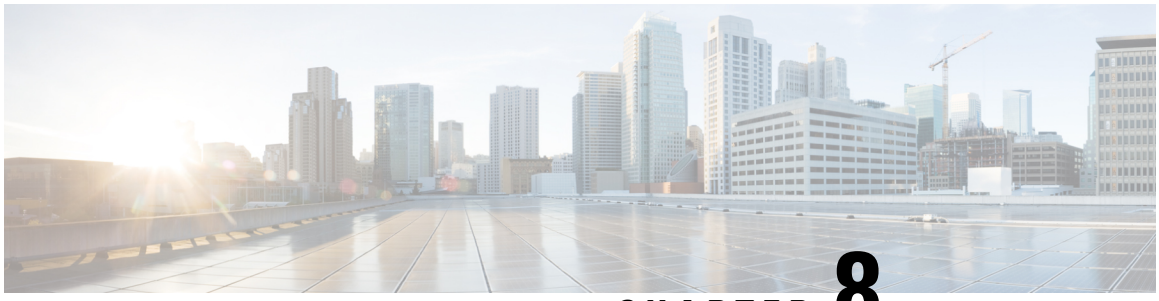
Table 29: Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

Table 30: MIBs

MIBs	MIBs Link
MIBs related to LDAP	To locate and download the supported MIBs, go to the following URL: https://example.com/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 8

Configuring SSH and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) and Telnet on Cisco NX-OS devices.

This chapter includes the following sections:

- About SSH and telnet, on page 145
- Prerequisites for SSH and telnet, on page 147
- Guidelines and limitations for SSH and telnet, on page 147
- Default settings for SSH and telnet, on page 149
- Configuring SSH , on page 149
- Configuring telnet, on page 170
- SSH and telnet configuration verification commands, on page 172
- Configuration example for SSH, on page 173
- Configuration example for SSH passwordless file copy, on page 174
- Configuration example for x.509v3 Certificate-Based SSH authentication, on page 175
- Additional references for SSH and telnet, on page 176

About SSH and telnet

This section includes information about SSH and Telnet.

SSH server

You can use the SSH server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device. SSH uses strong encryption for authentication. The SSH server in the Cisco NX-OS software can interoperate with publicly and commercially available SSH clients.

The user authentication mechanisms supported for SSH are RADIUS, TACACS+, LDAP, and the use of locally stored usernames and passwords.

SSH client

The SSH client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco NX-OS device to make a secure, encrypted connection to another Cisco NX-OS device or to any other device that runs the SSH server. This connection provides an outbound

connection that is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Cisco NX-OS software works with publicly and commercially available SSH servers.

SSH server keys

SSH requires server keys for secure communications to the Cisco NX-OS device. You can use SSH server keys for the following SSH options:

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography
- SSH version 2 using the Digital System Algrorithm (DSA)
- SSH version 2 using the Elliptic Curve Digital Signature Algorithm (ECDSA)

Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts the following types of key-pairs for use by SSH version 2:

- The **dsa** option generates the DSA key-pair for the SSH version 2 protocol.
- The **rsa** option generates the RSA key-pair for the SSH version 2 protocol.
- The **ecdsa** option generates the ECDSA key-pair for the SSH version 2 protocol.

By default, the Cisco NX-OS software generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)
- Public Key Certificate in Privacy-Enhanced Mail (PEM)



Caution If you delete all of the SSH keys, you cannot start the SSH services.

SSH authentication using digital certificates

SSH authentication on Cisco NX-OS devices provide X.509 digital certificate support for host authentication. An X.509 digital certificate is a data item that ensures the origin and integrity of a message. It contains encryption keys for secured communications and is signed by a trusted certification authority (CA) to verify the identity of the presenter. The X.509 digital certificate support provides either DSA or RSA algorithms for authentication.

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through a query or a notification. Verification of certificates is successful if the certificates are from any of the trusted CAs configured and if not revoked or expired.

You can configure your device for SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.

You can configure SSH authentication using X.509v3 certificates (RFC 6187). X.509v3 certificate-based SSH authentication uses certificates combined with a smartcard to enable two-factor authentication for Cisco device access. The SSH client is provided by Cisco partner Pragma Systems.

Telnet server

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

The Telnet server is disabled by default on the Cisco NX-OS device.

Post-quantum key exchange for SSH

Post-quantum key exchange for SSH is the key exchange phase used during SSH session establishment that

- derives per-connection keying material during SSH setup,
- uses post-quantum or hybrid key exchange algorithms to protect the shared secret, and
- negotiates a key exchange algorithm supported by both SSH peers.

Key exchange behavior

Hybrid key exchange combines a classical key exchange mechanism with a post-quantum mechanism. The resulting shared secret remains protected if at least one component remains secure.

PQC-only Module-Lattice-Based Key Encapsulation Mechanism (ML-KEM) options are available when both peers support the same ML-KEM algorithm.

SSH peers negotiate a mutually supported key exchange algorithm. If a peer does not support configured PQC or hybrid algorithms, the connection can fall back to a mutually supported classical key exchange algorithm only if classical algorithms remain enabled on both peers.

Prerequisites for SSH and telnet

Make sure that you have configured IP on a Layer 3 interface, out-of-band on the mgmt 0 interface, or inband on an Ethernet interface.

Guidelines and limitations for SSH and telnet

Use this reference to review the guidelines and limitations for SSH and telnet.

SSH and Telnet have the following configuration guidelines and limitations:

- The Cisco NX-OS software supports only SSH version 2 (SSHv2).
- Beginning with Cisco NX-OS Release 10.4(3)F, the Cisco Nexus 9000 Series switches support SSH authorization using X.509 certificates through a TACACS+ server. This feature is not supported with RADIUS.

- When you use the **no feature ssh feature** command, port 22 is not disabled . Port 22 is always open and a deny rule is pushed to deny all incoming external connections.
- Due to a Poodle vulnerability, SSLv3 is no longer supported.
- IPSG is not supported on the following:
  - The last six 40-Gb physical ports on the Cisco Nexus 9372PX, 9372TX, and 9332PQ switches
  - All 40G physical ports on the Cisco Nexus 9396PX, 9396TX, and 93128TX switches
- You can configure your device for SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.
- The SFTP server feature does not support the regular SFTP **chown** and **chgrp** commands.
- When the SFTP server is enabled, only the admin user can use SFTP to access the device.

SSH key persistence, timeout, and command display limitations

- SSH public and private keys imported into user accounts that are remotely authenticated through a AAA protocol (such as RADIUS or TACACS+) for the purpose of SSH Passwordless File Copy will not persist when the Nexus device is reloaded unless a local user account with the same name as the remote user account is configured on the device before the SSH keys are imported.
- SSH timeout period must be longer than the time of the tac-pac generation time. Otherwise, the VSH log might show %VSHD-2-VSHD_SYSLOG_EOL_ERR error. Ideally, set to 0 (infinity) before collecting tac-pac or showtech.
- Beginning with Cisco NX-OS Release 10.4(3)F, the **show running-config all** command does not display the details of the following commands:
  - no feature telnet
  - no feature nxdb
  - no feature scp-server
  - no feature sftp-server



Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

SNMP synchronization guidelines

- Beginning with Cisco NX-OS Release 10.2(2)F, a new desynchronization CLI is introduced to provide you an option to disable the user synchronization between the SNMP and the security components. For more information, refer to the *Configuring SNMP* chapter in the *System Management Configuration Guide*.

  For more information about the Cisco Nexus 9000 switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.
- When the desynchronization CLI is enabled, remote users will not be synced to SNMP database.

- The security users created using DCNM (also called as Nexus Dashboard Fabric Controller from Release 12.0.1a) will not have a corresponding SNMPv3 profile when the desynchronization CLI is enabled. When the synchronization is disabled, the users created on the security component can log in to the switch, but the switches will not be discovered by the controller, as the controller uses the SNMP configuration created for the security user to discover the switch. Furthermore, the SNMP does not recognize the security users created due to the desynchronized state of the userDB, resulting in failure to discover the switch. Therefore, to have the switches discovered by the controller, the SNMP user must be explicitly created. It is not recommended to use the desynchronization CLI along with DCNM functionality. For more information, refer to the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.

Default settings for SSH and telnet

Use this reference to review the default settings for SSH and telnet.

Table 31: Default SSH and telnet parameters

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Telnet server	Disabled
Telnet port number	23
Maximum number of SSH login attempts	3
SCP server	Disabled
SFTP server	Disabled

Configuring SSH

This section describes how to configure SSH.

Generating SSH server keys

You can generate an SSH server key based on your security requirements. The default SSH server key is an RSA key that is generated using 1024 bits.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no feature ssh**

Example:

```
switch(config)#no feature ssh
```

Disables SSH.

Step 3 **ssh key** {**dsa** [**force**] | **rsa** [*bits*[**force**]] | **ecdsa** [*bits* [**force**]]}

Example:

```
switch(config)#ssh key rsa 2048
```

Generates the SSH server key.

The *bits* argument is the number of bits used to generate the RSA key. The range is from 768 to 2048 . The default value is 1024.

You cannot specify the size of the DSA key. It is always set to 1024 bits.

Use the **force** keyword to replace an existing key.

Note
If you configure ssh key dsa, you must do the following additional configurations: ssh keytypes all and ssh kexalgos all

Step 4 **ssh rekey max-data** *max-data* **max-time** *max-time*i

Example:

```
switch(config)#ssh rekey max-data 1K max-time 1M
```

Configures the rekey parameters.

Step 5 **feature ssh**

Example:

```
switch(config)#feature ssh
```

Enables SSH.

Step 6 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 7 Verify and save the SSH server key configuration.

a) (Optional) **show ssh key** [**dsa** | **rsa** | **ecdsa**] [**md5**]

Example:

```
switch#show ssh key
```

Displays the SSH server keys.

This command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility.

b) **show running-config security all**

c) (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Specifying the SSH public keys for user accounts

You can configure an SSH public key to log in using an SSH client without being prompted for a password. You can specify the SSH public key in one of these formats:

- OpenSSH format
- IETF SECSH format

Specify the SSH public keys in IETF SECSH format

You can specify the SSH public keys in IETF SECSH format for user accounts.

Before you begin

Generate an SSH public key in IETF SCHSH format.

Procedure

Step 1 **copy** *server-file* **bootflash:** *filename*

Example:

```
switch#copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub
```

Downloads the file containing the SSH key in IETF SECSH format from a server. The server can be FTP, secure copy (SCP), secure FTP (SFTP), or TFTP.

Step 2 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 3 **username** *username* **sshkey file bootflash:** *filename*

Example:

```
switch(config)#username User1 sshkey file bootflash:secsh_file.pub
```

Configures the SSH public key in IETF SECSH format.

Step 4 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 5 (Optional) **show user-account**

Example:

```
switch#show user-account
```

Displays the user account configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Specify the SSH public keys in OpenSSH format

You can specify the SSH public keys in OpenSSH format for user accounts.

Before you begin

Generate an SSH public key in OpenSSH format.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **username** *username* **sshkey** *ssh-key*

Example:

```
switch(config)#username User1 sshkey
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPj
hOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+
XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5S4Tplx8=
```

Configures the SSH public key in OpenSSH format.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 4 (Optional) **show user-account**

Example:

```
switch#show user-account
```

Displays the user account configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure a maximum number of SSH login attempts

You can configure the maximum number of SSH login attempts. If the user exceeds the maximum number of permitted attempts, the session disconnects.

Note The total number of login attempts includes attempts through public-key authentication, certificate-based authentication, and password-based authentication. If public-key authentication is enabled, it takes priority. If only certificate-based and password-based authentication are enabled, certificate-based authentication takes priority. If you exceed the configured number of login attempts through all of these methods, a message appears indicating that too many authentication failures have occurred.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **ssh login-attempts** *number*

Example:

```
switch(config)#ssh login-attempts 5
```

Configures the maximum number of times that a user can attempt to log into an SSH session. The default maximum number of login attempts is 3. The range is from 1 to 10.

Note

The **no** form of this command removes the previous login attempts value and sets the maximum number of login attempts to the default value of 3.

Step 3 (Optional) **show running-config security all**

Example:

```
switch#show running-config security all
```

Displays the configured maximum number of SSH login attempts.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Starting SSH sessions

You can start SSH sessions using IPv4 or IPv6 to connect to remote devices from the Cisco NX-OS device.

Before you begin

Obtain the hostname for the remote device and, if needed, the username on the remote device.

Enable the SSH server on the remote device.

Procedure

Step 1 **ssh** [*username* **@**]{*ipv4-address* | *hostname*} [**vrf** *vrf-name*]

Example:

```
switch#ssh 10.10.1.1
```

Creates an SSH IPv4 session to a remote device using IPv4. The default VRF is the default VRF.

Step 2 **ssh6** [*username* **@**]{*ipv6-address* | *hostname*} [**vrf** *vrf-name*]

Example:

```
switch#ssh6 HostA
```

Creates an SSH IPv6 session to a remote device using IPv6.

Starting SSH sessions from boot mode

You can start SSH sessions from the boot mode of the Cisco NX-OS device to connect to remote devices.

Before you begin

Obtain the hostname for the remote device and, if needed, the username on the remote device.

Enable the SSH server on the remote device.

Procedure

Step 1 **ssh** [*username* **@**]*hostname*

Example:

```
switch(boot)#ssh user1@10.10.1.1
```

Creates an SSH session to a remote device from the boot mode of the Cisco NX-OS device. The default VRF is always used.

Step 2 **exit**

Example:

```
switch(boot)#exit
```

Exits boot mode.

Step 3 **copy scp:**//[*username* **@**]*hostname*/*filepath directory*

Example:

```
switch#copy scp://user1@10.10.1.1/users abc
```

Copies a file from the Cisco NX-OS device to a remote device using the Secure Copy Protocol (SCP). The default VRF is always used.

Configure SSH passwordless file copy

You can copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password. To do so, you must create an RSA or DSA identity that consists of public and private keys for authentication with SSH.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **username** *username* **keypair generate**{**rsa**[*bits* [**force**]] | **dsa** [**force**]}

Example:

```
switch(config)#username user1 keypair generate rsa 2048 force
```

Generates the SSH public and private keys and stores them in the home directory ($HOME/.ssh) of the Cisco NX-OS device for the specified user. The Cisco NX-OS device uses the keys to communicate with the SSH server on the remote machine.

The *bits* argument is the number of bits used to generate the key. The range is from 768 to 2048 . The default value is 1024.

Use the **force** keyword to replace an existing key. The SSH keys are not generated if the **force** keyword is omitted and SSH keys are already present.

Step 3 (Optional) **show username** *username* **keypair**

Example:

```
switch#show username user1 keypair
```

Displays the public key for the specified user.

Note
For security reasons, this command does not show the private key.

Step 4 Required: **username** *username* **keypair export** {**bootflash:** *filename* | **volatile:** *filename*} {**rsa**| **dsa**} [**force**]

Example:

```
switch(config)#username user1 keypair export bootflash:key_rsa rsa
```

Exports the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash or volatile directory.

Use the **force** keyword to replace an existing key. The SSH keys are not exported if the **force** keyword is omitted and SSH keys are already present.

To export the generated key pair, you are prompted to enter a passphrase that encrypts the private key. The private key is exported as the file that you specify, and the public key is exported with the same filename followed by a .pub extension. You can now copy this key pair to any Cisco NX-OS device and use SCP or SFTP to copy the public key file (*.pub) to the home directory of the server.

Note
For security reasons, this command can be executed only from global configuration mode.

Step 5 Required: **username** *username* **keypair import** {**bootflash:** *filename* | **volatile:** *filename*} {**rsa**| **dsa**} [**force**]

Example:

```
switch(config)#username user1 keypair import bootflash:key_rsa rsa
```

Imports the exported public and private keys from the specified bootflash or volatile directory to the home directory of the Cisco NX-OS device.

Use the **force** keyword to replace an existing key. The SSH keys are not imported if the **force** keyword is omitted and SSH keys are already present.

To import the generated key pair, you are prompted to enter a passphrase that decrypts the private key. The private key is imported as the file that you specify, and the public key is imported with the same filename followed by a .pub extension.

Note
For security reasons, this command can be executed only from global configuration mode.

Note

Only the users whose keys are configured on the server are able to access the server without a password.

What to do next

On the SCP or SFTP server, use the following command to append the public key stored in the *.pub file (for example, key_rsa.pub) to the authorized_keys file:

$ cat key_rsa.pub>> $HOME/.ssh/ authorized_keys

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

Configure SCP and SFTP servers

You can configure an SCP or SFTP server on the Cisco NX-OS device in order to copy files to and from a remote device. After you enable the SCP or SFTP server, you can execute an SCP or SFTP command on the remote device to copy the files to or from the Cisco NX-OS device.

Note The arcfour and blowfish cipher options are not supported for the SCP server.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **feature scp-server**

Example:

```
switch(config)#feature scp-server
```

Enables or disables the SCP server on the Cisco NX-OS device.

Step 3 Required: [**no**] **feature sftp-server**

Example:

```
switch(config)#feature sftp-server
```

Enables or disables the SFTP server on the Cisco NX-OS device.

Step 4 Required: **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 5 (Optional) **show running-config security**

Example:

```
switch#show running-config security
```

Displays the configuration status of the SCP and SFTP servers.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure x.509v3 Certificate-Based SSH authentication

You can configure SSH authentication using X.509v3 certificates.

Before you begin

Enable the SSH server on the remote device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **username** *user-id* [**password** [**0** | **5**] *password*]

Example:

```
switch(config)#username jsmith password 4Ty18Rnt
```

Configures a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters for both local and remote users. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames.

Usernames must begin with an alphanumeric character.

The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The default is **0** (clear text).

Note
If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device.

Note

If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.

Note
When the desynchronization CLI is enabled, if you create a user account, the corresponding SNMP user will not be created.

Step 3 **username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**}

Example:

```
switch(config)#username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1,
emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith"
rsa
```

Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively.

Step 4 [**no**] **crypto ca trustpoint** *trustpoint*

Example:

```
switch(config)#crypto ca trustpoint winca
switch(config-trustpoint)#
```

Configures a trustpoint.

Note
Before you delete a trustpoint using the **no** form of this command, you must first delete the CRL and CA certificate, using the **delete crl** and **delete ca-certificate** commands.

Step 5 **crypto ca authenticate** *trustpoint*

Example:

```
switch(config-trustpoint)#crypto ca authenticate winca
```

Configures a CA certificate for the trustpoint.

Note
To delete a CA certificate, enter the **delete ca-certificate** command in the trustpoint configuration mode.

Step 6 (Optional) **crypto ca crl request** *trustpoint* **bootflash:** *static-crl* **.crl**

Example:

```
switch(config-trustpoint)#crypto ca crl request winca bootflash:crllist.crl
```

This command is optional but highly recommended. Configures the certificate revocation list (CRL) for the trustpoint. The CRL file is a snapshot of the list of revoked certificates by the trustpoint. This static CRL list is manually copied to the device from the Certification Authority (CA).

Note
Static CRL is the only supported revocation check method.

Note
To delete the CRL, enter the **delete crl** command.

Step 7 Verify and save the configuration.

a) (Optional) **show crypto ca certificates**

Example:

```
switch#show crypto ca certificates
```

Displays the configured certificate chain and associated trustpoint.

b) (Optional) **show crypto ca crl** *trustpoint*

Example:

```
switch#show crypto ca crl winca
```

Displays the contents of the CRL list of the specified trustpoint.

c) (Optional) **show user-account**

Example:

```
switch#show user-account
```

Displays configured user account details.

d) (Optional) **show users**

Example:

```
switch#show users
```

Displays the users logged into the device.

e) (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configuring SSH-Cert-Authorization on TACACS servers

Beginning with Cisco NX-OS Release 10.4(3)F, the Cisco Nexus 9000 Series switches support SSH authorization using X.509 certificates through a TACACS+ server. This feature is not supported with RADIUS. This feature can be enabled using

aaa authorization ssh-certificate default group *tac-group-name* command. For more information on configuration, see
Configure AAA SSH-Cert-Authorization on TACACS servers, on page 46.

Configure legacy SSH algorithm SupportCustomizing SSH cryptographic algorithms

Cisco Nexus 9000 switches support strong algorithms by default. You can choose to remain with the default mode that enables only strong algorithms as defined by Cisco Product Security Baseline (PSB) or allow all supported algorithms. Note that these algorithms are applicable to the incoming server connections. You can also configure support for SSH key exchange algorithms, message authentication codes (MACs), key types, and ciphers.

Note MAC algorithms are negotiated and enforced only for non-AEAD ciphers (such as aes256-ctr, aes128-cbc), whereas AEAD ciphers (such as aes128-gcm@openssh.com, aes256-gcm@openssh.com, or chacha20-poly1305@openssh.com) provide built-in integrity protection that renders any client-specified MAC unused, in accordance with RFC 5647 standard SSH protocol behavior.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters the global configuration mode.

Step 2 (Optional) **ssh kexalgos**[**all** | **key-exchangealgorithm-name**]

Example:

```
switch(config)#ssh kexalgos ecdhsha2-nistp384
```

Use the **all** keyword to enable all supported KexAlgorithms which are the key exchange methods that are used to generate per-connection keys.

Note
All KexAlgorithms are enabled by default, but if you need to customize your configuration, you can specify the required algorithms and apply permit or deny rules to align with your specific security requirements.

To enable only the **ecdh-sha2-nistp384** KexAlgorithm, use the **ecdh-sha2-nistp384** keyword.

Note
Starting with Cisco NX-OS Release 10.4(2)F, you can configure any of the supported KexAlgorithms. From this release, keyword **ecdh-sha2-nistp384** is deprecated.

Supported KexAlgorithms are:

- curve25519-sha256
- curve25519-sha256@libssh.org
- diffie-hellman-group14-sha1
- diffie-hellman-group14-sha256
- diffie-hellman-group16-sha512
- ecdh-sha2-nistp256
- ecdh-sha2-nistp521

Unsupported KexAlgorithms are:

- diffie-hellman-group1-sha1
- diffie-hellman-group-exchange-sha256

Step 3 (Optional) **ssh macs [all | mac-name]**

Example:

```
switch(config)#ssh macs hmacsha2-256-etm@openssh.com
```

Enables all supported MACs which are the message authentication codes used to detect traffic modification.

Supported MACs are:

- hmac-sha1
- hmac-sha2-256
- hmac-sha2-512
- hmac-sha2-256-etm@openssh.com
- hmac-sha2-512-etm@openssh.com
- hmac-sha1-etm@openssh.com

Note
Starting with Cisco NX-OS Release 10.4(2)F, you can configure any of the supported MACs.

Step 4 (Optional) **ssh ciphers**[**all** | **cipher-name**]

Example:

```
switch(config)#ssh ciphers aes192-ctr
```

Use the **all** keyword to enable all supported ciphers to encrypt the connection.

Supported ciphers are:

- aes128-cbc
- aes192-cbc
- aes256-cbc
- aes128-ctr
- aes192-ctr
- aes256-ctr
- aes128-gcm@openssh.com
- chacha20-poly1305@openssh.com

Note
Starting with Cisco NX-OS Release 10.4(2)F, you can configure any of the supported ciphers. From this release, keyword **aes256-gcm** is deprecated.

Step 5 (Optional) **ssh keytypes [all | keytype-string]**

Example:

```
switch(config)#ssh keytypes ecdsa-sha2-nistp256
```

Enables all supported PubkeyAcceptedKeyTypes which are the public key algorithms that the server can use to authenticate itself to the client.

Supported key types are:

- ecdsa-sha2-nistp256
- ecdsa-sha2-nistp384
- ecdsa-sha2-nistp521
- ecdsa-sha2-nistp256-cert-v01@openssh.com
- ecdsa-sha2-nistp384-cert-v01@openssh.com
- ecdsa-sha2-nistp521-cert-v01@openssh.com
- ssh-dss
- rsa-sha2-256
- ssh-rsa-cert-v01@openssh.com

Configure legacy SSH algorithm SupportCustomizing SSH cryptographic algorithms continued

- ssh-rsa

  Note
  Starting with Cisco NX-OS Release 10.4(2)F, you can configure any of the supported keytypes.

Note
To enable rsa, dsa, and ecdsa key types, you need to generate corresponding SSH host keys.

Example configuration:

```
switch(config)#ssh key rsa 2048
switch(config)#ssh key dsa
switch(config)#ssh key ecdsa 256
```

Users can check the supported algorithms using **show ssh [ciphers | macs | keytypes | kexalogs | version]** commands.

```
switch# show ssh ciphers
Cipher                                          Status    FIPS
-----------------------------------------------------------------
aes128-ctr                                   permitted     yes
aes192-ctr                                      denied     yes
aes256-ctr                                   permitted     yes
aes128-cbc                                      denied     yes
aes192-cbc                                      denied     yes
aes256-cbc                                      denied     yes
aes256-gcm@openssh.com                    permitted     yes
aes128-gcm@openssh.com                     permitted     yes
chacha20-poly1305@openssh.com              permitted      no

switch# show ssh mac
MAC                                             Status    FIPS
-----------------------------------------------------------------
hmac-sha2-256-etm@openssh.com              permitted      no
```

```
hmac-sha2-512-etm@openssh.com              permitted      no
hmac-sha1-etm@openssh.com                  permitted      no
hmac-sha2-256                              permitted     yes
hmac-sha2-512                              permitted     yes
hmac-sha1                                  permitted     yes
hmac-sha1-96                             unsupported      no
hmac-md5                                 unsupported      no
hmac-md5-96                              unsupported      no
umac-64@openssh.com                      unsupported      no
umac-128@openssh.com                     unsupported      no
hmac-sha1-96-etm@openssh.com             unsupported      no
hmac-md5-etm@openssh.com                 unsupported      no
umac-64-etm@openssh.com                  unsupported      no
umac-128-etm@openssh.com                 unsupported      no


switch# show ssh keytypes
Keytype                                        Status    FIPS
-------------------------------------------------------------
ecdsa-sha2-nistp256-cert-v01@openssh.com   permitted      no
          <<Currently not suppported>>
ecdsa-sha2-nistp384-cert-v01@openssh.com   permitted      no
          <<Currently not suppported>>
ecdsa-sha2-nistp521-cert-v01@openssh.com   permitted      no
          <<Currently not suppported>>
ssh-rsa-cert-v01@openssh.com               permitted      no
ecdsa-sha2-nistp256                        permitted     yes
ecdsa-sha2-nistp384                        permitted     yes
ecdsa-sha2-nistp521                        permitted      no
rsa-sha2-256                               permitted      no
ssh-rsa                                    permitted     yes
ssh-dss                                       denied      no
ssh-ed25519                              unsupported      no
ssh-ed25519-cert-v01@openssh.com         unsupported      no
ssh-dss-cert-v01@openssh.com             unsupported      no


switch# show ssh kexalgos
KexAlgorithm                                   Status    FIPS
-------------------------------------------------------------
curve25519-sha256                          permitted      no
curve25519-sha256@libssh.org               permitted      no
ecdh-sha2-nistp256                         permitted     yes
ecdh-sha2-nistp384                         permitted     yes
ecdh-sha2-nistp521                         permitted     yes
diffie-hellman-group16-sha512              permitted     yes
diffie-hellman-group14-sha1                permitted     yes
diffie-hellman-group14-sha256              permitted      no


switch# show ssh version
CiscoSSH 1.9.29, OpenSSH_8.3p1, CiscoSSL 1.1.1t.7.2.500
```

Algorithms supported - FIPS mode enabled

The list of algorithms supported when the FIPs mode is enabled are as follows:

Table 32: Algorithms supported - FIPS mode enabled

Algorithms	Supported	Unsupported	
ciphers	• aes128-ctr • aes256-ctr • aes256-gcm@openssh.com • aes128-gcm@openssh.com • chacha20-poly1305@openssh.com	• aes192-ctr • aes128-cbc • aes192-cbc • aes256-cbc	
hmac	• hmac-sha2-256 • hmac-sha2-512 • hmac-sha1 • hmac-sha2-256-etm@openssh.com • hmac-sha2-512-etm@openssh.com • hmac-sha1-etm@openssh.com	-	
kexalgo	• ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521 • diffie-hellman-group16-sha512 • diffie-hellman-group14-sha1 • diffie-hellman-group14-sha256 • curve25519-sha256@libssh.org • curve25519-sha256	-	
	keytypes	• ecdsa-sha2-nistp256-cert-v01@openssh.com • ecdsa-sha2-nistp384-cert-v01@openssh.com • ecdsa-sha2-nistp521-cert-v01@openssh.com • ssh-rsa-cert-v01@openssh.com • rsa-sha2-256 • ssh-rsa • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384 • ecdsa-sha2-nistp521	• ed

Change the default SSH server port

Beginning with Cisco NX-OS Cisco Release 9.2(1), you can change the SSHv2 port number from the default port number 22. Encryptions used while changing the default SSH port provides you with connections that support stronger privacy and session integrity

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no feature ssh**

Example:

```
switch(config)#no feature ssh
```

Disables SSH.

Step 3 **show sockets** *local-port-range*

Example:

```
switch#show sockets local-port-range
Kstack local port range (15001 - 22002)
Netstack local port range (22003 - 65535)
```

Displays the available port range.

Step 4 **ssh port** *local-port*

Example:

```
switch(config)#ssh port 58003
```

Configures the port.

Note
When you upgrade from an earlier release to Release 9.3(1) or later releases, ensure that features with user-defined SSH port, are within the following range:

- For Release 9.3(1) and Release 9.3(2): Kstack local port range is from 15001 to 58000, netstack local port range is from 58001 - 63535, and nat port range is from 63536 to 65535
- From Release 9.3(3): Kstack local port range is from 15001 to 58000, netstack local port range is from 58001 to 60535, and nat port range is from 60536 to 65535

Step 5 **feature ssh**

Example:

```
switch(config)#feature ssh
```

Enables SSH.

Step 6 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 7 (Optional) **show running-config security all**

Example:

```
switch#show running-config security all
```

Displays the security configuration.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Clear SSH hosts

When you download a file from a server using SCP or SFTP, or when you start an SSH session from this device to a remote host, you establish a trusted SSH relationship with that server. You can clear the list of trusted SSH servers for your user account.

Procedure

clear ssh hosts

Example:

```
switch#clear ssh hosts
```

Clears the SSH host sessions and the known host file.

Disable the SSH server

By default, the SSH server is enabled on the Cisco NX-OS device. You can disable the SSH server to prevent SSH access to the switch.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no feature ssh**

Example:

```
switch(config)#no feature ssh
```

Disables SSH.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 4 (Optional) **show ssh server**

Example:

```
switch#show ssh server
```

Displays the SSH server configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Delete SSH server keys

You can delete SSH server keys on the Cisco NX-OS device after you disable the SSH server.

Note To reenable SSH, you must first generate an SSH server key.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no feature ssh**

Example:

```
switch(config)#no feature ssh
```

Disables SSH.

Step 3 **no ssh key** [**dsa** | **rsa** | **ecdsa**]

Example:

```
switch(config)#no ssh key rsa
```

Deletes the SSH server key.

The default is to delete all the SSH keys.

Step 4 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 5 (Optional) **show ssh key**

Example:

```
switch#show ssh key
```

Displays the SSH server key configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Generating SSH server keys, on page 149

Clear SSH sessions

You can clear SSH sessions from the Cisco NX-OS device.

Procedure

Step 1 **show users**

Example:

```
switch#show users
```

Displays user session information.

Step 2 **clear line** *vty-line*

Example:

```
switch#clear line pts/12
```

Clears a user SSH session.

Configuring telnet

This section describes how to configure Telnet on the Cisco NX-OS device.

Enable the telnet server

You can enable the Telnet server on the Cisco NX-OS device. By default, the Telnet server is disabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **feature telnet**

Example:

```
switch(config)#feature telnet
```

Enables the Telnet server. The default is disabled.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 4 (Optional) **show telnet server**

Example:

```
switch#show telnet server
```

Displays the Telnet server configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Starting telnet sessions to remote devices

You can start Telnet sessions to connect to remote devices from the Cisco NX-OS device. You can start Telnet sessions using either IPv4 or IPv6.

Before you begin

Obtain the hostname or IP address for the remote device and, if needed, the username on the remote device.

Enable the Telnet server on the Cisco NX-OS device.

Enable the Telnet server on the remote device.

Procedure

Step 1 **telnet** {*ipv4-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]

Example:

```
switch#telnet 10.10.1.1
```

Starts a Telnet session to a remote device using IPv4. The default port number is 23. The range is from 1 to 65535. The default VRF is the default VRF.

Step 2 **telnet6** {*ipv6-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]

Example:

```
switch#telnet6 2001:0DB8::ABCD:1 vrf management
```

Starts a Telnet session to a remote device using IPv6. The default port number is 23. The range is from 1 to 65535. The default VRF is the default VRF.

Related Topics

Enable the telnet server, on page 170

Clear telnet sessions

You can clear Telnet sessions from the Cisco NX-OS device.

Before you begin

Enable the Telnet server on the Cisco NX-OS device.

Procedure

Step 1 **show users**

Example:

```
switch#show users
```

Displays user session information.

Step 2 **clear line** *vty-line*

Example:

```
switch#clear line pts/12
```

Clears a user Telnet session.

SSH and telnet configuration verification commands

Use this reference to verify the SSH and telnet configuration.

Configuration information configuration example

To display the SSH and Telnet configuration information, perform one of the following tasks:

SSH and telnet configuration verification commands

Table 33: SSH and telnet configuration verification commands

Command	Purpose
show ssh key [**dsa** | **rsa**] [**md5**]	Displays the SSH server keys. For Cisco NX-OS Release 7.0(3)I4(6) and 7.0(3)I6(1) and any later releases, this command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility.
show running-config security [**all**]	Displays the SSH and user account configuration in the running configuration. The **all** keyword displays the default values for the SSH and user accounts.
show ssh server	Displays the SSH server configuration.
show telnet server	Displays the Telnet server configuration.
show username *username* **keypair**	Displays the public key for the specified user.
show user-account	Displays configured user account details.
show users	Displays the users logged into the device.
show crypto ca certificates	Displays the configured CA certificate and associated trustpoint for X.509v3 certificate-based SSH authentication.
show crypto ca crl *trustpoint*	Displays the contents of the CRL list of the specified trustpoint for X.509v3 certificate-based SSH authentication.

Configuration example for SSH

The following example shows how to configure SSH with an OpenSSH key:

Procedure

Step 1 Disable the SSH server.

Example:

```
switch#configure terminal
switch(config)#no feature ssh
```

Step 2 Generate an SSH server key.

Example:

```
switch(config)#ssh key rsa
generating rsa key(1024 bits)......
generated rsa key
```

Step 3 Enable the SSH server.

Example:

```
switch(config)#feature ssh
```

Step 4 Display the SSH server key.

Example:

```
switch#show ssh key
could not retrieve dsa key information
**************************************
rsa Keys generated:Tue Mar 14 13:13:47 2017

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDh4+DZboQJbJt10nJhgKBYL5lOlhsFM2oZRi9+JqEU
GA44I9ej+E5NIRZ1x8ohIt6Vx9Et5csO7Pw72rjUwR3UPmuAm79k7I/SyLGEP3WUL7sqbLvNF5GqKXph
oqMT075WUdbGWphorA2g0tTObRrFIQBJVQ0SSBh3oEaaALqYUQ==

bitcount:1024
fingerprint:
SHA256:V6KAeLAiKRRUPBZm1Yq3rl6JW7Eo7vhLi6CXYxnD/+Y
**************************************
**************************************


switch#show ssh key
rsa Keys generated:Sat Sep 29 00:10:39 2013

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAvWhEBsF55oaPHNDBnpXOTw6+/OdHoLJZKr
+MZm99n2U0ChzZG4svRWmHuJY4PeDWl0e5yE3g3EO3pjDDmt923siNiv5aSga60K36lr39
HmXL6VgpRVn1XQFiBwn4na+H1d3Q0hDt+uWEA0tka2uOtXlDhliEmn4HVXOjGhFhoNE=

bitcount:1024
fingerprint:
00:00:5E:00:53:01:00:00:5E:00:53:02:15:5d:9a:df
**************************************
could not retrieve dsa key information
**************************************
```

Step 5 Specify the SSH public key in OpenSSH format.

Example:

```
switch(config)#username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50r
v7gsEPjhOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQ
W3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5
4Tplx8=
```

Step 6 Save the configuration.

Example:

```
switch#copy running-config startup-config
```

Configuration example for SSH passwordless file copy

The following example shows how to copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password:

Procedure

Step 1 Generate the SSH public and private keys and store them in the home directory of the Cisco NX-OS device for the specified user.

Example:

```
switch#configure terminal
switch(config)#username admin keypair generate rsa
generating rsa key(1024 bits)......
generated rsa key
```

Step 2 Display the public key for the specified user.

Example:

```
switch#show username admin keypair
**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
00:00:5E:00:53:01:00:00:5E:00:53:02:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
```

Step 3 Export the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash directory.

Example:

```
switch(config)#username admin keypair export bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# dir
.
.
.
        951     Jul 09 11:13:59 2013  key_rsa
        221     Jul 09 11:14:00 2013  key_rsa.pub
.
.
```

Step 4 After copying these two files to another Cisco NX-OS device using the **copy scp** or **copy sftp** command, import them to the home directory of the Cisco NX-OS device.

Example:

```
switch(config)#username admin keypair import bootflash:key_rsa rsa
Enter Passphrase:
switch#show username admin keypair
**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZE1TfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
00:00:5E:00:53:01:00:00:5E:00:53:02:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
switch(config)#
```

Step 5 On the SCP or SFTP server, append the public key stored in key_rsa.pub to the authorized_keys file.

Example:

```
$ cat key_rsa.pub>> $HOME/.ssh/ authorized_keys
```

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

Step 6 (Optional) Repeat this procedure for the DSA keys.

Configuration example for x.509v3 Certificate-Based SSH authentication

The following example shows how to configure SSH authentication using X.509v3 certificates:

Configuration example for x.509v3 Certificate-Based SSH authentication information

Note Beginning with Cisco NX-OS Release 10.4(3)F, the Cisco Nexus 9000 Series switches support SSH authorization using X.509 certificates through a TACACS+ server. This feature is not supported with RADIUS.

Configuration example for x.509v3 Certificate-Based SSH authentication information command examples

```
configure terminal
username jsmith password 4Ty18Rnt
username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1,
emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith"
rsa
crypto ca trustpoint tp1
crypto ca authenticate tp1
crypto ca crl request tp1 bootflash:crl1.crl

show crypto ca certificates
Trustpoint: tp1
CA certificate 0:
subject= /CN=SecDevCA
issuer= /CN=SecDevCA
serial=01AB02CD03EF04GH05IJ06KL07MN
notBefore=Jun 29 12:36:26 2016 GMT
notAfter=Jun 29 12:46:23 2021 GMT
SHA1 Fingerprint=00:00:5E:00:53:01:00:00:5E:00:53:02:00:00:5E:00:53:03:53:AF
purposes: sslserver sslclient

show crypto ca crl tp1
Trustpoint: tp1 CRL: Certificate Revocation List (CRL):
    Version 2 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: /CN=SecDevCA
    Last Update: Aug 8 20:03:15 2016 GMT
    Next Update: Aug 16 08:23:15 2016 GMT
    CRL extensions:
        X509v3 Authority Key Identifier:
            keyid:00:00:5E:00:53:04:00:00:5E:00:53:05:00:00:5E:00:53:06:FF:2A

show user-account
user:user1
        this user account has no expiry date
        roles:network-operator
        ssh cert DN : /C = US, ST = New York, L = Metropolis, O = cisco , OU = csg, CN =
user1; Algo: x509v3-sign-rsa

show users
NAME      LINE        TIME          IDLE      PID         COMMENT
user1     pts/1       Jul 27 18:43  00:03     18796      (10.10.10.1)   session=ssh
```

Additional references for SSH and telnet

Use this reference to find additional references for SSH and telnet.

This section describes additional information related to implementing SSH and Telnet.

Related Documents

Table 34: Related documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

RFCs

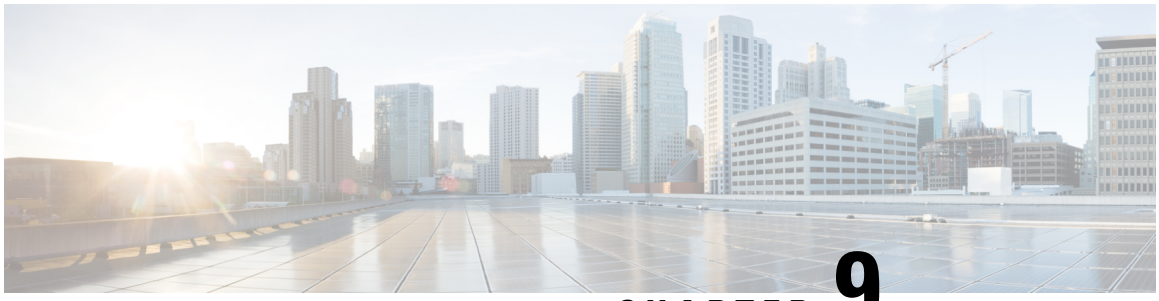
Table 35: Rfcs

RFCs	Title
RFC 6187	*X.509v3 Certificates for Secure Shell Authentication*

MIBs

Table 36: MIBs

MIBs	MIBs Link
MIBs related to SSH and Telnet	To locate and download supported MIBs, go to the following URL: https://example.com/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 9

Configuring PKI

This chapter describes the Public Key Infrastructure (PKI) support on the Cisco NX-OS device. PKI allows the device to obtain and use digital certificates for secure communication in the network and provides manageability and scalability for Secure Shell (SSH).

This chapter includes the following sections:

- About PKI, on page 179
- Guidelines and limitations for PKI, on page 188
- Default settings for PKI, on page 189
- Configuring CAs and digital certificates, on page 190
- PKI configuration verification commands, on page 210
- Configuration examples for PKI, on page 211
- Additional references for PKI, on page 231

About PKI

Public Key Infrastructure (PKI) is a framework that manages digital certificates and public-key encryption to secure communications and authenticate identities.

- PKI provides mechanisms for issuing, managing, and revoking digital certificates.
- It involves Certificate Authorities (CAs) that validate and sign certificates.
- PKI supports secure data exchange and identity verification in network environments.

CAs and digital certificates

Certificate Authorities (CAs) are trusted entities that manage certificate requests and issue digital certificates to participating entities, providing centralized key management and authentication in public key cryptography.

- CAs issue digital certificates to hosts, network devices, or users.
- Digital certificates contain identifying information and the entity's public key.
- CAs are trusted third parties that validate identities and sign certificates.

How CAs and digital certificates work

Certificate Authorities (CAs) manage certificate requests and issue certificates to participating entities such as hosts, network devices, or users. The CAs provide centralized key management for the participating entities.

- Digital signatures, based on public key cryptography, digitally authenticate devices and individual users.
- Each device or user has a key pair: a private key (kept secret) and a public key (shared).
- Anything encrypted with one key can be decrypted with the other.
- A signature is formed when data is encrypted with a sender's private key; the receiver verifies it using the sender's public key.
- This process relies on the receiver having a copy of the sender's public key and being certain of its authenticity.
- Digital certificates link the digital signature to the sender and contain identifying information and the public key.
- The CA that signs the certificate is a trusted third party for validating identities and creating certificates.
- To validate the CA's signature, the receiver must know the CA's public key, which is typically pre-installed in web browsers or handled out of band.

Trust model, trust points, and identity CAs

The trust model in PKI is hierarchical, involving multiple configurable trusted Certificate Authorities (CAs). Trust points and identity CAs are key components in this model.

- A trust point is the information about a trusted CA configured on a device, including the CA certificate or certificate chain and revocation checking information.
- The process of securely obtaining and storing a trusted CA's root certificate or certificate chain is called *CA authentication* .
- An *identity CA* is a trust point that issues identity certificates to devices for associating with key pairs.

Supporting reference information for trust model, trust points, and identity CAs

The PKI trust model allows each device to be configured with a list of trusted CAs. During security protocol exchanges, a peer certificate can be authenticated if it was issued by one of these trusted CAs. The device locally stores the self-signed root certificate of the trusted CA or the certificate chain for a subordinate CA.

Trust point and identity CA usage

For example, when a device enrolls with a trust point, it obtains an identity certificate from the identity CA. This certificate is then associated with a key pair, enabling secure authentication and communication within the network.

CA Certificate Hierarchy

The CA Certificate Hierarchy is a structure that organizes multiple trusted Certificate Authorities (CAs) into a chain or bundle, enabling secure services by validating certificates through a sequence of intermediate and root CAs.

- Multiple trusted CAs are typically installed as a bundle on all hosts.
- NX-OS PKI infrastructure supports importing certificate chains, but only one chain can be installed at a time.
- Managing several CA chains requires a facility to download CA bundles containing multiple intermediate and root CAs.

CA Certificate Hierarchy Reference Information

CA Certificate Hierarchy involves the installation and management of multiple trusted CAs to ensure secure communication between hosts.

Example of CA Certificate Hierarchy

For instance, when deploying secure services, an administrator may need to install a CA bundle that includes several intermediate and root CAs to ensure all certificates are trusted across the network.

Importing CA Bundle

The import CA bundle process in Cisco NX-OS enables the installation of multiple CA certificates under trustpoint labels using the pkcs7 file format.

- The **crypto CA trustpoint** command binds CA certificates, CRLs, identity certificates, and key pairs to a named label.
- All related files are stored in the NX-OS certstore directory and tagged with the trustpoint label.
- The import command supports the pkcs7 option to provide a CA bundle file in pkcs7 format.

Reference Information for Importing CA Bundle

The **crypto CA trustpoint** command is used to associate CA certificates, CRLs, identity certificates, and key pairs with a specific label. These files are stored in the NX-OS certstore directory and are identified by the trustpoint label.

- SSL applications can access CA certificates by specifying the NX-OS certstore as the CA path during SSL initialization, without needing to know the trustpoint label.
- To bind to an identity certificate, clients must use the trustpoint label as the binding point.

1. The import pkcs command is enhanced to install CA certificates under a trustpoint label.
2. The pkcs7 option allows the import of a CA bundle file in pkcs7 format.
3. Each CA chain in the bundle is installed under its own label, formed by appending an index to the main trustpoint label.

Once installed, there is no logical binding of all CA chains to a bundle.

Table 37: Comparison Table Title

Attributes	Trustpoint Label	CA Bundle (pkcs7)
File Storage	Stored in certstore under label	Stored in certstore, each chain under indexed label
Binding	Label used for binding	No logical binding of all chains to bundle

Note TIP: Use the pkcs7 option to efficiently import multiple CA certificates as a bundle.

Example: Importing a CA Bundle Using pkcs7

For example, to import a CA bundle in pkcs7 format, use the import command with the pkcs7 option. Each CA chain will be installed under a unique label derived from the main trustpoint label.

Import the CA Certificate Bundle in PKCS7 Format

To support the import of the CA certificate bundle which consists of multiple independent certificate chains, the option of 'pkcs7' is introduced in the crypto import command.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto ca import** *baselabel* **pksc7** *uri0* **force**

Example:

```
switch(config)# crypto ca import corp-ca-bundle pkcs7 bootflash:corp-ca-bundle.p7b force
```

There are two input arguments in the command. The source file which is the ca bundle file is given in the *uri0* , the input file has to be in pkcs7 format indicating that it is a cabundle file.

Multiple certificate chains will be extracted out of the cabundle. The command will generate multiple trustpoints with ca certificate chain attached to each one. Import command generates two configurations which are global CA bundle configuration and CA bundle sub-configuration with each trustpoint generated.

The force option removes the CA bundle and related trustpoint configurations, imports a new CA bundle with the same bundle name, and generates fresh trustpoint configurations related to the cabundle.

Step 3 **crypto ca cabundle** *bundle-name*

Example:

```
switch(config)# crypto ca cabundle corp-ca-bundle
switch(config)# exit
```

The bundle-name is same as baselabel for import case. You can use the **no** form of this command to delete the, CA bundle, trustpoints, and related certificate chains.

After importing CA bundle under a particular baselabel name and generating all the trustpoints, if a user try to execute the import command again under the same baselabel name, it will throw error saying CA bundle already exists. The user can use force option to modify the existing CA bundle.

Maximum number of Cabundles supported is 20.

Step 4 **exit**

Example:

```
switch(config)# exit
switch#
```

Exits configuration mode.

Step 5 (Optional) **show crypto ca certificates**

Example:

```
switch# show crypto ca certificates
```

Displays the CA certificates.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

RSA key pairs and identity certificates

RSA key pairs and identity certificates are fundamental components for device authentication and secure communication with certificate authorities (CAs).

- Each device can generate one or more RSA key pairs, each associated with a trust point CA for enrollment.
- Only one identity certificate and one key pair are allowed per trust point CA.
- The subject name in the identity certificate is the device's fully qualified domain name (FQDN).

Relationship between trust points, RSA key pairs, and identity certificates

Trust points, RSA key pairs, and identity certificates are interrelated to enable secure device authentication and application trust.

- A trust point corresponds to a specific CA that the device trusts for peer certificate verification for any application (such as SSH).
- A device can have many trust points, and all applications on the device can trust a peer certificate issued by any of the trust point CAs.

- A trust point is not restricted to a specific application.
- A device enrolls with the CA that corresponds to the trust point to obtain an identity certificate. Devices can enroll with multiple trust points, obtaining separate identity certificates from each. The certificates are used by applications depending on the purposes specified in the certificate by the issuing CA, stored as certificate extensions.
- When enrolling with a trust point, an RSA key pair must be generated and associated to the trust point before generating the enrollment request. The association between the trust point, key pair, and identity certificate remains valid until explicitly removed by deleting the certificate, key pair, or trust point.
- You can generate one or more RSA key pairs on a device, and each can be associated to one or more trust points. However, only one key pair can be associated to a trust point, allowing only one identity certificate from a CA.
- If a device obtains multiple identity certificates (each from a distinct CA), the certificate that an application selects for a security protocol exchange is application specific.
- It is not necessary to designate one or more trust points for an application. Any application can use any certificate issued by any trust point as long as the certificate purpose satisfies the application requirements.
- Only one identity certificate from a trust point and one key pair associated to a trust point are needed. A CA certifies a given identity (or name) only once and does not issue multiple certificates with the same name. If multiple identity certificates are needed for a CA and the CA allows it, another trust point must be defined for the same CA, with another key pair associated and certified.

The software allows you to generate RSA key pairs with a configurable key size (modulus). The default key size is 512. You can also configure an RSA key-pair label, with the default label being the device's fully qualified domain name (FQDN).

RSA key pair and identity certificate enrollment

For example, a device generates an RSA key pair labeled with its FQDN, associates it with a trust point for a specific CA, and enrolls to obtain an identity certificate. This certificate is then used by applications such as SSH for secure communication.

Multiple trusted CA support

Multiple Trusted CA Support allows a device to trust multiple Certificate Authorities (CAs) by configuring multiple trust points, each associated with a distinct CA.

- Enables configuration of multiple trust points on a device.
- Allows association of each trust point with a different CA.
- Facilitates certificate verification from peers using any of the trusted CAs.

Reference information for multiple trusted CA support

The device can be configured with multiple trusted CAs, eliminating the need to enroll with the specific CA that issued a peer's certificate.

Multiple trusted CA support

For example, if a device is configured with three trusted CAs, it can verify certificates from peers issued by any of those CAs, improving interoperability in environments with multiple certificate authorities.

PKI enrollment support

PKI enrollment is the process by which a device obtains an identity certificate from a certificate authority (CA) for secure communications.

- Enrollment involves generating an RSA key pair on the device.
- The device creates a certificate request and forwards it to the CA.
- The issued certificate is received from the CA and stored in nonvolatile memory on the device.

Supporting reference information for PKI enrollment support

PKI enrollment is essential for enabling secure device authentication and encrypted communications in network environments.

- Enrollment is required for applications like SSH that depend on device identity certificates.

1. Generate RSA key pair on the device.
2. Generate and forward certificate request to the CA.
3. Receive signed certificate from the CA.
4. Store the certificate in nonvolatile storage (bootflash).



Note The CA administrator may be required to manually approve the enrollment request at the CA server when the request is received by the CA.

PKI enrollment support

For example, when a device is configured for SSH, it initiates PKI enrollment by generating a key pair, sending a certificate request to the CA, and storing the signed certificate for secure SSH sessions.

Manual enrollment using Cut-and-Paste

Manual enrollment using cut-and-paste is a process where certificate requests and issued certificates are transferred between a device and a CA by copying and pasting base64-encoded text.

- Supported by the Cisco NX-OS software.
- Involves manual transfer of certificate requests and certificates.

- Uses base64-encoded text for communication with the CA.

Steps for manual Cut-and-Paste enrollment

To manually enroll a certificate using cut-and-paste, perform the following steps:

1. Create an enrollment certificate request, which the Cisco NX-OS device displays in base64-encoded text form.
2. Cut and paste the encoded certificate request text in an e-mail or in a web form and send it to the CA.
3. Receive the issued certificate (in base64-encoded text form) from the CA in an e-mail or in a web browser download.
4. Cut and paste the issued certificate to the device using the certificate import facility.

Multiple RSA key pair and identity CA support

- Multiple identity CAs allow a device to enroll with more than one trust point, resulting in multiple identity certificates from distinct CAs.
- The device can participate in SSH and other applications with many peers using certificates issued by CAs acceptable to those peers.
- The multiple RSA key-pair feature enables the device to maintain a distinct key pair for each CA, matching policy requirements such as key length without conflicts.

Reference information for multiple RSA key pair and identity CA support

With this feature, the device can generate multiple RSA key pairs and associate each key pair with a distinct trust point. When enrolling with a trust point, the associated key pair is used to construct the certificate request.

Peer certificate verification

Use this reference to verify the Peer certificate.

Peer certificate verification is the process by which a device checks the authenticity and validity of certificates received from peers during secure communications.

- Ensures the peer certificate is issued by a locally trusted Certificate Authority (CA).
- Checks that the peer certificate is valid and not expired at the time of verification.
- Verifies that the peer certificate has not been revoked by the issuing CA.

Peer certificate verification process and methods

The PKI support on a device can verify peer certificates. The software verifies certificates received from peers during security exchanges for applications, such as SSH. The applications verify the validity of the peer certificates. The software performs several steps when verifying peer certificates, including checking the issuing CA, certificate validity, and revocation status.

- Verifies the certificate is issued by a trusted CA.

- Checks the certificate validity period.
- Confirms the certificate is not revoked.

For revocation checking, the software supports the certificate revocation list (CRL). A trust point CA can use this method to verify that the peer certificate has not been revoked.

Certificate revocation checking

Certificate revocation checking is a process that determines whether a CA certificate is still valid or has been revoked.

- The software can check the revocation status of CA certificates.
- Applications can use different revocation checking mechanisms in a specified order.
- The available methods are CRL, **NDcPP: OCSP for Syslog** , none, or a combination of these methods.

CRL support

Certificate revocation lists (CRLs) are maintained by certificate authorities (CAs) to provide information about certificates that have been revoked before their expiration dates.

- CRLs are published by CAs in a repository and referenced in issued certificates.
- Clients can download and cache CRLs locally to verify the revocation status of peer certificates.
- The software allows manual configuration and caching of CRLs for trust points in the device bootflash (cert-store).

How CRLs are used for certificate revocation checking

CRLs enable clients to determine if a certificate has been revoked by consulting the latest list published by the issuing CA.

- Clients obtain the latest CRL from the issuing CA to check certificate status.
- Locally cached CRLs can be used until they expire.
- The software checks the CRL from the issuing CA only if it has already been cached and revocation checking is configured to use the CRL.

CRL usage in certificate verification

For example, when a client verifies a peer's certificate, it can use a cached CRL from the issuing CA to determine if the certificate has been revoked. If the CRL is not cached or revocation checking is not configured to use the CRL, the certificate is considered not revoked unless other revocation methods are configured.

Ndcpp: OCSP for syslog

OCSP for Syslog is a method that enables clients to validate the revocation status of server certificates by querying an external OCSP responder, rather than maintaining a local list of revoked certificates.

- Allows real-time certificate revocation status checks via an OCSP responder.
- Reduces the need for clients to store and manage local certificate revocation lists.
- Depends on the availability of the OCSP responder and the inclusion of an OCSP responder URL in the server certificate.

How OCSP for syslog works

OCSP for Syslog enables a client to send the server certificate to an external OCSP responder (CA) server, which validates the certificate and confirms its revocation status.

OCSP validation in syslog

For example, when a remote syslog server presents a certificate containing an OCSP responder URL, the client forwards this certificate to the OCSP responder. The CA server then validates the certificate and returns its status, allowing the client to determine if the certificate is valid or revoked without maintaining a local revocation list.

Import and export support for certificates and associated key pairs

Import and export support for certificates and associated key pairs enables secure management of digital certificates and RSA key pairs using standard formats.

- Certificates and key pairs can be imported in standard PEM (base64) format as part of the CA authentication and enrollment process.
- Identity information in a trust point can be exported to a password-protected PKCS#12 file, which includes the RSA key pair, identity certificate, and CA certificate (or chain).
- Exported PKCS#12 files can be imported to the same or replacement device for recovery or migration purposes.

Reference information for import and export of certificates and key pairs

Import and export operations for certificates and key pairs are essential for secure device authentication, backup, and recovery.

Importing and exporting certificates

For example, after a system crash, a previously exported PKCS#12 file containing the RSA key pair, identity certificate, and CA certificate can be imported to restore the device's identity and trust relationships.

Guidelines and limitations for PKI

Use this reference to review the guidelines and limitations for PKI.

- The maximum number of key pairs you can configure on a Cisco NX-OS device is 16.

- The maximum number of trust points you can declare on a Cisco NX-OS device is 16.
- The maximum number of identify certificates that you can configure on a Cisco NX-OS device are 16.
- The maximum number of certificates in a CA certificate chain is 10.
- The maximum number of trust points you can authenticate to a specific CA is 10.
- Configuration rollbacks do not support the PKI configuration.
- Beginning with Cisco NX-OS Release 9.3(5), Cisco NX-OS software supports NDcPP: OCSP for Syslog.
- Beginning with Cisco NX-OS Release 10.3(3)F, Elliptic Curve Cyptography (ECC) key pair support is provided to generate and import the certificate on Cisco Nexus switches.

Differences in PKI configuration commands

If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default settings for PKI

Use this reference to review the default settings for PKI.

The default settings for PKI define the initial values assigned to key PKI parameters on the system.

- Trust point and RSA key pair are set to **None** by default.
- The RSA key-pair label defaults to the device FQDN, and the modulus is set to 512.
- Revocation check method is set to **CRL** , and certificate authentication uses the local cert-store.

Default PKI parameters reference

This section provides a table listing the default values for each PKI parameter.

Table 38: Default PKI parameters

Parameters	Default
Trust point	None
RSA key pair	None
RSA key-pair label	Device FQDN
RSA key-pair modulus	512
RSA key-pair exportable	Enabled
Revocation check method	CRL

Configuring CAs and digital certificates

Configuring CAs and digital certificates involves performing specific tasks to allow CAs and digital certificates to interoperate on your device.

- Certificate Authorities (CAs) are trusted entities that issue digital certificates.
- Digital certificates are used to authenticate devices and secure communications.
- Proper configuration ensures secure interoperability between your device and other systems.

Tasks for configuring CAs and digital certificates

This section describes the tasks that you must perform to allow CAs and digital certificates on your Cisco NX-OS device to interoperate.

Configure the hostname and IP domain name

You must configure the hostname and IP domain name of the device if you have not yet configured them because the Cisco NX-OS software uses the fully qualified domain name (FQDN) of the device as the subject in the identity certificate. Also, the Cisco NX-OS software uses the device FQDN as a default key label when you do not specify a label during key-pair generation. For example, a certificate named DeviceA.example.com is based on a device hostname of DeviceA and a device IP domain name of example.com.

Caution Changing the hostname or IP domain name after generating the certificate can invalidate the certificate.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **hostname** *hostname*

Example:

```
switch(config)#hostname DeviceA
```

Configures the hostname of the device.

Step 3 **ip domain-name** *name* [**use-vrf** *vrf-name*]

Example:

```
DeviceA(config)#ip domain-name example.com
```

Configures the IP domain name of the device. If you do not specify a VRF name, the command uses the default VRF.

Step 4 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 5 (Optional) **show hosts**

Example:

```
switch#show hosts
```

Displays the IP domain name.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Generate an RSA key pair

You can generate an RSA key pair to sign and/or encrypt and decrypt the security payload during security protocol exchanges for applications. You must generate the RSA key pair before you can obtain a certificate for your device.

Beginning Cisco NX-OS Release 9.3(3), you must explicitly generate RSA key pairs before you associate the Cisco NX-OS device with a trust point CA. Prior to Cisco NX-OS Releases 9.3(3), if unavailable, the RSA key pairs would be auto generated.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
 switch(config)#
```

Enters global configuration mode.

Step 2 **crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]

Example:

```
switch(config)#crypto key generate rsa exportable
```

Generates an RSA key pair. The maximum number of key pairs on a device is 16.

The label string is alphanumeric, case sensitive, and has a maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.).

Valid modulus values are 512, 768, 1024, 1536, and 2048 . The default modulus size is 512.

Note
The security policy on the Cisco NX-OS device and on the CA (where enrollment is planned) should be considered when deciding the appropriate key modulus.

By default, the key pair is not exportable. Only exportable key pairs can be exported in the PKCS#12 format.

Caution
You cannot change the exportability of a key pair.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show crypto key mypubkey rsa**

Example:

```
switch#show crypto key mypubkey rsa
```

Displays the generated key.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Generate an ECC key pair

You can generate an ECC key pair to sign and/or encrypt and decrypt the security payload during security protocol exchanges for applications. You must generate the ECC key pair before you can obtain a certificate for your device. The ECC keys are stronger compared to RSA keys for a given length.

Beginning Cisco NX-OS Release 10.3(3)F, you can generate an ECC key pair to associate the Cisco NX-OS device with a trust point CA.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
 switch(config)#
```

Enters global configuration mode.

Step 2 **crypto key generate ecc** [**label** *ecc-key-label*] [**exportable**] [**modulus** *size*]

Example:

```
switch(config)#crypto key generate ecc exportable modulus 224
```

Generates an ECC key pair. The maximum number of key pairs on a device is 16.

The label string is alphanumeric, case sensitive, and has maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.).

Valid modulus values are 224, 384, and 521. The default modulus size is 224.

Note
The security policy on the Cisco NX-OS device and on the CA (where enrollment is planned) should be considered when deciding the appropriate key modulus.

By default, the key pair is not exportable. Only exportable key pairs can be exported in the PKCS#12 format.

Caution
You cannot change the exportability of a key pair.

Step 3 **no crypto key generate ecc** [**label** *ecc-key-label*]

Example:

```
switch(config)#no crypto key generate ecc label label-name
```

Deletes the ECC key.

Step 4 **exit**

Example:

```
switch(config)#exit
 switch#
```

Exits configuration mode.

Step 5 (Optional) **show crypto key mypubkey ecc**

Example:

```
switch#show crypto key mypubkey ecc
```

Displays the generated ECC key.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Create a trust point CA association

You must associate the Cisco NX-OS device with a trust point CA.

Before you begin

Generate the RSA key pair.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto ca trustpoint** *name*

Example:

```
switch(config)#crypto ca trustpoint admin-ca
switch(config-trustpoint)#
```

Declares a trust point CA that the device should trust and enters trust point configuration mode.

Note
The maximum number of trustpoints that can be configured is 50.

Step 3 **cabundle** *baselabel*

Example:

```
switch(config-trustpoint)#cabundle test
```

Groups the trustpoints under a specific CA bundle. The **No** form of this command detaches the trustpoints from the CA bundle. This command associates the trustpoints to an existing CA bundle and it does not configure any new CA bundle.

Step 4 **enrollment terminal**

Example:

```
switch(config-trustpoint)#enrollment terminal
```

Enables manual cut-and-paste certificate enrollment. The default is enabled.

Note
The Cisco NX-OS software supports only the manual cut-and-paste method for certificate enrollment.

Step 5 **rsakeypair** *label*

Example:

```
switch(config-trustpoint)#rsakeypair SwitchA
```

Specifies the label of the RSA key pair to associate to this trust point for enrollment.

Note
You can specify only one RSA key pair per CA.

Step 6 **exit**

Example:

```
switch(config-trustpoint)#exit
switch(config)#
```

Exits trust point configuration mode.

Step 7 (Optional) **show crypto ca trustpoints**

Example:

```
switch#show crypto ca trustpoints
```

Displays trust point information.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Generate an RSA key pair, on page 191

Configure certificate mapping filters

You can configure mapping filters to validate the CA certificates that are used for authentication. The mapping filters are used to match the CA certificate against a username.

Cisco NX-OS supports the following certificate mapping filters:

- %username%—Substitutes the user's login name.
- %hostname%—Substitutes the peer hostname.

Before you begin

Configure a cert-store for certificate authentication.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto certificatemap mapname** *map-name*

Example:

```
switch(config)#crypto certificatemap mapname filtermap1
```

Creates a new filter map.

Step 3 **filter** [**subject-name** *subject-name* | **altname-email** *e-mail-ID* | **altname-upn** *user-principal-name*]

Example:

```
switch(config-certmap-filter)#filter altname-upn %username%@cisco.com
```

Configures one or more certificate mapping filters within the filter map. These certificate field attributes are supported in the filters:

- subject-name—The required subject name in the LDAP distinguished name (DN) string format. For example:

  ```
  filter subject-name CN=%username%
  ```

  or

  ```
  filter subject-name /C=IN/ST=KA/L=BLR/O=CISCO/OU=ABC/CN=%username%
  ```

- altname-email—The e-mail address that must be present in the certificate as a subject alternative name. For example:

  ```
  filter altname-email %username%@cisco.com
  ```

- altname-upn—The principal name that must be present in the certificate as a subject alternative name. For example:

  ```
  filter altname-upn %username%@%hostname%
  ```

The validation passes if the certificate passes all of the filters configured in the map.

Step 4 **exit**

Example:

```
switch(config-certmap-filter)#exit
switch(config)#
```

Exits certificate mapping filter configuration mode.

Step 5 (Optional) **crypto cert ssh-authorize** [**default** | *issuer-CAname*] [**map** *map-name1* [*map-name2*]]

Example:

```
switch(config)#crypto cert ssh-authorize default map filtermap1
```

Configures a certificate mapping filter for the Secure Shell (SSH) protocol. You can use the default filter map for SSH authorization or specify the issuer of the CA certificate. If you do not use the default map, you can specify one or two filter maps for authorization.

If you specify the issuer of the CA certificate, the certificate bound to the user account is validated as successful if it passes one of the configured maps.

Step 6 (Optional) **show crypto certificatemap**

Example:

```
switch#show crypto certificatemap
```

Displays the certificate mapping filters.

Step 7 (Optional) **show crypto ssh-auth-map**

Example:

```
switch#show crypto ssh-auth-map
```

Displays the mapping filters configured for SSH authentication.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Authenticate the CA

Create an association with the CA. Obtain the CA certificate or CA certificate chain.

The configuration process of trusting a CA is complete only when the CA is authenticated to the Cisco NX-OS device. You must authenticate your Cisco NX-OS device to the CA by obtaining the self-signed certificate of the CA in PEM format, which contains the public key of the CA. Because the certificate of the CA is self-signed (the CA signs its own certificate) the public key of the CA should be manually authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate.



Note The CA that you are authenticating is not a self-signed CA when it is a subordinate CA to another CA, which itself may be a subordinate to yet another CA, and so on, finally ending in a self-signed CA. This type of CA certificate is called the *CA certificate chain* of the CA being authenticated. In this case, you must input the full list of the CA certificates of all the CAs in the certification chain during the CA authentication. The maximum number of certificates in a CA certificate chain is 10.

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto ca authenticate** *name* **pemfile** *uri0*

Example:

```
switch(config)#crypto ca authenticate admin-ca
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB
kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO
MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE
ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD
```

```
QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN
AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG
A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN
AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI
OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E
BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE-----
END OF INPUT
Fingerprint(s): MD5 Fingerprint=00:00:5E:00:53:01:00:00:5E:00:53:02:38:6F:78:12
Do you accept this certificate? [yes/no]: yes
```

Prompts you to cut and paste the certificate of the CA. Use the same name that you used when declaring the CA.

Also validates and attaches the CA chain directly to the specified trust point.

The maximum number of trust points that you can authenticate to a specific CA is 10.

Note
For subordinate CA authentication, the Cisco NX-OS software requires the full chain of CA certificates ending in a self-signed CA because the CA chain is needed for certificate verification as well as for PKCS#12 format export.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show crypto ca trustpoints**

Example:

```
switch#show crypto ca trustpoints
```

Displays the trust point CA information.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Create a trust point CA association, on page 193

Configure certificate revocation checking methods

During security exchanges with a client (for example, an SSH user), the Cisco NX-OS device performs the certificate verification of the peer certificate sent by the client. The verification process may involve certificate revocation status checking.

You can configure the device to check the CRL downloaded from the CA. Downloading the CRL and checking locally does not generate traffic in your network. However, certificates can be revoked between downloads and your device would not be aware of the revocation.

Before you begin

Authenticate the CA.

Ensure that you have configured the CRL if you want to use CRL checking.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto ca trustpoint** *name*

Example:

```
switch(config)#crypto ca trustpoint admin-ca
switch(config-trustpoint)#
```

Specifies a trust point CA and enters trust point configuration mode.

Step 3 **revocation-check** {**crl** [**none**] | **none**}

Example:

```
switch(config-trustpoint)#revocation-check none
```

Configures the certificate revocation checking methods. The default method is **crl** .

The Cisco NX-OS software uses the certificate revocation methods in the order that you specify.

Step 4 **exit**

Example:

```
switch(config-trustpoint)#exit
switch(config)#
```

Exits trust point configuration mode.

Step 5 (Optional) **show crypto ca trustpoints**

Example:

```
switch#show crypto ca trustpoints
```

Displays the trust point CA information.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Authenticate the CA, on page 197
Configure a CRL, on page 206

Generate a certificate request

Use this procedure to generate a certificate request for identity certificates in NX-OS.

- This task is required to obtain identity certificates from a Certificate Authority (CA).

You must generate a request to obtain identity certificates from the associated trust point CA for each of your device's RSA key pairs. You must then cut and paste the displayed request into an e-mail or in a website form for the CA.

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto ca enroll** *name*

Example:

```
switch(config)#crypto ca enroll admin-ca
Create the certificate request ..
Create a challenge password. You will need to verbally provide this
password to the CA Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.
Password:nbv123
The subject name in the certificate will be: DeviceA.cisco.com
Include the switch serial number in the subject name? [yes/no]: no
Include an IP address in the subject name [yes/no]: yes
ip address:172.22.31.162
The certificate request will be displayed...
-----BEGIN CERTIFICATE REQUEST-----
MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY
```

```
0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y
P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S
VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ
KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=
-----END CERTIFICATE REQUEST-----
```

Generates a certificate request for an authenticated CA.

Note
You must remember the challenge password. It is not saved with the configuration. You must enter this password if your certificate needs to be revoked.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits trust point configuration mode.

Step 4 (Optional) **show crypto ca certificates**

Example:

```
switch#show crypto ca certificates
```

Displays the CA certificates.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Create a trust point CA association, on page 193

Install an identity certificate

Install an identity certificate to enable secure communication and authentication using PKI in NX-OS.

You can receive the identity certificate from the CA by e-mail or through a web browser in base64 encoded text form. You must install the identity certificate from the CA by cutting and pasting the encoded text.

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto ca import** *name* **certificate**

Example:

```
switch(config)#crypto ca import admin-ca certificate
input (cut & paste) certificate in PEM format:
-----BEGIN CERTIFICATE-----
MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G
CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD
VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz
Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w
NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu
Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C
dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47
glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb
x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw
GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR
bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW
pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE
BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w
DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh
cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6
Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6
Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH
AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl
LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4
XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF
AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw
E36cIZu4WsExREqxbTk8ycx7V5o=
-----END CERTIFICATE-----
```

Prompts you to cut and paste the identity certificate for the CA named admin-ca.

The maximum number of identify certificates that you can configure on a device is 16.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show crypto ca certificates**

Example:

```
switch#show crypto ca certificates
```

Displays the CA certificates.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Create a trust point CA association, on page 193

Ensuring trust point configurations persist across reboots

Trust point configuration persistence ensures that trust points and their associated certificates, key pairs, and CRLs remain available after device reboots.

- Trust point configuration must be explicitly copied to the startup configuration to persist across reboots.
- Certificates, key pairs, and CRLs are automatically persistent if the trust point configuration is saved in the startup configuration.
- Deleting certificates or key pairs requires saving the running configuration to make deletions permanent.

Reference information for trust point configuration persistence

Trust point configuration is a normal device configuration that persists across system reboots only if you copy it explicitly to the startup configuration. The certificates, key pairs, and CRL associated with a trust point are automatically persistent if you have already copied the trust point configuration in the startup configuration. Conversely, if the trust point configuration is not copied to the startup configuration, the certificates, key pairs, and CRL associated with it are not persistent since they require the corresponding trust point configuration after a reboot. Always copy the running configuration to the startup configuration to ensure that the configured certificates, key pairs, and CRLs are persistent. Also, save the running configuration after deleting a certificate or key pair to ensure that the deletions are permanent.

The certificates and CRL associated with a trust point automatically become persistent when imported (that is, without explicitly copying to the startup configuration) if the specific trust point is already saved in startup configuration.

- Copy the running configuration to the startup configuration to ensure persistence.
- Save the running configuration after deleting certificates or key pairs.
- Importing certificates and CRLs is persistent if the trust point is already in the startup configuration.

Note Copying the configuration to an external server does include the certificates and key pairs.

It is recommended to create a password-protected backup of the identity certificates and save it to an external server.

Ensuring trust point configuration persistence

For example, after configuring a trust point and importing certificates, copy the running configuration to the startup configuration to ensure all trust point data persists after a device reboot. If you delete a certificate or key pair, save the running configuration to make the deletion permanent.

Related Topics

Export identity information in PKCS 12 format, on page 204

Export identity information in PKCS 12 format

Export the identity certificate and associated key pair and CA certificates for a trust point CA in PKCS#12 format for backup or recovery.

You can export the identity certificate along with the RSA key pair and CA certificate (or the entire chain in the case of a subordinate CA) of a trust point to a PKCS#12 file for backup purposes. You can import the certificate and RSA key pair to recover from a system crash on your device or when you replace the supervisor modules.



Note You can use only the bootflash: *filename* format when specifying the export URL.

Before you begin

Authenticate the CA.

Install an identity certificate.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto ca export** *name* **pkcs12 bootflash:** *filename password*

Example:

```
switch(config)#crypto ca export admin-ca pkcs12 bootflash:adminid.p12 nbv123
```

Exports the identity certificate and associated key pair and CA certificates for a trust point CA. The password is alphanumeric, case sensitive, and has a maximum length of 128 characters.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 **copy booflash:** *filename scheme* **://** *server* **/** [*url* /] *filename*

Example:

```
switch#copy bootflash:adminid.p12 tftp:adminid.p12
```

Copies the PKCS#12 format file to a remote server.

For the *scheme* argument, you can enter **tftp:** , **ftp:** , **scp:** , or **sftp:** . The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.

The *server* , *url* , and *filename* arguments are case sensitive.

Related Topics

Generate an RSA key pair, on page 191
Authenticate the CA, on page 197
Install an identity certificate, on page 201

Import identity information in PKCS 12 or PKCS 7 format

You can import the certificate and RSA key pair to recover from a system crash on your device or when you replace the supervisor modules.

Note You can use only the bootflash: *filename* format when specifying the import URL.

Before you begin

Ensure that the trust point is empty by checking that no RSA key pair is associated with it and no CA is associated with the trust point using CA authentication.

Procedure

Step 1 **copy** *scheme* **://** *server* **/** [*url* /] *filename* **bootflash:** *filename*

Example:

```
switch#copy tftp:adminid.p12 bootflash:adminid.p12
```

Copies the PKCS#12 format file from the remote server.

For the *scheme* argument, you can enter **tftp:** , **ftp:** , **scp:** , or **sftp:** . The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.

The *server* , *url* , and *filename* arguments are case sensitive.

Step 2 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 3 **crypto ca import** *name* **[pksc12 | pkcs7] bootflash:** *filename*

Example:

```
switch(config)#crypto ca import admin-ca pkcs12 bootflash:adminid.p12 nbv123
```

Imports the identity certificate and associated key pair and CA certificates for trust point CA.

Step 4 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 5 (Optional) **show crypto ca certificates**

Example:

```
switch#show crypto ca certificates
```

Displays the CA certificates.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure a CRL

You can manually configure CRLs that you have downloaded from the trust points. The Cisco NX-OS software caches the CRLs in the device bootflash (cert-store). During the verification of a peer certificate, the Cisco NX-OS software checks the CRL from the issuing CA only if you have downloaded the CRL to the device and you have configured certificate revocation checking to use the CRL.

Before you begin

Ensure that you have enabled certificate revocation checking.

Procedure

Step 1 **copy** *scheme* **:** [*// server* / [*url* /]] *filename* **bootflash:** *filename*

Example:

```
switch#copy tftp:adminca.crl bootflash:adminca.crl
```

Downloads the CRL from a remote server.

For the *scheme* argument, you can enter **tftp:** , **ftp:** , **scp:** , or **sftp:** . The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.

The *server* , *url* , and *filename* arguments are case sensitive.

Step 2 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 3 **crypto ca crl request** *name* **bootflash:** *filename*

Example:

```
switch(config)#crypto ca crl request admin-ca bootflash:adminca.crl
```

Configures or replaces the current CRL with the one specified in the file.

Step 4 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 5 (Optional) **show crypto ca crl** *name*

Example:

```
switch#show crypto ca crl admin-ca
```

Displays the CA CRL information.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Delete certificates from the CA configuration

You can delete the identity certificates and CA certificates that are configured in a trust point. You must first delete the identity certificate, followed by the CA certificates. After deleting the identity certificate, you can disassociate the RSA key pair from a trust point. You must delete certificates to remove expired or revoked certificates, certificates that have compromised (or suspected to be compromised) key pairs, or CAs that are no longer trusted.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto ca trustpoint** *name*

Example:

```
switch(config)#crypto ca trustpoint admin-ca
switch(config-trustpoint)#
```

Specifies a trust point CA and enters trust point configuration mode.

Step 3 **delete ca-certificate**

Example:

```
switch(config-trustpoint)#delete ca-certificate
```

Deletes the CA certificate or certificate chain.

Step 4 **delete certificate** [**force**]

Example:

```
switch(config-trustpoint)#delete certificate
```

Deletes the identity certificate.

You must use the **force** option if the identity certificate you want to delete is the last certificate in a certificate chain or only identity certificate in the device. This requirement ensures that you do not mistakenly delete the last certificate in a certificate chain or only the identity certificate and leave the applications (such as SSH) without a certificate to use.

Step 5 **exit**

Example:

```
switch(config-trustpoint)#exit
switch(config)#
```

Exits trust point configuration mode.

Step 6 (Optional) **show crypto ca certificates** [*name*]

Example:

```
switch#show crypto ca certificates admin-ca
```

Displays the CA certificate information.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Delete RSA key pairs from a device

You can delete the RSA key pairs from a Cisco NX-OS device if you believe the RSA key pairs were compromised in some way and should no longer be used.



Note After you delete RSA key pairs from a device, ask the CA administrator to revoke your device's certificates at the CA. You must supply the challenge password that you created when you originally requested the certificates.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **crypto key zeroize rsa** *label*

Example:

```
switch(config)#crypto key zeroize rsa MyKey
```

Deletes the RSA key pair.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show crypto key mypubkey rsa**

Example:

```
switch#show crypto key mypubkey rsa
```

Displays the RSA key pair configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Generate a certificate request, on page 200

PKI configuration verification commands

Use this reference to verify the PKI configuration.

- Verification of PKI configuration involves displaying information about RSA key-pairs, CA certificates, CRLs, and trust points.
- Several show commands are available to display specific PKI-related configuration details.
- These commands help administrators confirm the status and configuration of PKI components on the device.

Display PKI configuration information

To display PKI configuration information, perform one of the following tasks:

Table 39: PKI configuration display commands

Command	Purpose	Product
show crypto key mypubkey rsa	Displays information about the RSA public keys generated on the device.	
show crypto ca certificates	Displays information about CA and identity certificates.	
show crypto ca crl	Displays information about CA CRLs.	
show crypto ca certstore	Displays the cert-store configuration.	nxos_5_0_1
show crypto ca remote-certstore	Displays the remote cert-store configuration.	nxos_5_0_1
show crypto certificatemap	Displays the certificate mapping filters.	nxos_5_0_1
show crypto ssh-auth-map	Displays the mapping filters configured for SSH authorization.	nxos_5_0_1
show crypto ca trustpoints	Displays information about CA trust points.	

Configuration examples for PKI

Configuration examples for PKI demonstrate the steps required to set up certificates and CRLs on network devices using a certificate server.

- Shows how to configure certificates and CRLs on devices.
- Uses a Microsoft Windows Certificate server as an example.
- Applicable to any type of certificate server, not limited to Microsoft Windows Certificate server.

Reference information for PKI configuration examples

This section provides supporting information for configuring certificates and CRLs on devices using a certificate server.



Note You can use any type of certificate server to generate digital certificates. You are not limited to using the Microsoft Windows Certificate server.

Configure certificates on a device

To configure certificates on a Cisco NX-OS device, follow these steps:

Procedure

Step 1 Configure the device FQDN.

```
switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)#hostname Device-1
Device-1(config)#
```

Step 2 Configure the DNS domain name for the device.

```
Device-1(config)#ip domain-name cisco.com
```

Step 3 Create a trust point.

```
Device-1(config)#crypto ca trustpoint myCA
Device-1(config-trustpoint)#exit
Device-1#show crypto ca trustpoints
trustpoint: myCA; key:
revokation methods:  crl
```

Step 4 Create an RSA key pair for the device.

```
Device-1(config)#crypto key generate rsa label myKey exportable modulus 1024
Device-1(config)#exit
Device-1#show crypto key mypubkey rsa
key label: myKey
```

```
key size: 1024
exportable: yes
```

Step 5 Associate the RSA key pair to the trust point.

```
Device-1(config)#crypto ca trustpoint myCA
Device-1(config-trustpoint)#rsakeypair myKey
Device-1(config-trustpoint)#exit
Device-1#show crypto ca trustpoints
trustpoint: myCA; key: myKey
revokation methods:  crl
```

Step 6 Download the CA certificate from the Microsoft Certificate Service web interface and Authenticate the CA that you want to enroll to the trust point.

```
Device-1(config)#crypto ca authenticate myCA
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
 -----BEGIN CERTIFICATE-----
MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB
kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO
MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE
ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD
QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN
AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG
A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN
AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI
OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E
BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE-----
END OF INPUT
Fingerprint(s): MD5 Fingerprint=00:00:5E:00:53:01:00:00:5E:00:53:02:38:6F:78:12
Do you accept this certificate? [yes/no]:
y
Device-1#show crypto ca certificates
Trustpoint: myCA
CA certificate 0:
subject= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/
L=Bangalore/O=Yourcompany/OU=netstorage/CN=Aparna CA
issuer= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/
L=Bangalore/O=Yourcompany/OU=netstorage/CN=Aparna CA
serial=0560D289ACB419944F4912258CAD197A
notBefore=May  3 22:46:37 2005 GMT
notAfter=May  3 22:55:17 2007 GMT
MD5 Fingerprint=00:00:5E:00:53:01:00:00:5E:00:53:02:38:6F:78:12
purposes: sslserver sslclient ike
```

Step 7 Generate a request certificate to use to enroll with a trust point.

```
Device-1(config)#crypto ca enroll myCA
  Create the certificate request ..
 Create a challenge password. You will need to verbally provide this
password to the CA Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.
Password: nbv123
The subject name in the certificate will be:
Device-1.cisco.com
```

```
Include the switch serial number in the subject name? [yes/no]:
no
Include an IP address in the subject name [yes/no]:
yes
ip address:
10.10.1.1
The certificate request will be displayed...
 -----BEGIN CERTIFICATE REQUEST-----
MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY
0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y
P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S
VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ
KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=
-----END CERTIFICATE REQUEST-----
```

Step 8 Request an identity certificate from the Microsoft Certificate Service web interface. and import the identity certificate.

```
Device-1(config)#crypto ca import myCA certificate
input (cut & paste) certificate in PEM format:
-----BEGIN CERTIFICATE-----
MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G
CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD
VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz
Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w
NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu
Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C
dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47
glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb
x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw
GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR
bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW
pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE
BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w
DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh
cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6
Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6
Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH
AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRfbnJvbGwvc3Nl
LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4
XENlcnRfbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF
AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw
E36cIZu4WsExREqxbTk8ycx7V5o=
-----END CERTIFICATE-----
Device-1(config)#exit
Device-1#
```

Step 9 Verify the certificate configuration and Save the certificate configuration to the startup configuration.

Related Topics

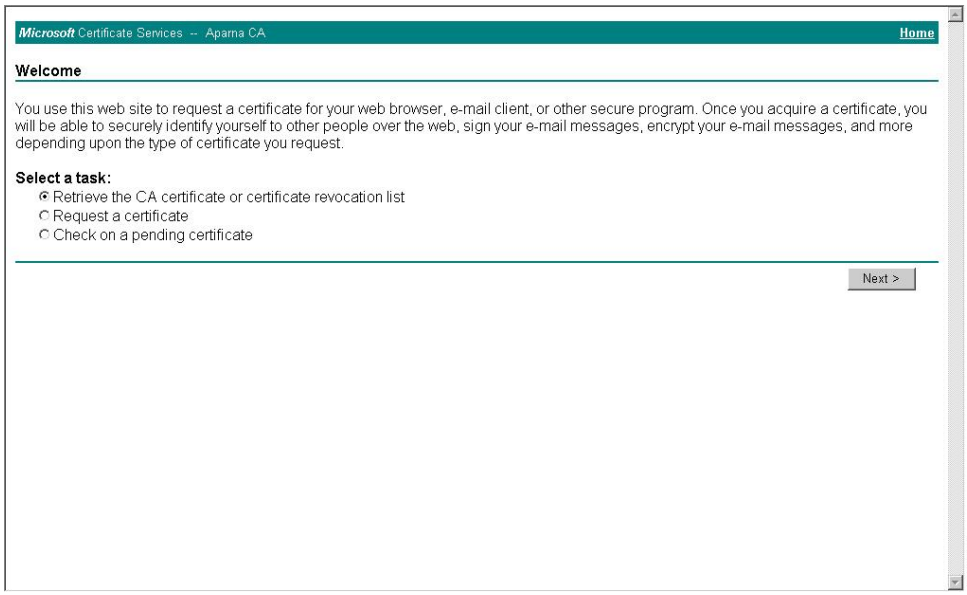
Download a CA certificate, on page 213
Request an identity certificate, on page 217

Download a CA certificate

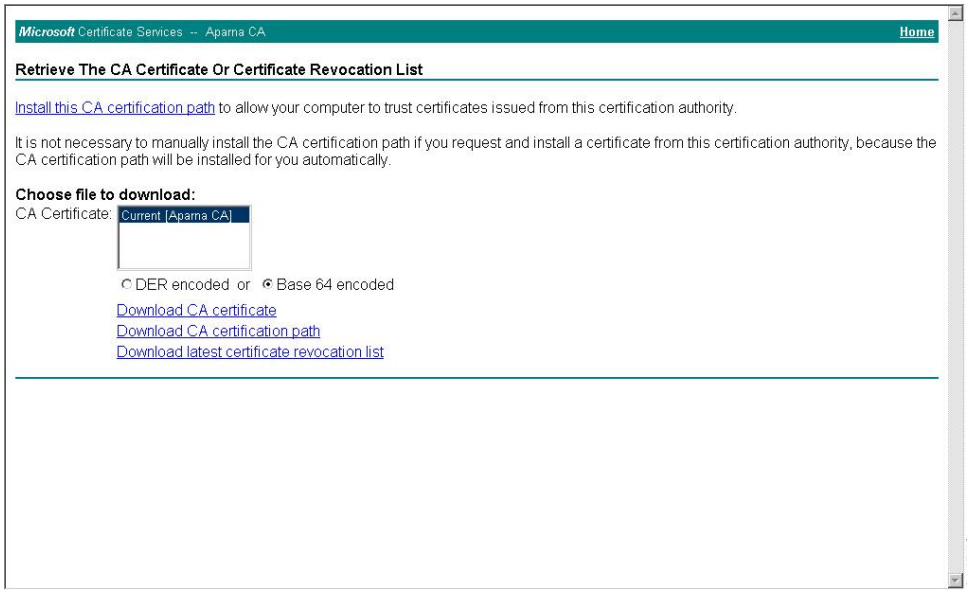
To download a CA certificate from the Microsoft Certificate Services web interface, follow these steps:

Procedure

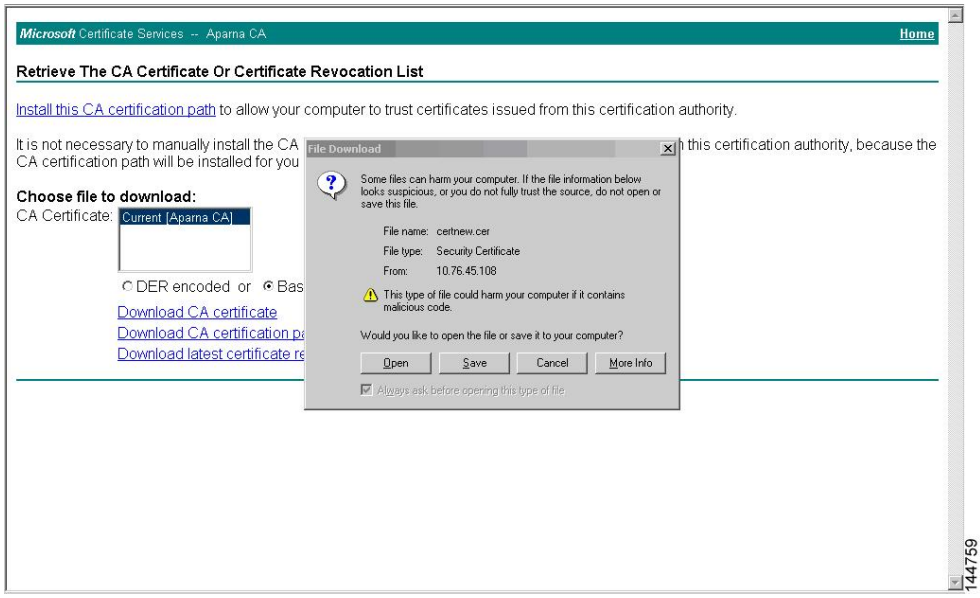
Step 1 From the Microsoft Certificate Services web interface, click **Retrieve the CA certificate or certificate revocation task** and click **Next** .



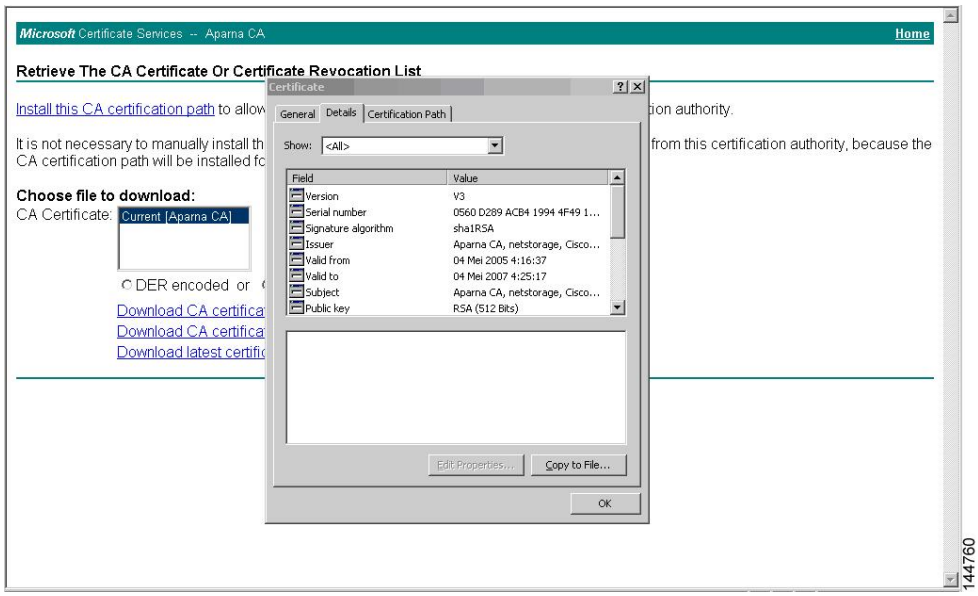
Step 2 From the display list, choose the CA certificate file to download from the displayed list. Then click **Base 64 encoded** and click **Download CA certificate** .



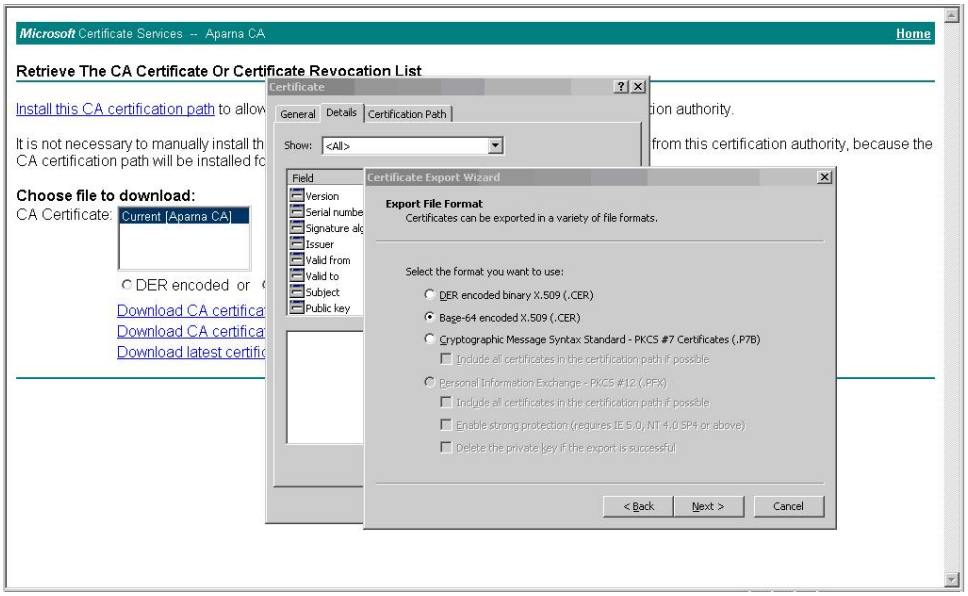
Step 3 Click **Open** in the File Download dialog box.



Step 4 In the Certificate dialog box, click **Copy to File** and click **OK** .



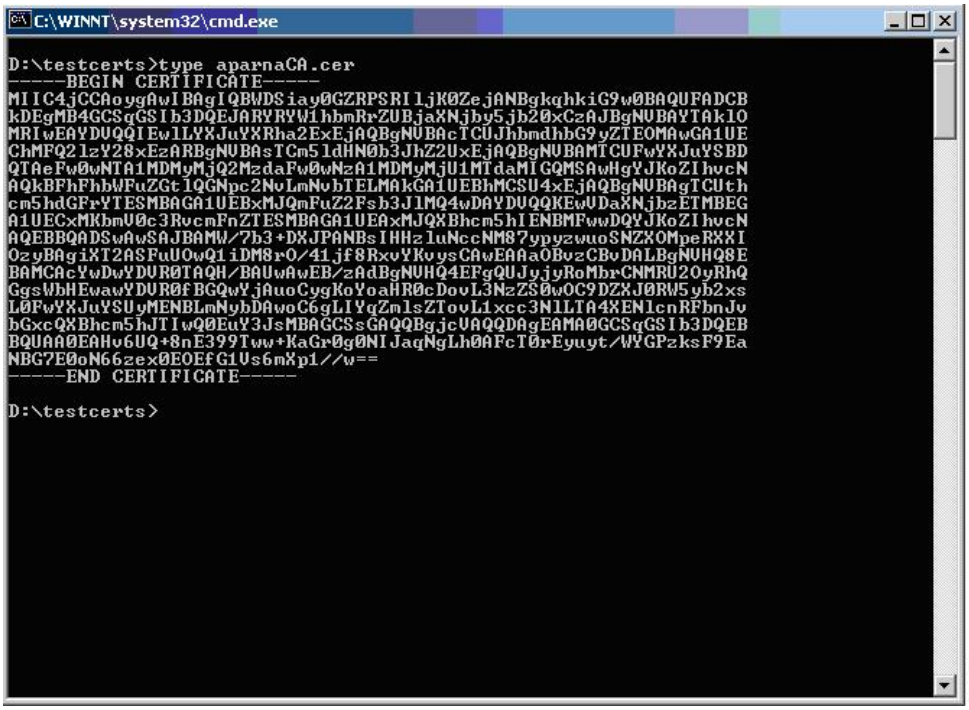
Step 5 From the Certificate Export Wizard dialog box, choose the **Base-64 encoded X.509 (CER)** and click **Next** .



Step 6 In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and click **Next** .

Step 7 In the Certificate Export Wizard dialog box, click **Finish** .

Step 8 Enter the Microsoft Windows **type** command to display the CA certificate stored in Base-64 (PEM) format.

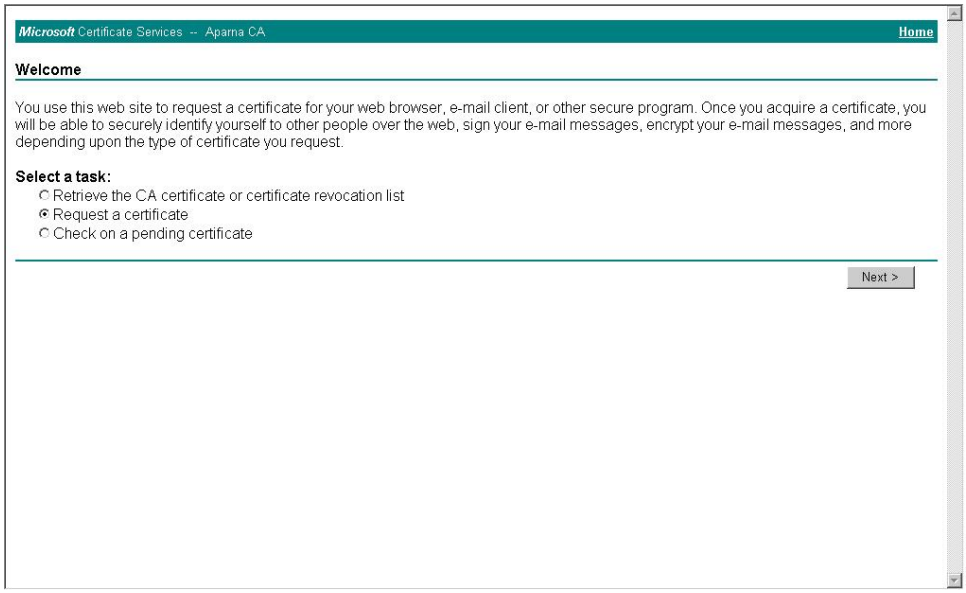


Request an identity certificate

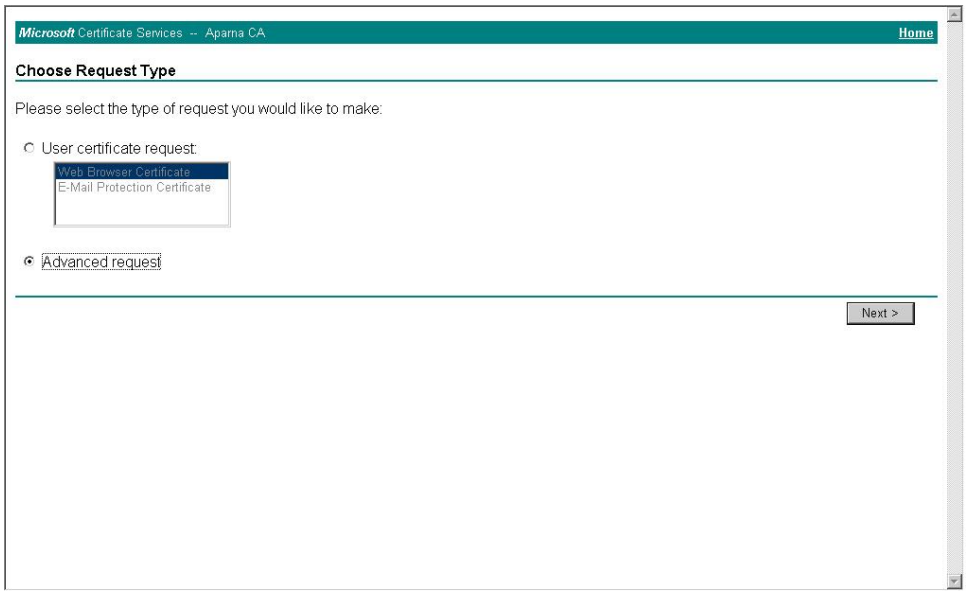
To request an identity certificate from a Microsoft Certificate server using a PKCS#12 certificate signing request (CSR), follow these steps:

Procedure

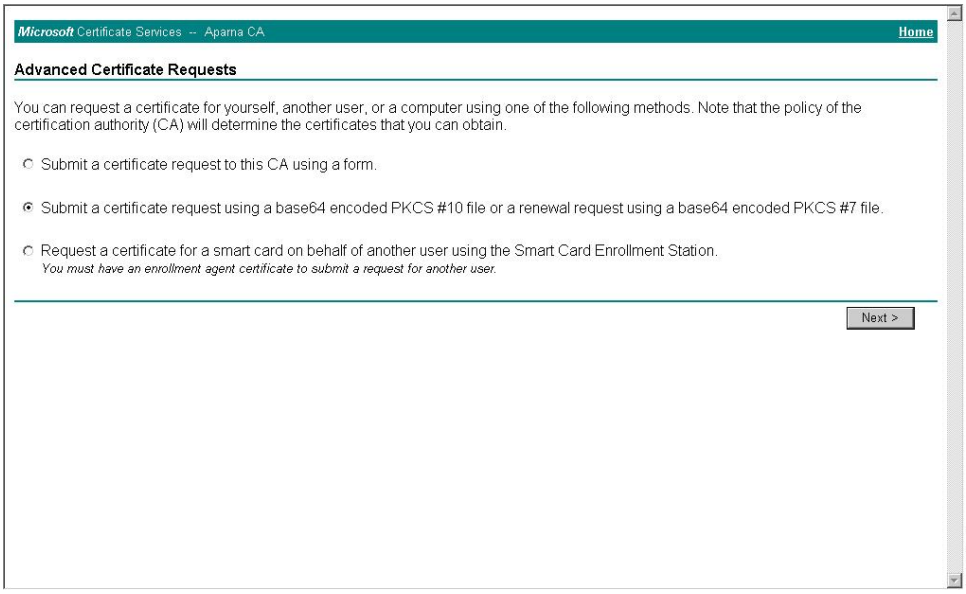
Step 1 From the Microsoft Certificate Services web interface, click **Request a certificate** and click **Next** .



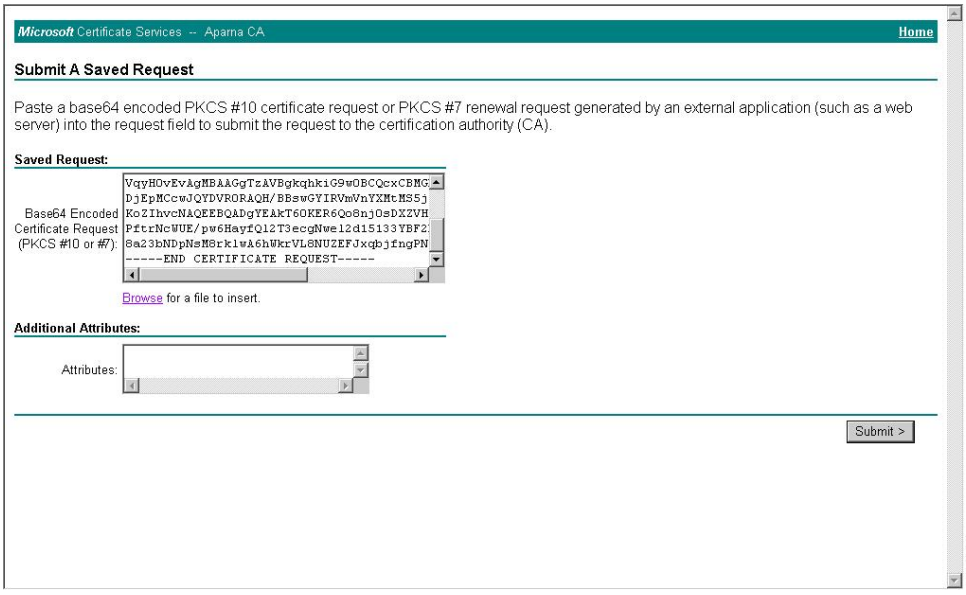
Step 2 Click **Advanced request** and click **Next** .



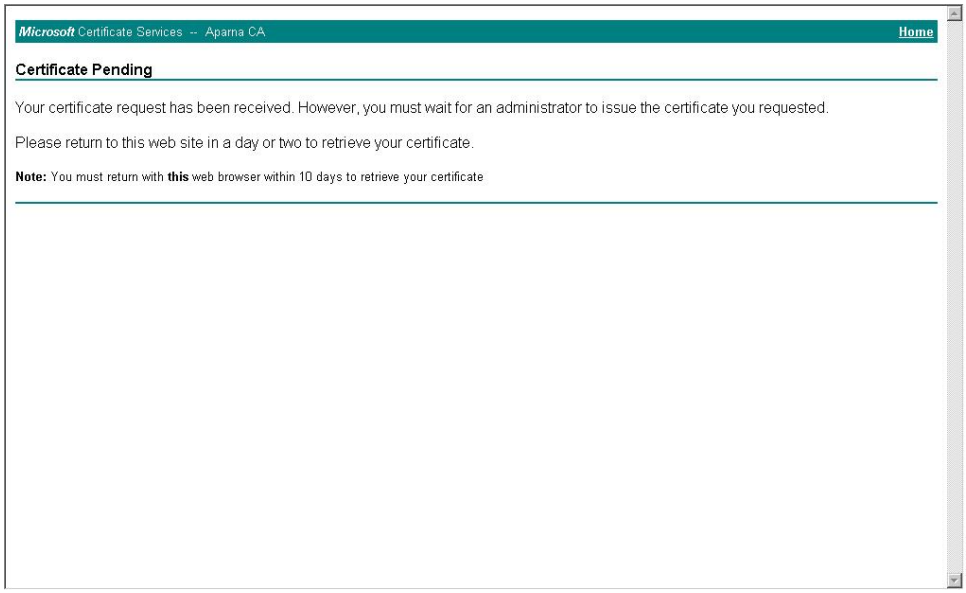
Step 3 Click **Submit a certificate request using a base64 encoded PKCS#10 file or a renewal request using a base64 encoded PKCS#7 file** and click **Next** .



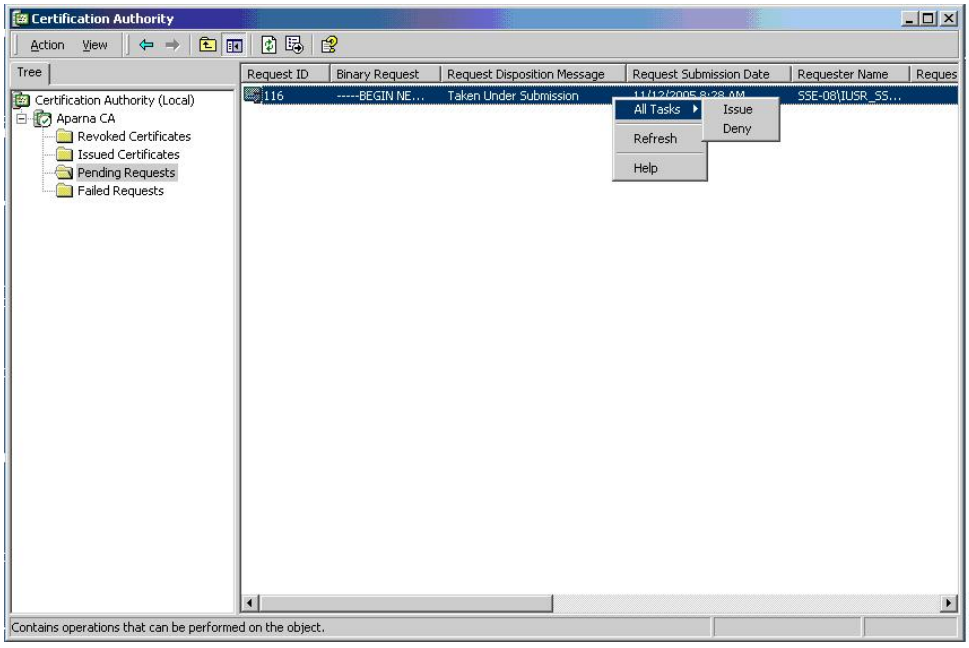
Step 4 In the Saved Request text box, paste the base64 PKCS#10 certificate request and click **Next** . The certificate request is copied from the Cisco NX-OS device console.



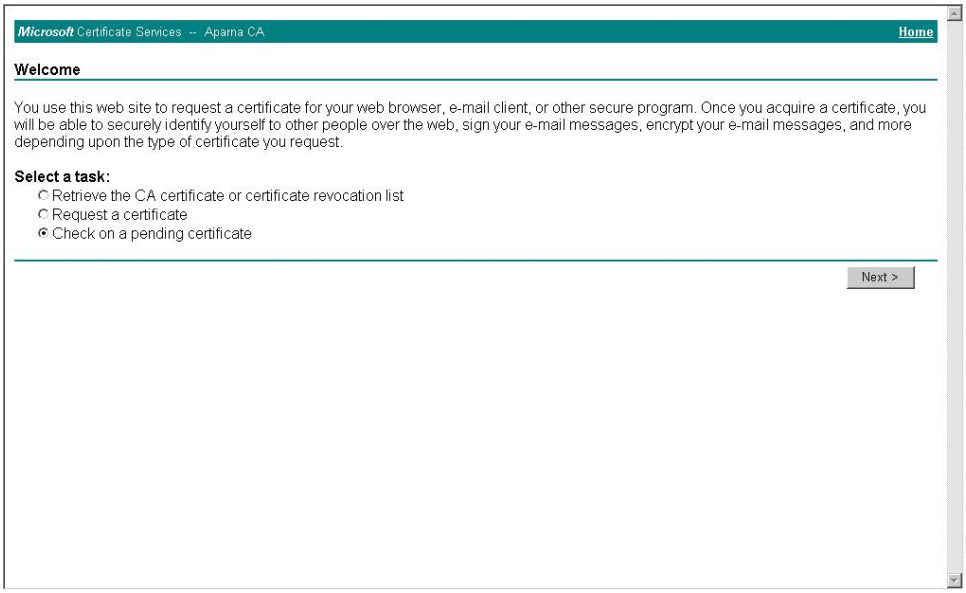
Step 5 Wait one or two days until the certificate is issued by the CA administrator.



Step 6 Note that the CA administrator approves the certificate request.

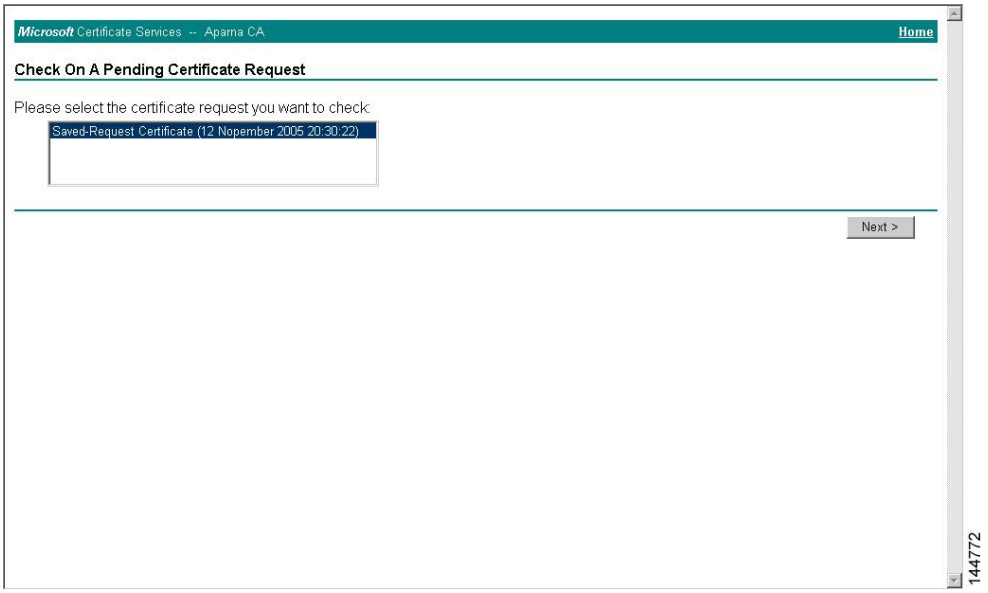


Step 7 From the Microsoft Certificate Services web interface, click **Check on a pending certificate** and click **Next**



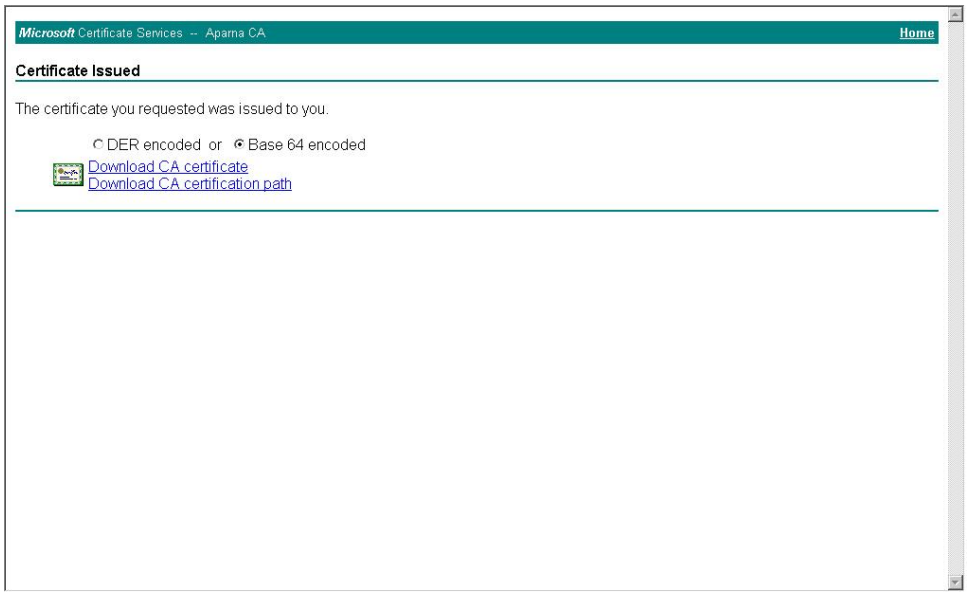
.

Step 8 Choose the certificate request that you want to check and click **Next** .

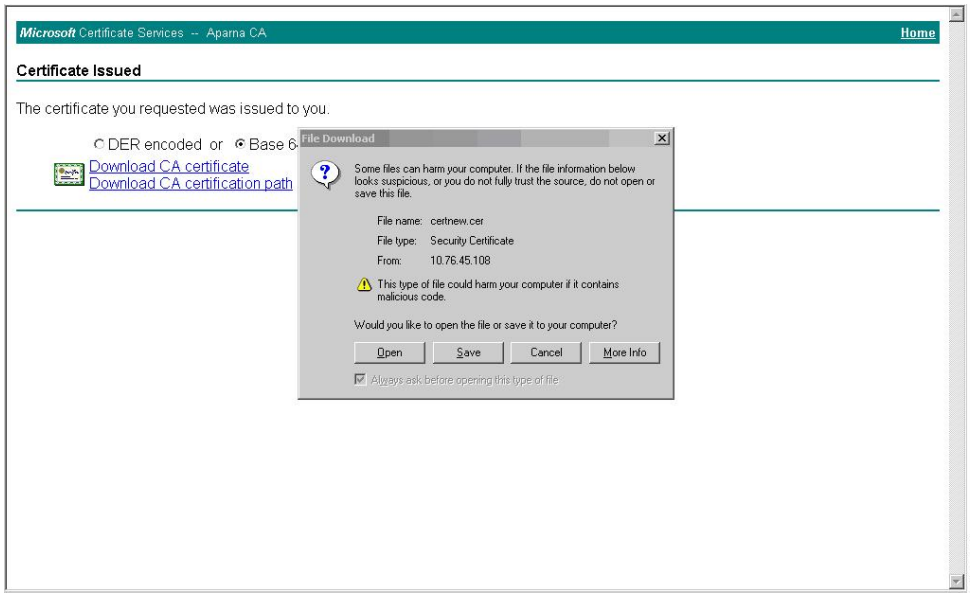


Step 9 Complete the remaining identity certificate steps.

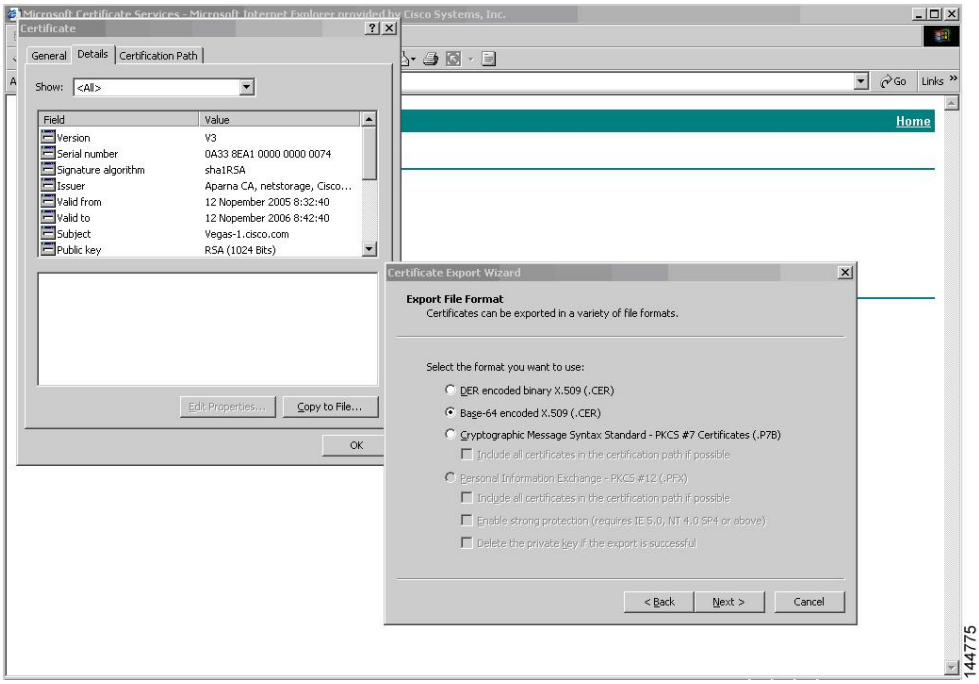
a) Click **Base 64 encoded** and click **Download CA certificate** .



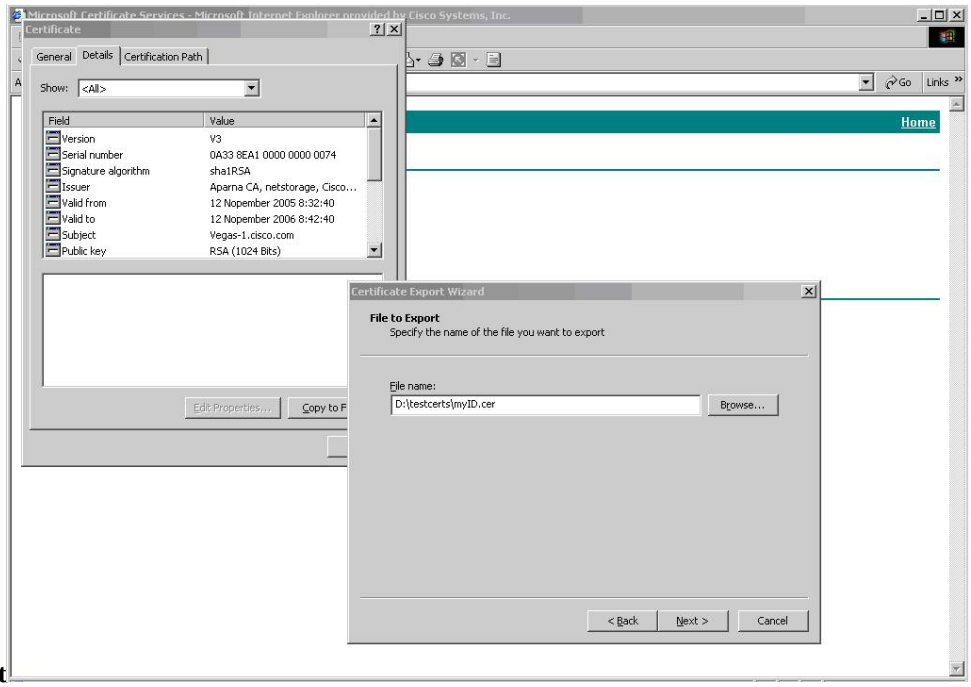
b) In the File Download dialog box, click **Open** .



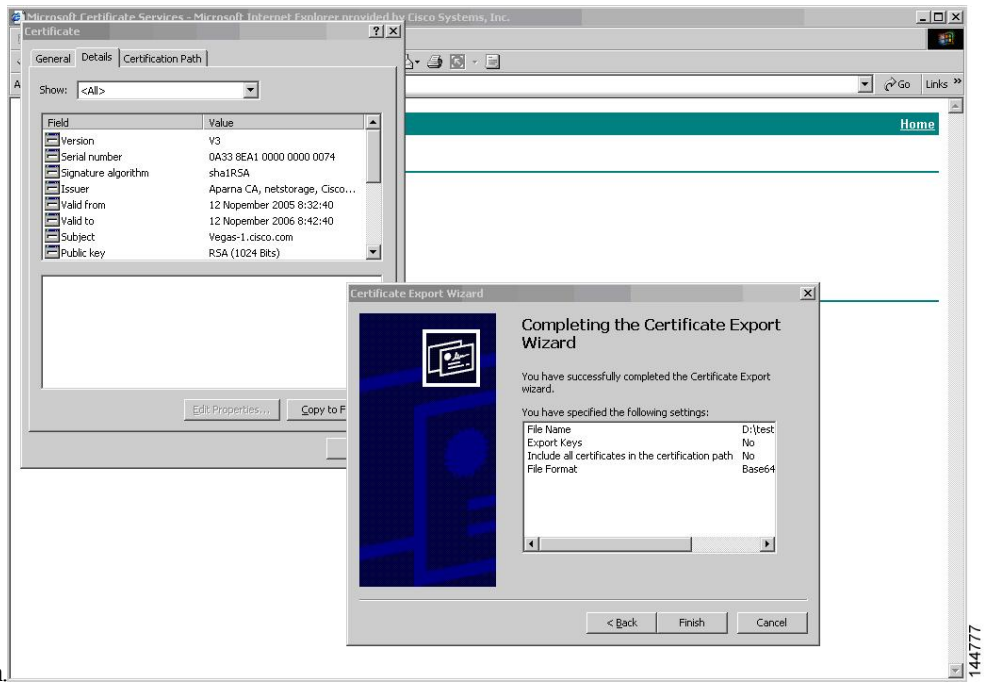
c) In the Certificate box, click **Details** tab and click **Copy to File...** . In the Certificate Export Dialog box, click **Base-64 encoded X.509 (.CER)** , and click **Next** .



d) In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name

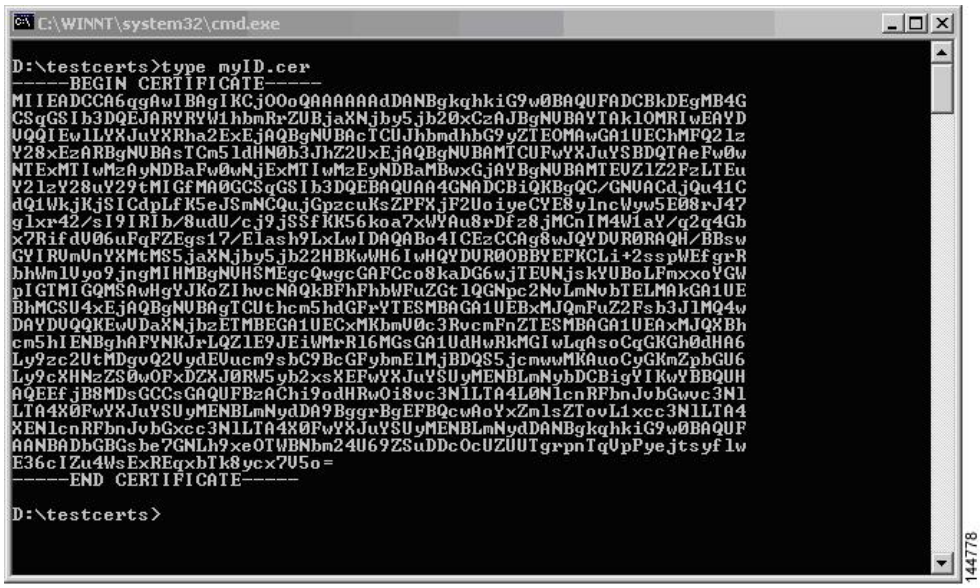


and click **Next**



e) Click **Finish**.

f) Enter the Microsoft Windows **type** command to display the identity certificate in base64-encoded format.



Related Topics

Generate a certificate request, on page 200
Configure certificates on a device, on page 211

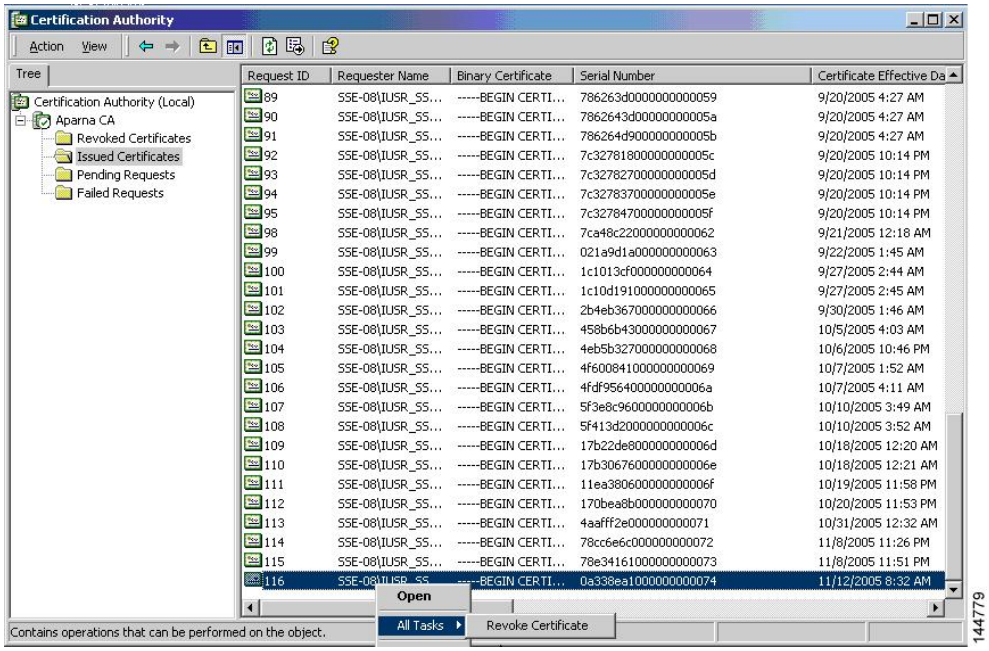
Revoke a certificate

To revoke a certificate using the Microsoft CA administrator program, follow these steps:

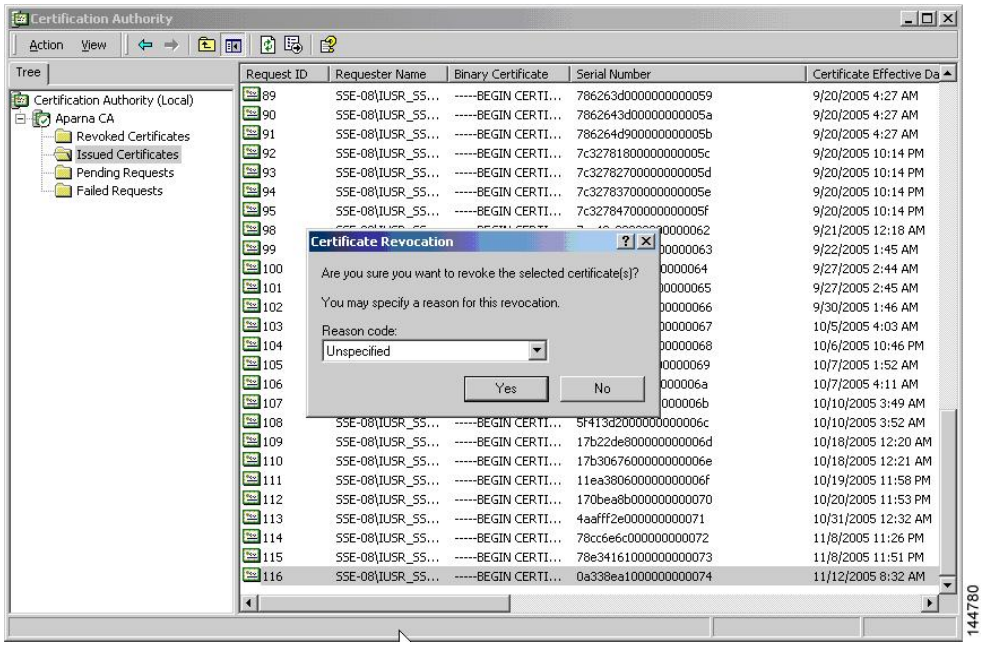
Procedure

Step 1 From the Certification Authority tree, click **Issued Certificates** folder. From the list, right-click the certificate that you want to revoke.

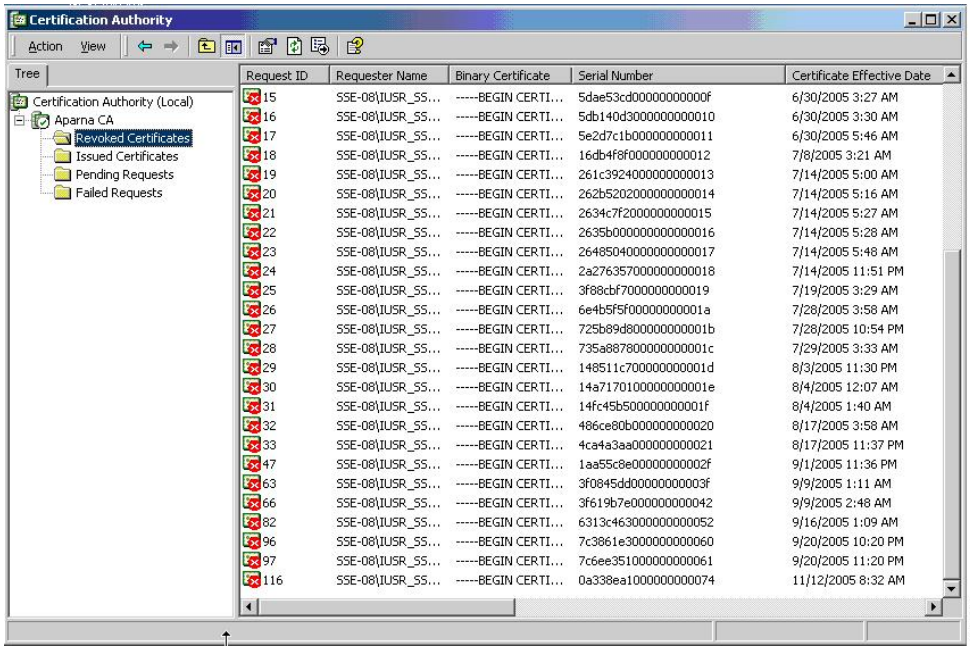
Step 2 Choose **All Tasks> Revoke Certificate** .



Step 3 From the Reason code drop-down list, choose a reason for the revocation and click **Yes** .



Step 4 Click the **Revoked Certificates** folder to list and verify the certificate revocation.

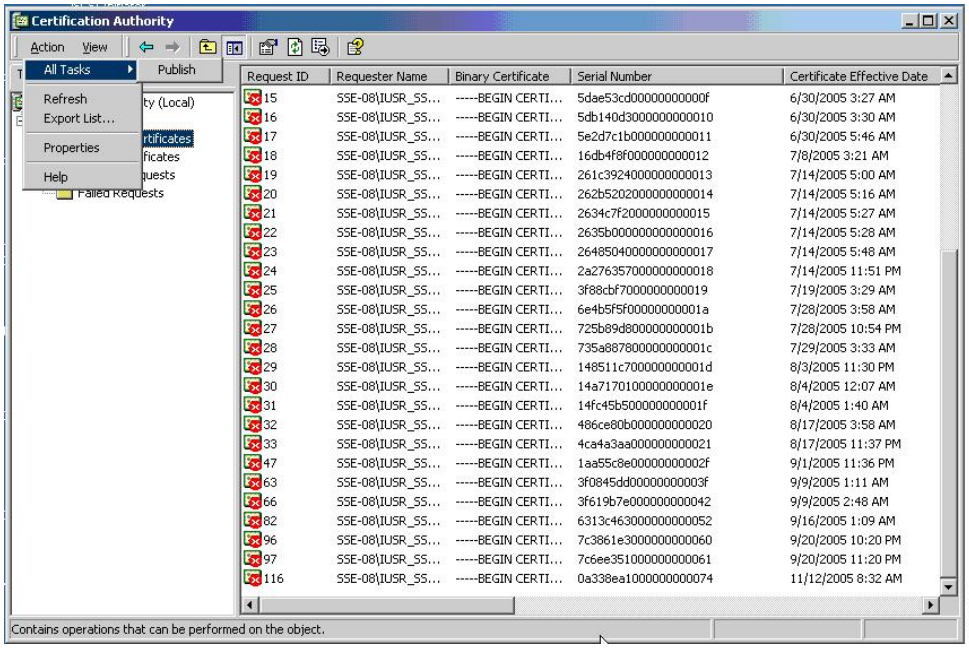


Generate and publish the CRL

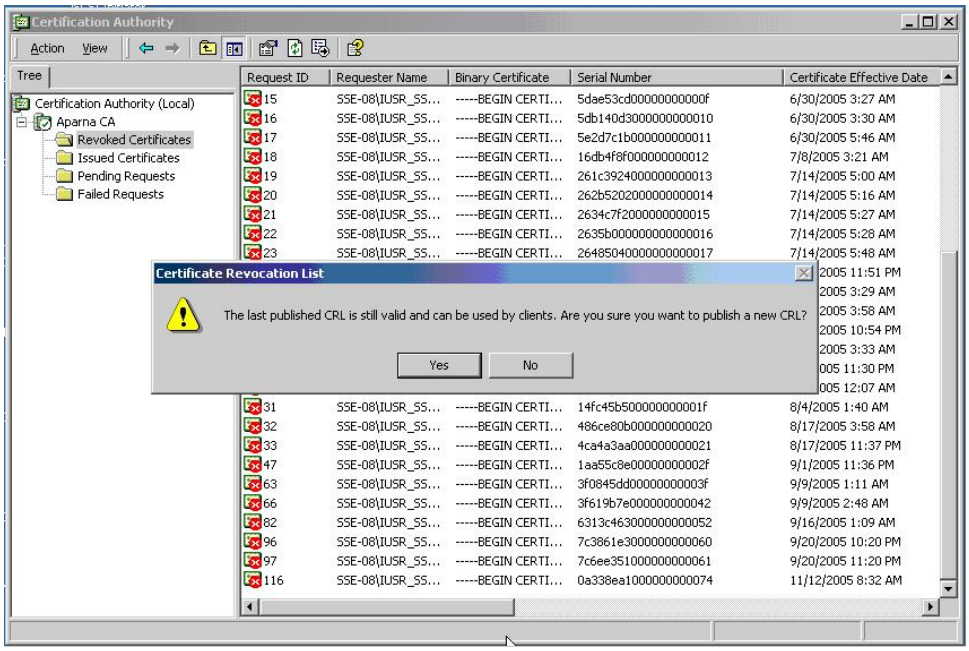
To generate and publish the CRL using the Microsoft CA administrator program, follow these steps:

Procedure

Step 1 From the Certification Authority screen, choose **Action> All Tasks> Publish** .



Step 2 In the Certificate Revocation List dialog box, click **Yes** to publish the latest CRL.

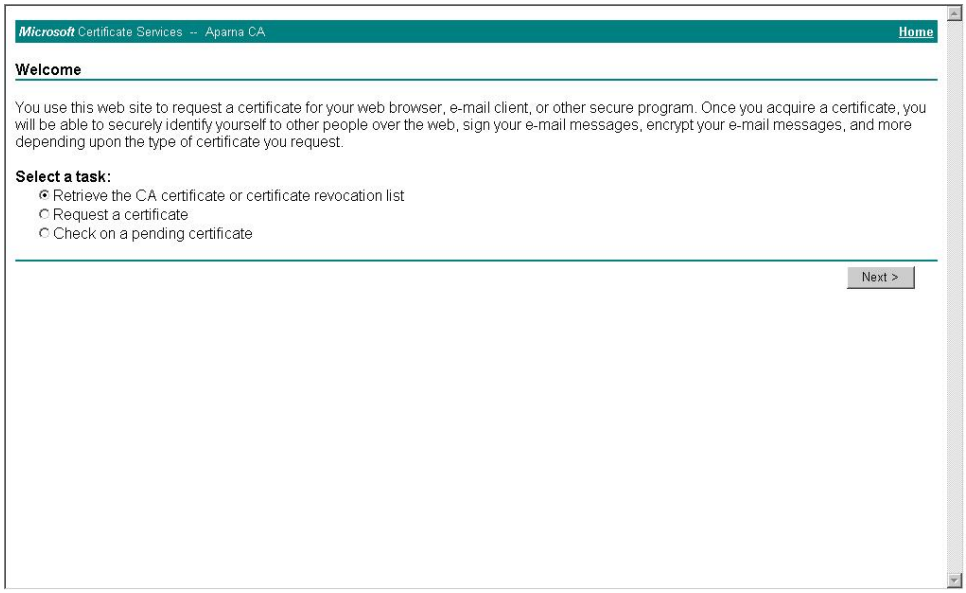


Download the CRL

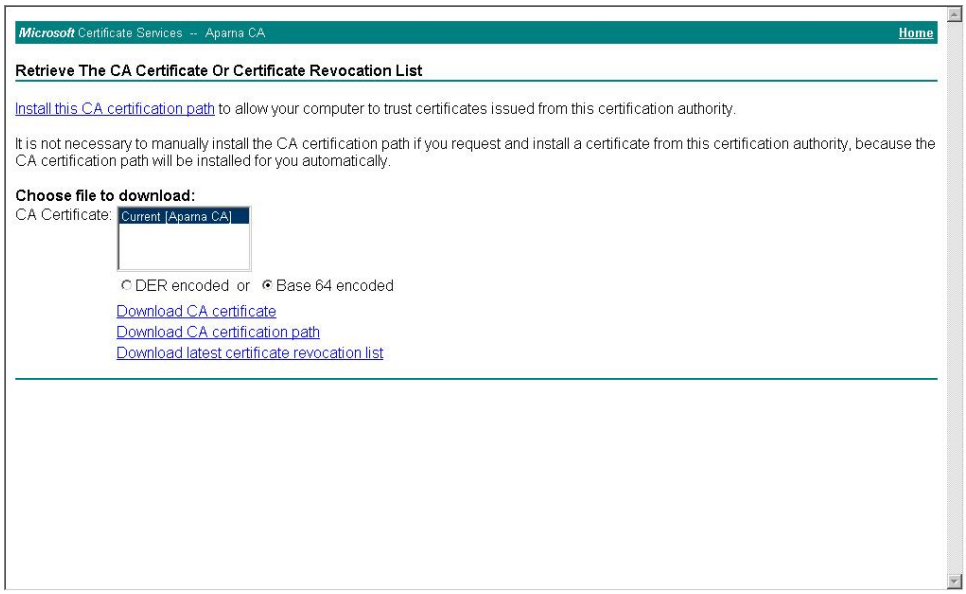
To download the CRL from the Microsoft CA website, follow these steps:

Procedure

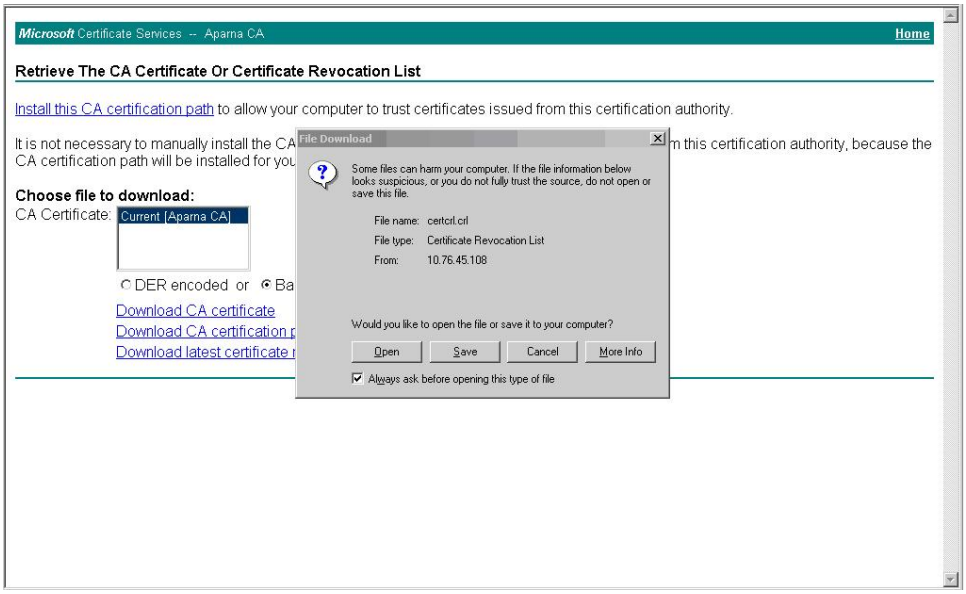
Step 1 From the Microsoft Certificate Services web interface, click **Retrieve the CA certificate or certificate revocation list** and click **Next** .



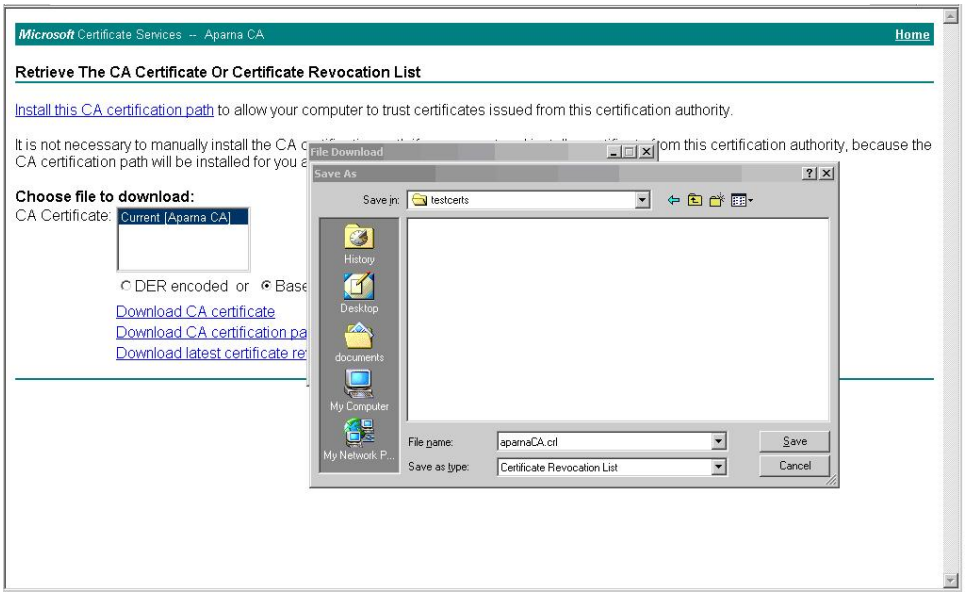
Step 2 Click **Download latest certificate revocation list** .



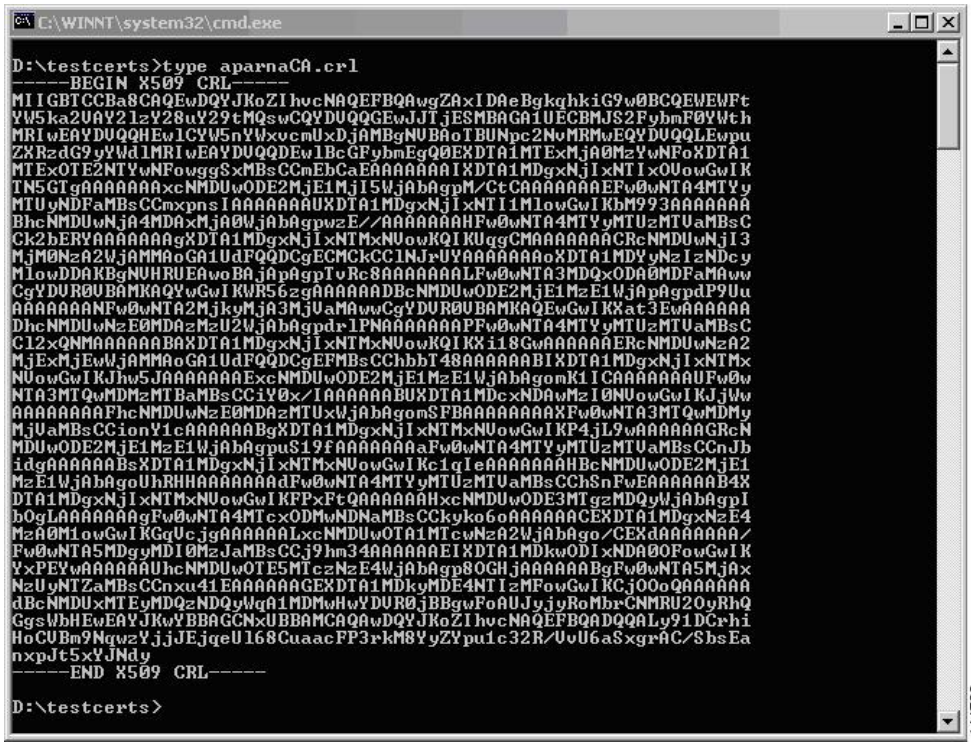
Step 3 In the File Download dialog box, click **Save** .



Step 4 In the Save As dialog box, enter the destination file name and click **Save** .



Step 5 Enter the Microsoft Windows **type** command to display the CRL.



Related Topics

Configure certificate revocation checking methods, on page 199

Import the CRL

To import the CRL to the trust point corresponding to the CA, follow these steps:

Procedure

Step 1 Copy the CRL file to the Cisco NX-OS device bootflash.

Example:

```
Device-1#copy tftp:apranaCA.crl bootflash:aparnaCA.crl
```

Step 2 Configure the CRL.

Example:

```
Device-1#configure terminal
Device-1(config)#crypto ca crl request myCA bootflash:aparnaCA.crl
Device-1(config)#exit
Device-1#
```

Step 3 Display the contents of the CRL.

Example:

```
Device-1#show crypto ca crl myCA
Trustpoint: myCA
CRL:
Certificate Revocation List (CRL):
Version 2 (0x1)
Signature Algorithm: sha1WithRSAEncryption
Issuer: /emailAddress=admin@yourcompany.com/C=IN/ST=Karnatak
Yourcompany/OU=netstorage/CN=Aparna CA
Last Update: Nov 12 04:36:04 2005 GMT
Next Update: Nov 19 16:56:04 2005 GMT
CRL extensions:
X509v3 Authority Key Identifier:
keyid:00:00:5E:00:53:01:00:00:5E:00:53:02:C9:18:50:1
1.3.6.1.4.1.311.21.1:
...
Revoked Certificates:
Serial Number: 611B09A1000000000002
Revocation Date: Aug 16 21:52:19 2005 GMT
Serial Number: 4CDE464E000000000003
Revocation Date: Aug 16 21:52:29 2005 GMT
Serial Number: 4CFC2B42000000000004
Revocation Date: Aug 16 21:52:41 2005 GMT
Serial Number: 6C699EC2000000000005
Revocation Date: Aug 16 21:52:52 2005 GMT
Serial Number: 6CCF7DDC000000000006
Revocation Date: Jun  8 00:12:04 2005 GMT
Serial Number: 70CC4FFF000000000007
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 4D9B1116000000000008
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 52A80230000000000009
Revocation Date: Jun 27 23:47:06 2005 GMT
CRL entry extensions:
X509v3 CRL Reason Code:
CA Compromise
Serial Number: 5349AD4600000000000A
Revocation Date: Jun 27 23:47:22 2005 GMT
CRL entry extensions:
X509v3 CRL Reason Code:
CA Compromise
Serial Number: 53BD173C00000000000B
Revocation Date: Jul  4 18:04:01 2005 GMT
CRL entry extensions:
X509v3 CRL Reason Code:
Certificate Hold
Serial Number: 591E7ACE00000000000C
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 5D3FD52E00000000000D
Revocation Date: Jun 29 22:07:25 2005 GMT
CRL entry extensions:
X509v3 CRL Reason Code:
Key Compromise
Serial Number: 5DAB771300000000000E
Revocation Date: Jul 14 00:33:56 2005 GMT
Serial Number: 5DAE53CD00000000000F
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 5DB140D3000000000010
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 5E2D7C1B000000000011
Revocation Date: Jul  6 21:12:10 2005 GMT
CRL entry extensions:
X509v3 CRL Reason Code:
Cessation Of Operation
Serial Number: 16DB4F8F000000000012
```

```
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 261C3924000000000013
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 262B5202000000000014
Revocation Date: Jul 14 00:33:10 2005 GMT
Serial Number: 2634C7F2000000000015
Revocation Date: Jul 14 00:32:45 2005 GMT
Serial Number: 2635B000000000000016
Revocation Date: Jul 14 00:31:51 2005 GMT
Serial Number: 26485040000000000017
Revocation Date: Jul 14 00:32:25 2005 GMT
Serial Number: 2A276357000000000018
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 3F88CBF7000000000019
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 6E4B5F5F00000000001A
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 725B89D800000000001B
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 735A887800000000001C
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 148511C700000000001D
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 14A7170100000000001E
Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 14FC45B500000000001F
Revocation Date: Aug 17 18:30:42 2005 GMT
Serial Number: 486CE80B000000000020
Revocation Date: Aug 17 18:30:43 2005 GMT
Serial Number: 4CA4A3AA000000000021
Revocation Date: Aug 17 18:30:43 2005 GMT
Serial Number: 1AA55C8E00000000002F
Revocation Date: Sep  5 17:07:06 2005 GMT
Serial Number: 3F0845DD00000000003F
Revocation Date: Sep  8 20:24:32 2005 GMT
Serial Number: 3F619B7E000000000042
Revocation Date: Sep  8 21:40:48 2005 GMT
Serial Number: 6313C463000000000052
Revocation Date: Sep 19 17:37:18 2005 GMT
Serial Number: 7C3861E3000000000060
Revocation Date: Sep 20 17:52:56 2005 GMT
Serial Number: 7C6EE351000000000061
Revocation Date: Sep 20 18:52:30 2005 GMT
Serial Number: 0A338EA1000000000074   <-- Revoked identity certificate
Revocation Date: Nov 12 04:34:42 2005 GMT
Signature Algorithm: sha1WithRSAEncryption
00:00:5E:00:53:03:00:00:5E:00:53:04:00:00:5E:00:53:05:
00:00:5E:00:53:06:00:00:5E:00:53:07:00:00:5E:00:53:08:
00:00:5E:00:53:09:00:00:5E:00:53:0A:00:00:5E:00:53:0B:
00:00:5E:00:53:0C:58:24:d7:72
```

Note
The identity certificate for the device that was revoked (serial number 0A338EA1000000000074) is listed at the end.

Additional references for PKI

This section includes additional information related to implementing PKI.

Related documents for PKI

Use this reference to find related documents for PKI.

Table 40: Related documents for PKI

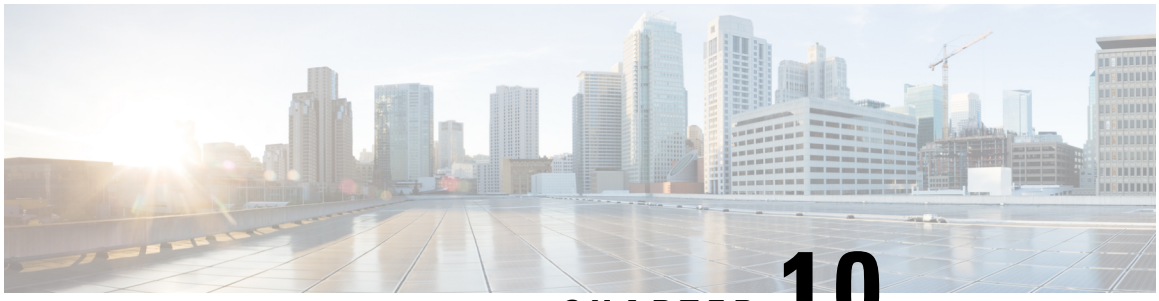
Related topic	Document title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards for PKI

Use this reference to review standards for PKI.

Table 41: Standards for PKI

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	Not applicable



CHAPTER 10

Configuring User Accounts and RBAC

This chapter describes how to configure user accounts and role-based access control (RBAC) on Cisco NX-OS devices.

This chapter includes these sections:

- About user accounts and RBAC, on page 233
- Guidelines and limitations for user accounts and RBAC, on page 236
- Default settings for user accounts and RBAC, on page 237
- Enable Password-Strength checking, on page 238
- Enable consecutive characters check in passwords, on page 238
- Configure user accounts, on page 239
- Configuring roles, on page 242
- About no service Password-Recovery, on page 248
- Enable no service Password-Recovery, on page 248
- User accounts and RBAC configuration verification commands, on page 251
- Configuration examples for user accounts and RBAC, on page 251
- Additional references for user accounts and RBAC, on page 253

About user accounts and RBAC

You can create and manage users accounts and assign roles that limit access to operations on the Cisco NX-OS device. RBAC allows you to define rules for assigned roles that restrict the authorization a user has to access management operations.

User accounts

You can configure up to a maximum of 256 user accounts. By default, the user account does not expire unless you explicitly configure it to expire. The expire option determines the date when the user account is disabled.

These words are reserved and cannot be used to configure users: bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, gopher, ftp, nobody, nscd, mailnull, root, rpc, rpcuser, xfs, gdm, mtsuser, ftpuser, man, and sys.

Note User passwords are not displayed in the configuration files.

Caution Usernames must begin with an alphanumeric character . Usernames can contain only these special characters: (+ = . _ \ -). The #, @, and ! symbols are not supported. If you use any other characters, the specified user cannot log in.

Characteristics of strong passwords

This section explains strong passwords and their essential attributes for securing access to network devices such as NX-OS systems.

A strong password is a password type that

- is at least eight characters long,
- contains both uppercase and lowercase letters and numbers, and
- avoids dictionary words, personal information, consecutive or repeating characters.

Special characters, such as the dollar sign ($) or percent sign (%), can be used in Cisco Nexus device passwords. All printable ASCII characters are allowed if enclosed in quotation marks.

The following are examples of strong passwords:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

Additional reference information

- Clear text passwords cannot start with: quotation marks (" or '), vertical bars (|), or right angle brackets (>).
- Trivial passwords (such as short or easily decoded values) are rejected if password-strength checking is enabled.
- Passwords are case-sensitive.

Related Topics

Enable Password-Strength checking, on page 238

User roles

User roles contain rules that define the operations allowed for the user who is assigned the role. Each user role can contain multiple rules, and each user can have multiple roles. For example, if role1 allows access only to configuration operations, and role2 allows access only to debug operations, then a user who belongs

to both role1 and role2 can access both configuration and debug operations. You can also limit access to specific virtual routing and forwarding instances (VRFs), VLANs, and interfaces.

The Cisco NX-OS software provides these user roles:

- network-admin—Complete read-and-write access to the entire Cisco NX-OS device
- network-operator or vdc-operator—Complete read access to the entire Cisco NX-OS device

Note
- The Cisco Nexus 9000 Series switches support a single VDC. Therefore, the vdc-admin has the same privileges and limitations as the network-admin.
- The Cisco Nexus 9000 Series switches support a single VDC due to which the vdc-admin has the same privileges and limitations as the network-admin.

Note You cannot change the user roles.

Note Some **show** commands may be hidden from network-operator users. Some non- **show** commands, such as **telnet** , may be available for this user role.

By default, the user accounts without an administrator role can access only the **show** , **exit** , **end** , and **configure terminal** commands. You can add rules to allow users to configure features.

Note If you belong to multiple roles, you can execute a combination of all the commands permitted by these roles. Access to a command takes priority over being denied access to a command. For example, suppose a user has RoleA, which denies access to configuration commands, and RoleB, which allows access to configuration commands. In this case, the user has access to configuration commands.

User role rules

The rule is the basic element of a role. A rule defines what operations the role allows the user to perform. You can apply rules for these parameters:

- Command—command or group of commands defined in a regular expression,
- Feature—command or group of commands defined in a regular expression,
- Feature group—default or user-defined group of features, and
- OID—an SNMP object identifier (OID).

The command, feature, and feature group parameters form a hierarchy. The command is the most basic control parameter. The feature parameter represents all commands associated with a feature. The feature group is the

highest-level control parameter, combining related features for simplified rule management. The Cisco NX-OS software supports the predefined feature group L3 for your use.

SNMP OID is supported for RBAC. You can configure a read-only or read-and-write rule for an SNMP OID.

You can configure up to 256 rules for each role. The user-specified rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

Guidelines and limitations for user accounts and RBAC

Use this reference to review the guidelines and limitations for user accounts and RBAC.

User and role limits

- A user role can include up to 256 rules.
- Up to 64 user-defined feature groups can be added in addition to the default group, L3.
- The system supports up to 256 user accounts.
- A maximum of 64 user roles can be assigned to a single user account.

Default accounts and role restrictions

- The default admin and SNMP user accounts cannot be deleted.
- Default user roles cannot be removed from default admin user accounts.
- The **configure terminal** command is always permitted and not affected by role configuration rules.
- The network-operator role does not have access to the **show running-config** or **show startup-config** commands.
- Per AAA policy, a role associated as the last role to a user cannot be deleted until disassociated.

Account synchronization behavior

- If a user account exists locally with the same name as a remote user on an AAA server, local roles take precedence for the remote user.
- Starting with NX-OS 10.2(2)F, a desynchronization CLI is available to disable user synchronization between SNMP and security components.
    - When enabled, remote users are not synced to the SNMP database.
    - Security users created through DCNM, also known as Nexus Dashboard Fabric Controller from Release 12.0.1a, do not have a corresponding SNMPv3 profile when desynchronization is enabled. These users can log in to the switch but are not discovered by the controller, because SNMP configuration is used for discovery. Explicit SNMP user creation is then required. Use of desynchronization CLI with DCNM is not recommended.

Platform and feature support

- On Cisco Nexus 9000 Series switches, a single VDC is supported; thus, the vdc-admin has the same privileges as network-admin.
- For information about Cisco Nexus 9000 switches with feature support by release, refer to the Nexus Switch Platform Support Matrix.

Password feature support

- Starting with NX-OS 10.3(1)F, type 8 and 9 password hashes are supported. Type 8 and 9 hashes cannot be downgraded, while type 5 is backward compatible.
- Beginning with NX-OS 10.3(1)F, a consecutive-character check in passwords is enforced.

CLI interoperability notes

- Commands for this feature may differ from those used in Cisco IOS CLI.

Further references

- For SNMP configuration, see the "Configuring SNMP" chapter in the System Management Configuration Guide.
- For security user configuration, see the Cisco Nexus 9000 NX-OS Security Configuration Guide.

Default settings for user accounts and RBAC

Use this reference to review the default settings for user accounts and RBAC.

This table lists the default settings for user accounts and RBAC parameters.

Table 42: Default user accounts and RBAC parameters

Parameters	Default
User account password	Undefined
User account expiry date	None
User account role	Network-operator, if the creating user has the network-admin role
Default user role	Network-operator
Interface policy	All interfaces are accessible
VLAN policy	All VLANs are accessible
VRF policy	All VRFs are accessible
Feature group	L3

Enable Password-Strength checking

You can enable password-strength checking. This feature prevents the creation of weak passwords for user accounts.

Note When you enable password-strength checking, the Cisco NX-OS software does not check the strength of existing passwords.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enable password-strength checking.

Example:

```
switch(config)#password strength-check
```

Password-strength checking is enabled by default. To disable it, use the **no** form of this command.

Step 3 Exit the global configuration mode.

Example:

```
switch(config)#exit
switch#
```

Step 4 (Optional) Display the password-strength check configuration.

Example:

```
switch#show password strength-check
```

Step 5 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Characteristics of strong passwords, on page 234

Enable consecutive characters check in passwords

Strengthen password policies by restricting the use of consecutive and repeating characters in user passwords.

This configuration helps defend against attackers who exploit weak, easily guessed password patterns. By limiting repeated and consecutive characters, the risk of password compromise is reduced.

Before you begin

- Access to the device with the appropriate configuration privilege.
- Know the acceptable range for each sequence check (e.g. 2–10 characters).

Procedure

Step 1 Enter configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Set the maximum number of sequential alphabetical characters allowed.

Example:

```
switch(config)#userpassphrase sequence alphabet length 4
```

This limits the use of alphabetical sequences (e.g. 'abcd') to the specified length; passwords with longer sequences are not accepted.

To disable this check, use the **no userpassphrase sequence alphabet length** command.

Step 3 Set the maximum number of sequential keyboard characters allowed.

Example:

```
switch(config)#userpassphrase sequence keyboard length 4
```

This limits the use of keyboard patterns (e.g. 'qwer', 'asdf').

To disable this check, use the **no userpassphrase sequence keyboard length** command.

Passwords with more than the allowed consecutive or sequential characters are rejected, increasing overall password security.

What to do next

Test password creation with different character sequences to confirm the enforcement works as intended.

Configure user accounts

You can create a maximum of 256 user accounts on a Cisco NX-OS device. User accounts have these attributes:

- Username
- Password
- Expiry date

- User roles

You can enter the password in clear text format or encrypted format. The Cisco NX-OS password encrypts clear text passwords before saving them to the running configuration. Encrypted format passwords are saved to the running configuration without further encryption.

SHA256 is the hashing algorithm used for password encryption. As part of the encryption, 5000 iterations of a 64-bit SALT are added to the password.

SHA256 is the default hashing algorithm used for password encryption. To generate a hash for a type 8 or type 9 password, you must provide the PBKDF2 or SCRYPT option along with the clear text password.

User accounts can have a maximum of 64 user roles. The user can determine available commands by using the CLI context-sensitive help utility.

Note Changes to user account attributes do not take effect until the user logs in and creates a new session.

Procedure

Step 1 (Optional) Display the available user roles.

Example:

```
switch#show role
```

You can configure other user roles, if necessary.

Step 2 Enter the global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 3 Configure a user account.

Example:

```
switch(config)#username NewUser password 4Ty18Rnt
```

The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters for both local and remote users . Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames.

Usernames must begin with an alphanumeric character.

The default password is undefined.

- The **0** option indicates that the password is clear text
- The **5** option indicates that the password is SHA-256 hashed.
- The **8** option indicates that the password is PBKDF2 hashed.
- The **9** option indicates that the password is Scrypt hashed.

The default option is **0** (clear text).

Note
The pbkdf2 and scrypt keywords are optional and are not stored in running configurations.

Note
If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device.

Note
If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.

Note
When the desynchronization CLI is enabled, if you create a user account, the corresponding SNMP user will not be created.

The **expire** *date* option format is YYYY-MM-DD. The default is no expiry date.

User accounts can have a maximum of 64 user roles.

Step 4 Specify if an SSH X.509 certificate distinguished name and DSA or RSA algorithm must be used to authenticate an existing user account.

Example:

```
switch(config)#username NewUser ssh-cert-dn "/CN = NewUser, OU = Cisco Demo, O = Cisco, C = US" rsa
```

Example:

```
switch(config)#username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1,
emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith"
rsa
```

The distinguished name can be up to 512 characters and must use the format shown in the examples. Make sure the email address and state are configured as "emailAddress" and "ST", respectively.

Step 5 Exit the global configuration mode.

Example:

```
switch(config)#exit
switch#
```

Step 6 (Optional) Display the role configuration.

Example:

```
switch#show user-account
```

Step 7 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configuring roles, on page 242
Configure user roles and rules, on page 242

Configuring roles

This section describes how to configure user roles.

Configure user roles and rules

- Use this procedure to create user roles and rules in NX-OS.

You can configure up to 64 user roles. Each user role can have up to 256 rules. You can assign a user role to more than one user account.

The rule number that you specify determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

When processing an RBACL for a match, a partial match does not stop the evaluation process. Evaluation continues through each rule until an exact match is found. If no exact match is found, the most precise rule in the list will be chosen for the result. Also, if a permit and deny rule exists for the same match logic, the higher numbered rule (evaluated first) will be chosen for the result.



Note Regardless of the read-write rule configured for a user role, some commands can be executed only by using the predefined network-admin role.

Before you begin

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify a user role and enters role configuration mode.

Example:

```
switch(config)#role name UserA
switch(config-role)#
```

The *role-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 16 characters.

Step 3 Configure from these available options.

- Configure a command rule.
- Configure a read-only or read-and-write rule for all operations.

- Configure a read-only or read-and-write rule for a feature.
- Configure a read-only or read-and-write rule for a feature group.
- Configure a read-only or read-and-write rule for an SNMP object identifier (OID).

Example:

```
switch(config-role)#rule 1 deny command clear users
switch(config-role)#rule 2 deny read-write
switch(config-role)#rule 3 permit read feature router-bgp
switch(config-role)#rule 4 deny read-write feature-group L3
switch(config-role)#rule 5 deny read-write oid 1.3.6.1.2.1.1.9
```

The *command-string* argument can contain spaces and regular expressions. For example, interface ethernet includes all Ethernet interfaces.

Use the **show role feature** command to display a list of features.

Use the **show role feature-group** command to display a list of feature groups.

You can enter up to 32 elements for the OID. The **rule <num> deny read-write oid** command can be used to allow SNMP-based performance monitoring tools to poll devices but restrict their access to system-intensive branches such as the IP routing table, MAC address tables, specific MIBs, and so on.

Note
The deepest OID can be at the scalar level or at the table root level.

Repeat the commands for each rule you need to configure.

Step 4 (Optional) Configure the role description. You can include spaces in the description.

Example:

```
switch(config-role)#description This role does not allow users to use clear commands
```

Step 5 Exit the role configuration mode.

Example:

```
switch(config-role)#exit
switch(config)#
```

Step 6 (Optional) Display the user role configuration.

Example:

```
switch#show role
```

Step 7 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Create feature groups

You can create custom feature groups to add to the default list of features provided by the Cisco NX-OS software. These groups contain one or more of the features. You can create up to 64 feature groups.

Note You cannot change the default feature group L3.

Before you begin

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify a user role feature group and enters role feature group configuration mode.

Example:

```
switch(config)#role feature-group name GroupA
switch(config-role-featuregrp)#
```

The *group-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 32 characters.

Step 3 Specify a feature for the feature group.

Example:

```
switch(config-role-featuregrp)#feature radius
```

Repeat this command for as many features as needed.

Note
Use the **show role component** command to display a list of features.

Step 4 Exit the role feature group configuration mode.

Example:

```
switch(config-role-featuregrp)#exit
switch(config)#
```

Step 5 (Optional) Display the role feature group configuration.

Example:

```
switch#show role feature-group
```

Step 6 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Change user role interface policies

You can change a user role interface policy to limit the interfaces that the user can access. By default, a user role allows access to all interfaces.

Before you begin

Create one or more user roles.

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify a user role and enters role configuration mode.

Example:

```
switch(config)#role name UserA
switch(config-role)#
```

Step 3 Enter the role interface policy configuration mode.

Example:

```
switch(config-role)#interface policy deny
switch(config-role-interface)#
```

Step 4 Specify a list of interfaces that the role can access.

Example:

```
switch(config-role-interface)#permit interface ethernet 2/1-4
```

Repeat this command for as many interfaces as needed.

Step 5 Exit the role interface policy configuration mode.

Example:

```
switch(config-role-interface)#exit
switch(config-role)#
```

Step 6 (Optional) Display the role configuration.

Example:

```
switch#show role
```

Step 7 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure user roles and rules, on page 242

Change user role VLAN policies

You can change a user role VLAN policy to limit the VLANs that the user can access. By default, a user role allows access to all VLANs.

Before you begin

Create one or more user roles.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify a user role and enters role configuration mode.

Example:

```
switch(config)#role name UserA
switch(config-role)#
```

Step 3 Enter role VLAN policy configuration mode.

Example:

```
switch(config-role)#vlan policy deny
switch(config-role-vlan)#
```

Step 4 Specify a range of VLANs that the role can access.

Example:

```
switch(config-role-vlan)#permit vlan 1-4
```

Repeat this command for as many VLANs as needed.

Step 5 Exit the role VLAN policy configuration mode.

Example:

```
switch(config-role-vlan)#exit
switch(config-role)#
```

Step 6 (Optional) Display the role configuration.

Example:

```
switch#show role
```

Step 7 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure user roles and rules, on page 242

Change user role VRF policies

You can change a user role VRF policy to limit the VRFs that the user can access. By default, a user role allows access to all VRFs.

Before you begin

Create one or more user roles.

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **role name** *role-name*

Example:

```
switch(config)#role name UserA
switch(config-role)#
```

Specifies a user role and enters role configuration mode.

Step 3 **vrf policy deny**

Example:

```
switch(config-role)#vrf policy deny
switch(config-role-vrf)#
```

Enters role VRF policy configuration mode.

Step 4 **permit vrf** *vrf-name*

Example:

```
switch(config-role-vrf)#permit vrf vrf1
```

Specifies the VRF that the role can access.

Repeat this command for as many VRFs as needed.

Step 5 **exit**

Example:

```
switch(config-role-vrf)#exit
switch(config-role)#
```

Exits role VRF policy configuration mode.

Step 6 (Optional) **show role**

Example:

```
switch#show role
```

Displays the role configuration.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Configure user roles and rules, on page 242

About no service Password-Recovery

The No Service Password-Recovery feature enables anyone with console access, the ability to access the router and its network.

The No Service Password-Recovery feature prevents the password recovery with standard procedure as described in the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide.

Enable no service Password-Recovery

If the no service password-recovery feature is enabled, then none except the administrator with network privileges will be able to modify the administrator password.

Before you begin

If you plan to enter the no service password-recovery command, Cisco recommends that you save a copy of the system configuration file in a location away from the device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no service password-recovery**

Example:

```
switch(config)#no service password-recovery
WARNING: Executing this command will disable the password recovery mechanism. Do not execute
 this command without another plan for password recovery. Are you sure you want to continue?
 (y/n) : [y] y
```

Disables the password recovery mechanism.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 4 **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Step 5 **reload**

Example:

```
switch#reload
This command will reboot the system. (y/n)?  [n] y
2018 Jun 26 16:23:19 BAR %$ VDC-1 %$ %PLATFORM-2-PFM_SYSTEM_RESET: Manual system restart
from Command Line Interface

CISCO SWITCH Ver 8.34

CISCO SWITCH Ver 8.34
Manual system restart from Command Line Interface
writing reset reason 9,
..
..
GNU GRUB  version 0.97
Autobooting bootflash:/n9000-dk9.x.x.x.bin bootflash:/n...
Filesystem type is ext2fs, partition type 0x83
Booting nx-os image: bootflash:/n9000-dk9.x.x.x.bin....(---->Press Ctrl + C)
....Aborting Image Boot
GNU GRUB  version 0.97
Loader Version 3.22.0
loader>cmdline recoverymode=1
loader>boot nxos.9.2.0.145.bin
Booting nxos.9.2.0.145.bin
Trying diskboot
Filesystem type is ext2fs, partition type 0x83
IOFPGA ID: 1168153
Image valid


Image Signature verification was Successful.

Boot Time: 6/26/2018  16:26:52
Installing klm_card_index
done
Linking n9k flash devices
creating flash devices BOOT_DEV= sda
INIT: version 2.88 booting
```

```
Installing ata_piix module ... done.
Unsquashing rootfs ...
Enabling 8250 serial driver spurious INTs workaround
Installing isan procfs ... done.
is_lxc: is_stby: suffix: klm_ftrace: /isanboot/lib/modules/klm_ftrace.o
Installing ftrace in non-lxc mode done
Installing SSE module with card index 21000 ... done.
Creating SSE device node 247 ... done.
Loading I2C driver ... done.
Loading MEM scrub driver ... done.
Installing CCTRL driver for card_type 3 without NEED_GEM ... done.
old data: 4000004 new data: 1
Loading IGB driver ... done.
Checking SSD firmware ...
Model Number:       C400-MTFDDAT064MAM
Serial Number:      MSA173407DW
Firmware Revision:  0BMH

Checking all filesystems.
Installing SPROM driver ... IS_N9K done.

loader>cmdline recoverymode=1
loader>boot n9000-dk9.x.x.x.bin
Example:
loader>boot n9000-dk9.x.x.x.bin
Booting iash
Trying diskboot
Filesystem type is ext2fs, partition type 0x83
Image valid
MD5Sum mismatch

INIT: Loading IGB driver ... Signature Envelope.(36)Invalid Tag in Signature Envelope
Installing SSE module ... done
Creating the sse device node ... done
Installing CCTRL driver for card_type 3 ...

Checking all filesystems.......
Installing SPROM driver ...
Installing default sprom values ...

----------------------------snip-----------------------------------
the GNU General Public License (GPL) version 2.0 or
GNU General Public License (GPL) version 3.0  or the GNU
Lesser General Public License (LGPL) Version 2.1 or
Lesser General Public License (LGPL) Version 2.0.
A copy of each such license is available at
http://example.com/licenses/gpl-2.0.php and
http://example.com/licenses/gpl-3.0.html and
http://example.com/licenses/lgpl-2.1.php and
http://example.com/licenses/old-licenses/library.txt.
switch(boot)#config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(boot)(config)# admin-password Abcd!123$
ERROR: service password-recovery disabled. Cannot change password!
switch(boot)(config)#
```

Step 6 **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Step 7 (Optional) **show user-account**

Example:

```
switch#show user-account
```

Displays user account information.

User accounts and RBAC configuration verification commands

Use this reference to verify the User accounts and RBAC configuration.

To display user account and RBAC configuration information, perform one of the following tasks:

Table 43: User accounts and RBAC configuration verification commands

Command	Purpose
show cli syntax roles network-admin	Displays the syntax of the commands that the network-admin role can use.
show cli syntax roles network-operator	Displays the syntax of the commands that the network-operator role can use.
show role	Displays the user role configuration.
show role feature	Displays the feature list.
show role feature-group	Displays the feature group configuration.
show startup-config security	Displays the user account configuration in the startup configuration.
show running-config security [**all**]	Displays the user account configuration in the running configuration. The **all** keyword displays the default values for the user accounts.
show user-account	Displays user account information.

Configuration examples for user accounts and RBAC

The following example shows how to configure a user role:

```
role name User-role-A
  rule 2 permit read-write feature bgp
  rule 1 deny command clear *
```

The following example shows how to create a user role that can configure an interface to enable and show BGP and show EIGRP:

```
role name iftest
      rule 1 permit command config t; interface *; bgp *
      rule 2 permit read-write feature bgp
      rule 3 permit read feature eigrp
```

In the above example, rule 1 allows you to configure BGP on an interface, rule 2 allows you to configure the **config bgp** command and enable the exec-level **show** and **debug** commands for BGP, and rule 3 allows you to enable the exec-level **show** and **debug eigrp** commands.

The following example shows how to configure a user role that can configure only a specific interface:

```
role name Int_Eth2-3_only
  rule 1 permit command configure terminal; interface *
  interface policy deny
    permit interface Ethernet2/3
```

The following example shows how to configure a user role feature group:

```
role feature-group name Security-features
  feature radius
  feature tacacs
  feature aaa
  feature acl
  feature access-list
```

The following example shows how to configure a user account:

```
username user1 password A1s2D4f5 role User-role-A
```

The following example shows how to add an OID rule to restrict access to part of the OID subtree:

```
role name User1
        rule 1 permit read feature snmp
        rule 2 deny read oid 1.3.6.1.2.1.1.9
show role name User1

Role: User1
  Description: new role
  Vlan policy: permit (default)
  Interface policy: permit (default)
  Vrf policy: permit (default)
  ----------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ----------------------------------------------------------------------
  2       deny    read        oid                 1.3.6.1.2.1.1.9
  1       permit  read        feature             snmp
```

The following example shows how to give write permission to a specified OID subtree:

```
role name User1
rule 3 permit read-write oid 1.3.6.1.2.1.1.5
show role name User1

Role: User1
  Description: new role
  Vlan policy: permit (default)
  Interface policy: permit (default)
  Vrf policy: permit (default)
  ----------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ----------------------------------------------------------------------
  3       permit  read-write  oid                 1.3.6.1.2.1.1.5
```

```
2       deny    read        oid                 1.3.6.1.2.1.1.9
1       permit  read        feature             snmp
```

Additional references for user accounts and RBAC

Use this reference to find additional references for user accounts and RBAC.

This section includes additional information related to implementing user accounts and RBAC.

Related Documents

Table 44: Related documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards

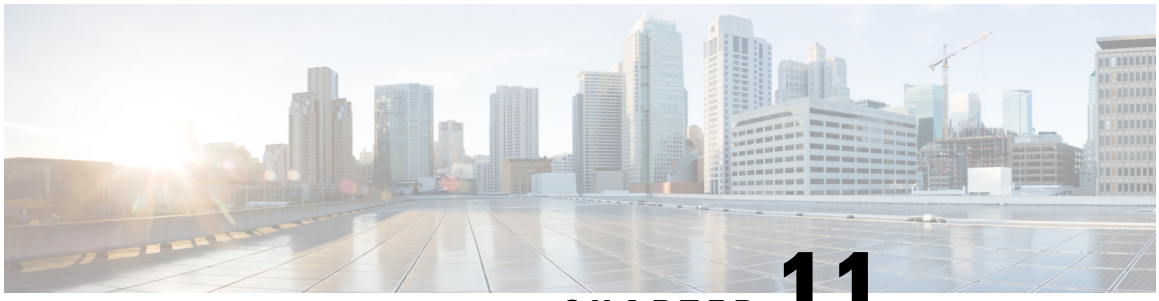
Table 45: Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

Table 46: MIBs

MIBs	MIBs Link
MIBs related to user accounts and RBAC	To locate and download supported MIBs, go to the following URL: https://example.com/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 11

Configuring 802.1X

This chapter describes how to configure IEEE 802.1X port-based authentication on Cisco NX-OS devices.

This chapter includes the following sections:

- About 802.1x, on page 255
- About DACL, on page 264
- Prerequisites for 802.1x, on page 265
- 802.1X guidelines and limitations, on page 265
- Guidelines and limitations for Per-User DACL support for 802.1x, on page 269
- Guidelines and limitations for critical authentication, on page 270
- Default settings for 802.1x, on page 270
- Configuring 802.1x, on page 271
- 802.1X configuration verification commands, on page 296
- 802.1X support for VXLAN EVPN, on page 298
- Critical authentication verification, on page 302
- Monitor 802.1x statistics, on page 303
- 802.1X port configuration commands, on page 303
- Configuration example for Per-User DACL, on page 305
- Additional references for 802.1x, on page 306

About 802.1x

802.1X is a client-server based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports.

- Authenticates each client connected to a Cisco NX-OS device port using an authentication server.
- Allows only Extensible Authentication Protocol over LAN (EAPOL) traffic until authentication is successful.
- Permits normal traffic through the port after successful authentication.

Device roles

Device roles in 802.1X define the responsibilities of network devices in port-based authentication, including the supplicant, authenticator, and authentication server.

- **Supplicant** : The client device that requests access to the LAN and responds to requests from the Cisco NX-OS device, requiring 802.1X-compliant client software.
- **Authentication server** : Performs the actual authentication of the supplicant, validates its identity, and notifies the Cisco NX-OS device about authorization. RADIUS with EAP extensions is the only supported authentication server.
- **Authenticator** : Controls physical access to the network based on the supplicant's authentication status, acts as an intermediary between the supplicant and authentication server, and includes the RADIUS client for EAP frame handling.

Details of device roles in 802.1x authentication

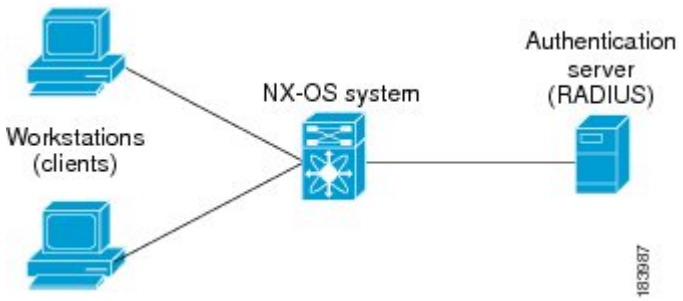
With 802.1X port-based authentication, devices in the network have specific roles that determine how access is granted and managed.

- The supplicant is typically a workstation or client device running 802.1X-compliant software, such as Microsoft Windows XP.
- The authentication server is usually a RADIUS server with EAP extensions, such as Cisco Secure Access Control Server, version 3.0.
- The authenticator is the Cisco NX-OS device, which acts as a proxy between the supplicant and the authentication server.

1. When the authenticator receives EAPOL frames from the supplicant, it strips the Ethernet header and encapsulates the EAP frame in RADIUS format for the authentication server.
2. The authentication server processes the EAP frame and sends a response back to the authenticator.
3. The authenticator removes the server's frame header, encapsulates the EAP frame for Ethernet, and sends it to the supplicant.

Note The Cisco NX-OS device can only be an 802.1X authenticator.

Figure 6: 802.1X device roles



Device roles in 802.1x

For example, when a user connects a workstation (supplicant) to a network port, the Cisco NX-OS device (authenticator) requests identity information. The authentication server (RADIUS) validates the credentials and authorizes access if the supplicant is verified.

Authentication initiation and message exchange

Authentication initiation and message exchange describe the process by which either the authenticator (Cisco NX-OS device) or the supplicant (client) starts the authentication process, and how EAP frames are exchanged to establish network access.

- Authentication can be initiated by either the authenticator or the supplicant.
- The authenticator sends EAP-request/identity frames when a port transitions from down to up.
- The supplicant responds with EAP-response/identity frames or can initiate authentication with an EAPOL-start frame if no request is received.

Details of authentication initiation and message exchange

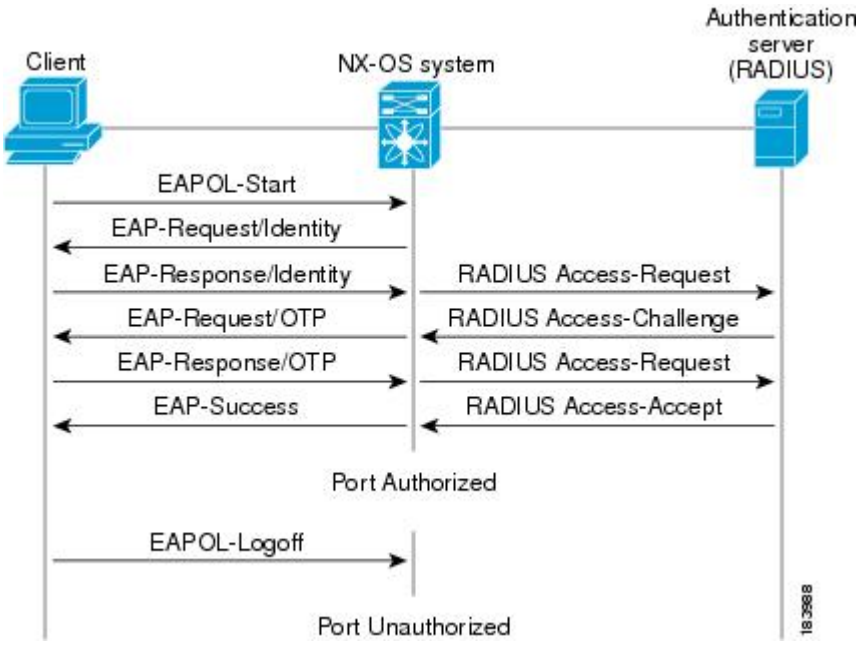
Authentication on Cisco NX-OS devices can be initiated by either the authenticator or the supplicant. When authentication is enabled on a port and the link state transitions from down to up, the authenticator sends an EAP-request/identity frame to the supplicant. The supplicant responds with an EAP-response/identity frame. If the supplicant does not receive a request, it can initiate authentication by sending an EAPOL-start frame. The authenticator then acts as an intermediary, passing EAP frames between the supplicant and the authentication server until authentication succeeds or fails. If authentication is successful, the port becomes authorized.



Note If 802.1X is not enabled or supported on the network access device, the Cisco NX-OS device drops any EAPOL frames from the supplicant. If the supplicant does not receive an EAP-request/identity frame after three attempts to start authentication, the supplicant transmits data as if the port is in the authorized state. A port in the authorized state means that the supplicant has been successfully authenticated.

The specific exchange of EAP frames depends on the authentication method being used.

Figure 7: Message Exchange



The user's secret pass-phrase never crosses the network at any time such as during authentication or during pass-phrase changes.

Authentication message exchange

This example shows a message exchange initiated by the supplicant using the One-Time-Password (OTP) authentication method with a RADIUS server. The OTP authentication device uses a secret pass-phrase to generate a sequence of one-time (single use) passwords.

Authenticator PAE status for interfaces

An authenticator PAE (Port Access Entity) is a protocol entity that supports authentication on an interface when 802.1X is enabled.

- Created automatically when 802.1X is enabled on an interface.
- Not automatically cleared when 802.1X is disabled on the interface.
- Can be explicitly removed and reapplied as needed.

When 802.1X is enabled on an interface, the Cisco NX-OS software creates an authenticator PAE instance to support authentication. Disabling 802.1X does not automatically remove the PAE instance; manual removal and reapplication may be required.

Managing authenticator PAE instances

For example, after disabling 802.1X on an interface, you may need to explicitly remove the authenticator PAE before re-enabling it to ensure proper authentication behavior.

Ports in authorized and unauthorized states

The port authorization state determines whether a supplicant is granted access to the network. Ports can be in authorized or unauthorized states, which control the flow of network traffic based on authentication status.

- In the unauthorized state, the port blocks all ingress and egress traffic except for 802.1X protocol packets.
- When a supplicant is successfully authenticated, the port transitions to the authorized state, allowing all traffic for the supplicant.
- Ports support three authorization states: Force authorized, Force unauthorized, and Auto.

Authorization states and port behavior

Ports can operate in different authorization states, each affecting how authentication and network access are managed.

- **Force authorized** : Disables 802.1X port-based authentication and transitions to the authorized state without requiring any authentication exchange. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This authorization state is the default.
- **Force unauthorized** : Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The authenticator cannot provide authentication services to the client through the interface.
- **Auto** : Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received from the supplicant. The authenticator requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each supplicant that attempts to access the network is uniquely identified by the authenticator by using the supplicant's MAC address.

When a client that does not support 802.1X is connected to an unauthorized 802.1X port, the authenticator requests the client's identity. If the client does not respond, the port remains in the unauthorized state and the client is not granted access to the network.

When an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. If no response is received, the client retries a fixed number of times and then begins sending frames as if the port is in the authorized state.

If the supplicant is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated supplicant are allowed through the port. If authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the authenticator can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and the supplicant is not granted network access.

When a supplicant logs off, it sends an EAPOL-logoff message, which causes the authenticator port to transition to the unauthorized state. If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

Port state transitions

For example, if a supplicant is authenticated successfully, the port transitions to the authorized state and allows all traffic. If authentication fails or the supplicant logs off, the port returns to the unauthorized state, restricting network access.

If a non-802.1X client is connected to an unauthorized port, it does not respond to identity requests, and the port remains unauthorized, denying network access.

Port authorization states are like a security checkpoint: only authenticated individuals (supplicants) are allowed through (authorized state), while others are stopped at the gate (unauthorized state).

MAC authentication bypass

MAC Authentication Bypass is a feature that enables a Cisco NX-OS device to authorize a supplicant based on its MAC address when 802.1X authentication is not available or times out.

- Allows network access for devices without 802.1X capability by using their MAC address as identity.
- Triggers when 802.1X authentication times out or is not supported by the connected device.
- Relies on an authentication server with a database of permitted MAC addresses for authorization.

How MAC authentication bypass works and interacts with other features

MAC Authentication Bypass (MAB) is used to authorize devices on Cisco NX-OS interfaces when 802.1X authentication is unavailable or times out. The device uses the MAC address as the supplicant identity and communicates with the authentication server to grant or deny network access.

- When 802.1X authentication times out, the device attempts MAB.
- If an EAPOL packet is detected, 802.1X authentication is used instead of MAB.
- Reauthentication can occur for clients authorized by MAB, following the same process as 802.1X clients.
- MAB interacts with features such as 802.1X authentication, port security, and Network Admission Control (NAC) Layer 2 IP validation.

1. Enable 802.1X authentication on the port.
2. If 802.1X times out, MAB is triggered and the MAC address is used for authorization.
3. If EAPOL is detected, revert to 802.1X authentication.

- 802.1X authentication: MAB can only be enabled if 802.1X is enabled on the port.
- Port security: 802.1X authentication and port security cannot be configured on the same Layer 2 ports.
- NAC Layer 2 IP validation: Takes effect after an 802.1X port is authenticated with MAB, including hosts in the exception list.

Comparison of authentication methods and their triggers:

Table 47: Authentication method comparison

Attributes	802.1X Authentication	MAC Authentication Bypass
Trigger	EAPOL packet detected	802.1X timeout or not supported
Identity Used	Supplicant credentials	MAC address

Tip Enable MAC Authentication Bypass only if 802.1X authentication is enabled on the port, and do not configure port security on the same Layer 2 ports.

Using MAC authentication bypass for a printer

For example, if a printer is connected to an interface configured for 802.1X and does not support 802.1X, the Cisco NX-OS device can use MAC Authentication Bypass to authorize the printer based on its MAC address, granting it network access.

If a device connected to the interface is 802.1X-capable and sends an EAPOL packet, the Cisco NX-OS device uses 802.1X authentication instead of MAC Authentication Bypass.

MAC Authentication Bypass acts like a backup key for network access, allowing entry when the primary method (802.1X) is unavailable or fails.

Dynamic VLAN assignment based on MAC-Based authentication (MAB)

Dynamic VLAN assignment based on MAC-Based Authentication (MAB) is a process where a switch assigns a VLAN to a device after successful authentication, as directed by a RADIUS server.

- Supports dynamic VLAN assignment on Cisco Nexus 9000 Series switches.
- Uses 802.1x or MAB for authentication before port activation.
- RADIUS server specifies the VLAN using tunnel attributes in the Access-Accept message.

How dynamic VLAN assignment works with MAB

Dynamic VLAN assignment allows a switch to place authenticated devices into specific VLANs as determined by a RADIUS server, enhancing network segmentation and security.

Dynamic VLAN assignment with MAB

For example, when a device connects to a Cisco Nexus 9000 Series switch port, the switch uses MAB to authenticate the device's MAC address. Upon successful authentication, the RADIUS server responds with tunnel attributes specifying the VLAN, and the switch assigns the port to that VLAN dynamically.

VLAN assignment from RADIUS

VLAN assignment from RADIUS is the process of dynamically assigning a VLAN to a port based on tunnel attributes received from the RADIUS server after successful authentication.

- The RADIUS server sends tunnel attributes in the Accept-Access message.
- The required tunnel attributes for VLAN assignment are Tunnel-type=VLAN(13), Tunnel-Medium-Type=802, and Tunnel-Private-Group-ID=VLANID.
- All three parameters must be received to configure the access VLAN.

After authentication via 802.1X or MAB, the RADIUS server can provide dynamic VLAN information in the Accept-Access message using tunnel attributes.

- **Tunnel-type=VLAN(13)**
- **Tunnel-Medium-Type=802**
- **Tunnel-Private-Group-ID=VLANID**

VLAN assignment from RADIUS

For example, when a device authenticates using 802.1X, the RADIUS server responds with the required tunnel attributes, allowing the switch to assign the appropriate VLAN to the port dynamically.

Single host and multiple hosts support

Single-host and multiple-host support in 802.1X define how many endpoint devices can access a port and how authentication and security violations are handled.

- Single-host mode restricts port access to only one authenticated endpoint device.
- Multiple-host mode allows multiple endpoint devices to access a port after the first device is authenticated.
- Security violations are handled differently in each mode, with stricter enforcement in single-host mode.

Details of single host and multiple hosts support in 802.1x

The 802.1X feature can restrict traffic on a port to only one endpoint device (single-host mode) or allow traffic from multiple endpoint devices on a port (multi-host mode).

- Single-host mode: Only one endpoint device is allowed on the port. After authentication, the port is authorized. If the device leaves, the port becomes unauthorized. Any traffic from a different MAC address triggers a security violation, disabling the interface. This mode is used for host-to-switch topologies and applies to both Layer 2 and Layer 3 ports.
- Multiple-host mode: Only the first host must be authenticated. Once authorized, additional hosts can access the network without separate authentication. If the port becomes unauthorized, all hosts lose access. Security violation shutdown is disabled in this mode. It is applicable to both switch-to-switch and host-to-switch topologies.

Single host and multiple hosts support

For example, in single-host mode, a single computer connected to a switch port must authenticate before gaining network access. In multiple-host mode, once the first device authenticates, other devices connected to the same port can access the network without additional authentication.

Supported topology

Supported topology for 802.1X port-based authentication refers to the network configuration in which the authentication protocol operates.

- Supports point-to-point topology.
- Only one supplicant (client) can connect to the 802.1X-enabled authenticator port at a time.
- The authenticator detects the supplicant when the port link state changes to up, and returns to unauthorized state if the supplicant leaves or is replaced.

The 802.1X port-based authentication is designed for point-to-point topologies, ensuring that only a single client is authenticated per port.

For example, when a client device connects to a Cisco NX-OS device port configured for 802.1X, the device authenticates the client. If the client disconnects and another device connects, the port transitions to the unauthorized state until the new device is authenticated.

About Per-User DACLs

Per-user DACLs are dynamic access control lists that are downloaded from the Cisco ISE Server and applied to authenticated users on 802.1X ports in Cisco NX-OS.

- They provide different levels of network access and service to 802.1X-authenticated users.
- The switch applies the DACL attributes for the duration of the user session and removes them when the session ends or authentication fails.
- Per-user DACLs use RADIUS vendor-specific attributes (VSAs) in the octet-string format, specifically `inacl#<n>` for ingress direction.

Per-user DACLs are configured and enforced using RADIUS authentication and vendor-specific attributes (VSAs) in Cisco NX-OS environments.

- VSAs are used in the ingress direction only.
- The syntax for per-user DACLs is: `ip:inacl#<n>=permit | deny [protocol] [source_subnet] [dest_subnet] [operator][port]`

Table 48: Per-user DACL VSA syntax comparison

Attribute	Example 1	Example 2
VSA Syntax	`ip:inacl#1=permit udp any any eq 5555`	`ip:inacl#2=deny udp any any eq 6666`



Note TIP: The switch supports VSAs only in the ingress direction.

Per-user DACL example

For example, to permit UDP traffic on port 5555 for a user, use `ip:inacl#1=permit udp any any eq 5555` . To deny UDP traffic on port 6666, use `ip:inacl#2=deny udp any any eq 6666` .

Critical authentication

Critical Authentication is a feature that allows 802.1X users to access the network when RADIUS or ISE servers are unreachable or authentication fails.

- Permits network access for 802.1X users who fail RADIUS authentication due to server unreachability.
- Supported when 802.1X authentication is performed only through RADIUS or ISE servers.
- Can be enabled using the **dot1x authentication event server dead action authorize** command and disabled with the **no** command.

Beginning with Cisco NX-OS Release 10.5(2)F, the 802.1X feature and critical authentication feature are supported on multi-domain ports.

About DACL

Dynamic ACL (DACL) is a single access control list that specifies what users and groups can access, and is used to restrict access to dot1x MAB clients.

- The DACL policy is pushed from the Cisco ISE server to blacklist a MAC address.
- It applies ACLs on the blacklisted MAC, enabling limited access to the MAB client.
- A single DACL supports all blacklisted MAB clients.

Preconfiguration of DACL

In Cisco NX-OS Release 9.3(5), the DACL is preconfigured on the Cisco Nexus switches.

Prerequisites for 802.1x

Use this reference to review prerequisites for 802.1x.

General 802.1X prerequisites

- 802.1X requires Cisco Nexus Release 7.0(3)I7(1) software.
- One or more RADIUS servers must be accessible in the network.
- 802.1X supplicants must be attached to the ports unless MAC address authentication bypass is enabled.
- System-message logging levels for 802.1X must meet or exceed Cisco DCNM requirements; exceptions apply for Cisco Nexus 7000 Series switches running Cisco NX-OS Release 4.0.

EAP-TLS prerequisites

- For EAP-TLS profile, PKI infrastructure is required for certificate management, including RSA key-pair generation, trustpoint creation, and CA authentication.
- 802.1X with EAP-TLS requires a remote EAP server such as ISE; local authentication server is not supported.
- All participating devices, the CA server, and Cisco Identity Services Engine (ISE) must be synchronized using Network Time Protocol (NTP) to ensure certificate validation.
- AAA server reachability is required for mutual authentication between switches, and both switches must have proper AAA configurations and connectivity.

802.1X prerequisite example

For example, to enable 802.1X with EAP-TLS, ensure that the switch is running Cisco Nexus Release 7.0(3)I7(1), a RADIUS server is reachable, PKI infrastructure is in place for certificate management, and all devices are synchronized using NTP.

802.1X guidelines and limitations

Use this reference to review the 802.1X guidelines and limitations.

802.1X guideline overview

- 802.1X port-based authentication is subject to specific guidelines and limitations depending on the Cisco NX-OS release and switch platform.
- Support and restrictions apply to features such as multi-authentication, dynamic VLAN assignment, MAC authentication bypass, and port types (physical, FEX, vPC, port-channels, etc.).
- Some features and configurations are mutually exclusive or unsupported, including certain protocol enhancements, interface types, and feature combinations.

Port-based authentication upgrade and interface guidelines

802.1X port-based authentication on Cisco Nexus Series switches is governed by a set of guidelines and limitations that vary by NX-OS release and hardware platform. The following points summarize the key considerations and restrictions.

- When upgrading to Cisco NX-OS Release 9.2(1), disable 802.1X using **no feature dot1x** and re-enable it with **feature dot1x** for multi-authentication to function.
- From Release 9.2(1), multi-authentication mode is enabled on 802.1X ports. Dynamic VLAN assignment is successful for the first authenticated host; subsequent hosts must have no VLAN assignment or match the first host's VLAN.
- Release 9.2(3) adds support for 802.1X authentication on FEX-ST and host interface (HIF) ports, including both straight-through and dual-homed FEX.
- 802.1X is not supported on transit topology setups, vPC ports, PVLAN ports, L3 (routed) ports, port security, ports enabled with CTS and MACsec PSK, or with LACP port-channels. Static port-channels are supported.
- Disable 802.1X on vPC ports and all unsupported features.
- 802.1X authentication is supported only on physical ports; not on port channels or subinterfaces.
- 802.1X is supported on member ports of a port channel, but not on the port channel itself. Only multi-host mode is supported on port channel members.
- Member ports with and without 802.1X configuration can coexist in a port channel, but identical 802.1X configuration is required for channeling to operate.
- Supplicants are authenticated before any other Layer 2 or Layer 3 features are enabled on an Ethernet interface when 802.1X is enabled.

Interface and feature compatibility limitations

- STP BPDUs are permitted only after successful authentication on 802.1X enabled ports. Enable 802.1X only on STP edge ports to avoid STP disputes.
- 802.1X authentication is supported only on Ethernet/FEX HIF ports and orphan ports with single access-vlan.
- Dynamic VLAN assignment is supported only on Cisco Nexus 9300-FX/EX/FX2 Platform switches.
- 802.1X does not work with CTS or MACsec PSK features. "mac-learn disable" and 802.1X are mutually exclusive and cannot be configured together.
- 802.1X is mutually exclusive with IP Source Guard and uRPF features. Disable one of these features when upgrading to Release 9.2(3).
- Single host mode is not supported on trunk interfaces or member interfaces in a port channel.
- MAC address authentication bypass is not supported on port channels; only multi-host mode is supported on port-channels.
- In Release 9.2(1), MAC authentication bypass is not supported on the N3K-C3164Q-40GE switch.
- 802.1X is not supported on vPC ports and MCT.

Authentication behavior and protocol limitations

- During a switch reload, 802.1X does not generate RADIUS accounting stops.
- 802.1X protocol enhancements not supported: one-to-many logical VLAN name to ID mapping, web authorization, dynamic domain bridge assignment, IP telephony, and guest VLANs.
- To prevent reauthentication of inactive sessions, set the inactivity timer to an interval shorter than the reauthentication interval.
- Selective enabling/disabling of 802.1X on N9K-M12PQ uplink module ports is not supported for Cisco Nexus 9300 platform switches.
- A security violation occurs when the same MAC is learned on a different VLAN with 802.1X enabled on the interface.
- Configuring mac learn disable with 802.1X enabled on a DME enabled platform does not display error messages.
- In Release 9.2(1), tagged EAPOL frames are processed even if the VLAN is not configured on the interface, and authentication is successful for the client.
- Secure MAC learned on the orphan port is not synced on the vPC peer.
- MAC authentication bypass is supported on Cisco Nexus 9300-EX/FX/FX2 TOR switches from Release 9.2(1).

Platform and release support

- From Release 10.5(2)F, MAC authentication bypass is supported on Cisco Nexus 9300-FX3, GX, GX2, H2R, and H1 Series switches.
- From Release 9.3(5), 802.1X is supported on Cisco Nexus 9300-FX3 platform switches.
- From Release 10.1(2), 802.1X is supported on Cisco Nexus 9300-GX platform switches.
- From Release 10.2(1)F, 802.1X is supported on N9K-C9364D-GX2A and N9K-C9332D-GX2B switches.
- From Release 10.3(2)F, MAC authentication bypass and multi-auth are supported on Cisco Nexus 9508 switches with N9K-X9788TC-FX and N9K-X97160YC-EX line cards.
- Cisco Nexus N9K-C9348GC-FXP switches and C9508 switches with N9K-X9788TC-FX and N9K-X97160YC-EX line cards do not support DVLAN, DACL, FEX-AA, VXLAN and mac-move, CoA, only MAB as authentication method (no EAP), and support is for access port with single access VLAN.
- For Cisco Nexus 9000 PX/TX/PQ EoR or ToR switches, when 802.1X is configured on a vPC domain, traffic traversing the peer-link may get punted to CPU if the source MAC belongs to the vPC peer and traffic needs to be bridged over the same VLAN to an orphan port.
- From Release 10.3(3)F, IPv6 underlay is supported on 802.1X for VXLAN EVPN on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 switches and Cisco Nexus 9500 switches with X97160YC-EX, 9700-EX/FX/GX line cards.
- From Release 10.4(1)F, certain Cisco Nexus switches and line cards support 802.1X port-based authentication using EAP/EAP-TLS for uplink ports where MACSec is required, with specific limitations (see below).

EAP-TLS and MACsec configuration limitations

- From Release 10.4(3)F, EAP-TLS supports TLS v1.3 and v1.2 on Cisco Nexus switches. If the RADIUS server does not support TLS v1.3, TLS v1.2 is used as the minimum supported version.
- EAP-TLS supported TLS version is 1.2.
- Single EAP profile per switch is supported; multiple interfaces can use the same profile.
- No support for MAC Move profiles of supplicants.
- Authenticator profile is enabled for L3 ports, trunk ports, vPC for only MACsec EAP-TLS. 802.1X authenticator functionality for MAB/EAP clients is not supported for L3 or Trunk and vPC ports.
- EAP-TLS is supported only for EAP on MACsec configured interfaces and only on Multi-Host mode.
- DACL/Critical AUTH/FEX-AA and other 802.1X features on 802.1X MACsec enabled interfaces are not supported.
- EAP-TLS is supported only for remote authentication (ISE/RADIUS - ISE 3.0 and above); local authentication is not supported.
- Order for EAP-TLS configuration: configure **macsec eap policy** first, then **dot1x supplicant eap profile TLS** . For removal, remove **dot1x supplicant eap profile TLS** first, then **macsec eap policy** . For **no feature** , remove 802.1X feature first, then MACsec feature.

MACsec session and interface limitations

- Single EAP profile configured across the switch can be applied on different interfaces.
- If **macsec eap policy** is configured on interfaces, regular 802.1X authenticator functions or commands are not supported.
- Peer-to-peer MACsec enabled switches must have the same 802.1X or MACsec configurations. If commands differ, shut/no-shut is required to recover.
- Once a MACsec secure session is created with a trust point and eap profile is added to the interface, removal of trustpoint or 802.1X supplicant command will not delete the MACsec session; only MACsec interface-specific command removal will delete the session.
- MACsec PKI is supported only on directly connected switches (no intermediate switches or hops). MACsec PKI (802.1X EAP-TLS) mode does not support EoR Stateful Switch Over (SSO).
- EAP-TLS is supported only on L2/L3 ports, port-channel member ports, trunk ports, and breakout ports. Unsupported interface types have no command-level restriction.
- The number of MACsec sessions supported depends on the physical interface scale.

Upgrade limitations

For example, when upgrading a Cisco Nexus 9300-FX switch to NX-OS Release 9.2(1), you must disable 802.1X before the upgrade and re-enable it after the upgrade to ensure multi-authentication works. Additionally, dynamic VLAN assignment is only available for the first authenticated host on a port, and 802.1X is not supported on vPC or L3 ports.

Attempting to configure 802.1X on a port channel or enabling MAC authentication bypass on a port channel will not work, as these configurations are not supported by Cisco NX-OS software.

Guidelines and limitations for Per-User DACL support for 802.1x

Use this reference to review the guidelines and limitations for Per-User DACL support for 802.1x.

Follow these guidelines and limitations when configuring Per-User DACL:

Supported platforms

The following switch platforms support per-user DACL for 802.1X:

- Cisco Nexus 9300-EX platform switches
- Cisco Nexus 9300-FX platform switches
- Cisco Nexus 9300-FX2 platform switches
- Beginning with Cisco NX-OS Release 10.2(1), the DACL feature is supported on Cisco Nexus 9300-FX/FX2/EX TOR switches.

Feature limitations and configuration notes

Feature limitations and configuration notes:

- Per-user DACL supports IPv4 TCP, UDP, and ICMP ACL rules, but does not support IPv6 ACL rules.
- Per-user DACLs are limited to a single RADIUS response of less than 4KB and a maximum of 32 ACEs.
- This feature does not support standard ACLs on the switch port.
- Only one DACL per port is supported. The maximum number of DACLs supported across a switch is the same as the number of ports in that switch.
- DACL and dynamic VLAN are not supported together on the same port.
- Dynamically modifying DACL content from ISE is not supported. To update, clear the previously applied DACL from the port using the **clear dot1x interface** command, then apply the new one from ISE. This causes transient traffic disruption for all clients on the port.
- Cisco Nexus 9000 series switches in AA FEX mode do not support per-user DACL.
- Per-user DACL supports only MAB and multi-auth host mode.
- Per-user DACL is supported only on physical ports (regular L2 access ports) and not on trunk, vPC, port-channel and its members, or subinterfaces.
- The maximum limit of the per-user DACL is 4000 ASCII characters.
- MAC-move profiles for the per-user DACL feature are not supported.

Per-User DACL support example

For example, on a Cisco Nexus 9300-FX2 switch running NX-OS Release 10.2(1), a per-user DACL can be applied to a physical access port for a MAB client, supporting up to 32 ACEs in a single RADIUS response.

Guidelines and limitations for critical authentication

Use this reference to review the guidelines and limitations for critical authentication.

- Critical Authentication is supported only for basic MAB clients and is not supported on topologies such as FEX-AA and VxLAN.
- Enabling the **authentication event server dead action authorize** command at all times poses a security risk, as it allows all unauthorized client traffic.
- Beginning with Cisco NX-OS Release 10.1(2), the Critical Authentication feature is supported on Cisco Nexus 9300- EX/ FX/FX2/FX3/GX TOR switches.
- Beginning with Cisco NX-OS Release 10.2(1)F, the Critical Authentication feature is supported on the N9K-C9364D-GX2A and N9K-C9332D-GX2B switches.
- Beginning with Cisco NX-OS Release 10.3(2)F, the Critical Authentication feature is supported on Cisco Nexus 9508 switches with N9K-X9788TC-FX and N9K-X97160YC-EX line cards.

Default settings for 802.1x

Use this reference to review the default settings for 802.1x.

- 802.1X is disabled by default on Cisco NX-OS devices.
- The default authentication method for AAA 802.1X is not configured.
- Per-interface 802.1X protocol is disabled (**force-authorized**), allowing normal traffic without authentication.
- Periodic reauthentication is disabled by default.
- The default number of seconds between reauthentication attempts is 3600 seconds.
- The quiet timeout period is 60 seconds, which is the time the device remains in the quiet state after a failed authentication exchange.
- The retransmission timeout period is 30 seconds, representing the wait time for a response to an EAP request/identity frame before retransmitting.
- The maximum retransmission number is 2 times, indicating how many times the device will send an EAP-request/identity frame before restarting authentication.
- The default host mode is single host.
- The supplicant timeout period is 30 seconds, which is the wait time for a response from the supplicant before retransmitting the request.
- The authentication server timeout period is 30 seconds, which is the wait time for a reply from the authentication server before retransmitting the response.

Default 802.1x parameters reference

This section provides a detailed reference table listing the default settings for 802.1X parameters in Cisco NX-OS.

Table 49: Default 802.1x parameters

Parameters	Default
802.1X feature	Disabled
AAA 802.1X authentication method	Not configured
Per-interface 802.1X protocol enable state	Disabled (**force-authorized**) **Note** The port transmits and receives normal traffic without 802.1X-based authentication of the supplicant.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet timeout period	60 seconds (number of seconds that the Cisco NX-OS device remains in the quiet state following a failed authentication exchange with the supplicant)
Retransmission timeout period	30 seconds (number of seconds that the Cisco NX-OS device should wait for a response to an EAP request/identity frame from the supplicant before retransmitting the request)
Maximum retransmission number	2 times (number of times that the Cisco NX-OS device will send an EAP-request/identity frame before restarting the authentication process)
Host mode	Single host
Supplicant timeout period	30 seconds (when relaying a request from the authentication server to the supplicant, the amount of time that the Cisco NX-OS device waits for a response before retransmitting the request to the supplicant)
Authentication server timeout period	30 seconds (when relaying a response from the supplicant to the authentication server, the amount of time that the Cisco NX-OS device waits for a reply before retransmitting the response to the server)

Configuring 802.1x

802.1X configuration refers to the process of enabling and setting up the 802.1X authentication feature on Cisco NX-OS and DCNM devices.

- 802.1X provides port-based network access control.
- Configuration steps may differ between Cisco NX-OS and Cisco IOS platforms.
- Familiarity with the specific CLI commands for NX-OS is important for successful configuration.

Key information for configuring 802.1x

This section describes how to configure the 802.1X feature.

Configure the 802.1x feature

This section describes the process for configuring 802.1X.

Procedure

Step 1 Enable the 802.1X feature.
Step 2 Configure the connection to the remote RADIUS server.
Step 3 Enable 802.1X feature on the Ethernet interfaces.

Enable the 802.1x feature

You must enable the 802.1X feature on the Cisco NX-OS device before authenticating any supplicant devices.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminalswitch(config)#
```

Enters global configuration mode.

Step 2 **feature dot1x**

Example:

```
switch(config)#feature dot1x
```

Enables the 802.1X feature. The default is disabled.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show dot1x**

Example:

```
switch#show dot1x
```

Displays the 802.1X feature status.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure AAA authentication methods for 802.1x

You can use remote RADIUS servers for 802.1X authentication. You must configure RADIUS servers and RADIUS server groups and specify the default AAA authentication method before the Cisco NX-OS device can perform 802.1X authentication.

Before you begin

Obtain the names or addresses for the remote RADIUS server groups.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **aaa authentication dot1x default group** *group-list*

Example:

```
switch(config)#aaa authentication dot1x default group rad2
```

Specifies the RADIUS server groups to use for 802.1X authentication.

The *group-list* argument consists of a space-delimited list of group names. The group names are the following:

- **radius** —Uses the global pool of RADIUS servers for authentication.
- *named-group* —Uses the global pool of RADIUS servers for authentication.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show radius-server**

Example:

```
switch#show radius-server
```

Displays the RADIUS server configuration.

Step 5 (Optional) **show radius-server group** [*group-name*]

Example:

```
switch#show radius-server group rad2
```

Displays the RADIUS server group configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Control 802.1x authentication on an interface

You can control the 802.1X authentication performed on an interface. An interface can have the following 802.1X authentication states:

- **Auto** : Enables 802.1X authentication on the interface.
- **Force-authorized** : Disables 802.1X authentication on the interface and allows all traffic on the interface without authentication. This state is the default.
- **Force-unauthorized** : Disallows all traffic on the interface.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Selects the interface to configure and enters interface configuration mode.

Step 3 **dot1x port-control** {**auto** | **force-authorized** | **forced-unauthorized**}

Example:

```
switch(config-if)#dot1x port-control auto
```

Changes the 802.1X authentication state on the interface. The default is force-authorized.

Step 4 **exit**

Example:

```
switch(config-if)#exit
switch(config)#exit
switch#
```

Exits configuration mode.

Step 5 (Optional) **show dot1x all**

Example:

```
switch#show dot1x all
```

Displays all 802.1X feature status and configuration information.

Step 6 (Optional) **show dot1x interface ethernet** *slot* / *port*

Example:

```
switch#show dot1x interface ethernet 2/1
```

Displays 802.1X feature status and configuration information for an interface.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure the EAP-TLS profile for 802.1x authentication

Beginning with Cisco NX-OS Release 10.4(1)F, you can use EAP-TLS profile for 802.1X authentication.

Before you begin

- Enable the 802.1X feature on the Cisco NX-OS device.
- On the interface, configure the MACsec EAP policy and then attach the **dot1x supplicant eap profile** *eap profile name* . For configuring MACsec EAP policy, see Configure MACsec EAP, on page 662 section.

Procedure

Step 1 **configure terminal**

Enters global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 **[no] eap profile TLS**

Example:

```
switch(config)#eap profile TLS
switch(config-eap-profile)#
```

Configures the 802.1X EAP profile mode.

The **no** form of the command is used to remove the eap profile.

Step 3 **pki-trustpoint** *trustpoint name*

Example:

```
switch(config-eap-profile)#pki-trustpoint tp1
switch(config-eap-profile)#
```

Specifies the trustpoint to be used.

Step 4 **method** *type*

Enters global configuration mode.

Example:

```
switch(config-eap-profile)#method TLS
switch(config-eap-profile)#
```

Specifies the EAP method to be used.

Step 5 **interface ethernet** *slot / port*

Example:

```
switch(config-eap-profile)#interface ethernet 1/30
switch(config-if)#
```

Selects the interface to configure and enters interface configuration mode.

Step 6 **[no] dot1x supplicant eap profile** *eap profile name*

Enters global configuration mode.

Example:

```
switch(config-if)#dot1x supplicant eap profile TLS
```

Configures the 802.1X supplicant to the EAP profile.

Create or remove an authenticator PAE on an interface

Use this procedure to create or remove an authenticator PAE instance on an interface.

You can create or remove the 802.1X authenticator port access entity (PAE) instance on an interface.

Note By default, the Cisco NX-OS software creates the authenticator PAE instance on the interface when you enable 802.1X on an interface.

Before you begin

Enable the 802.1X feature.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 (Optional) **show dot1x interface ethernet** *slot* / *port*

Example:

```
switch#show do1x interface ethernet 2/1
```

Displays the 802.1X configuration on the interface.

Step 3 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Selects the interface to configure and enters interface configuration mode.

Step 4 [**no**] **dot1x pae authenticator**

Example:

```
switch(config-if)#dot1x pae authenticator
```

Creates an authenticator PAE instance on the interface. Use the **no** form to remove the PAE instance from the interface.

Note
If an authenticator PAE already exists on the interface the **dot1x pae authenticator** command does not change the configuration on the interface.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable critical authentication

Before you begin

- Enable monitoring of RADIUS.
- Ensure that all servers in the group are RADIUS servers.

Procedure

Step 1 **radius-server test idle-time** *minutes*

Example:

```
switch#configure terminal
switch(config)#radius-server test idle-time 1
```

Specifies parameters for global server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.

Note
For periodic RADIUS server monitoring, the idle timer value must be greater than 0. If there are multiple servers in the group, set the idle timer to 1 for each server.

Step 2 **radius-server deadtime** *minutes*

Example:

```
switch(config)#radius-server deadtime 1
```

Specifies the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 0 to 1440 minutes.

Note
Set the dead time to a value greater than 0 to enable monitoring.

Step 3 **radius-server host** *ipv4-address* **key** [**0** | **6** | **7**] *key-value*

Example:

```
switch(config)#radius-server host 10.105.222.183 key 7 "fewhg" authentication accounting
```

Specifies a RADIUS key for all RADIUS servers. You can specify if the key-value is in clear text format (**0**), type-6 encrypted (**6**), or type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters. By default, no RADIUS key is configured.

Note
If you already configured a shared secret using the **generate type7_encrypted_secret** command, enter it in quotation marks, as shown in the second example. For more information, see Configure the shared secret for RADIUS or TACACS+, on page 53 .

Step 4 **radius-server host** *ipv4-address* **test idle-time** *minutes*

Example:

```
switch(config)#radius-server host 10.105.222.183 test idle-time 1
```

Specifies parameters for individual server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.

Note
For periodic RADIUS server monitoring, set the idle timer to a value greater than 0.

Step 5 **aaa group server radius** *group-name*

Example:

```
switch(config)#aaa group server radius ISE_2.4
 switch(config-radius)#
```

Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.

To delete a RADIUS server group, use the **no** form of this command.

Note
You are not allowed to delete the default system-generated default group (RADIUS).

Step 6 **server** {*ipv4-address* | *ipv6-address* | *hostname*}

Example:

```
switch(config-radius)#server 10.105.222.183
```

Configures the RADIUS server as a member of the RADIUS server group. If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.

Step 7 **use-vrf** *vrf-name*

Example:

```
switch(config-radius)#use-vrf management
```

Specifies the VRF to use to contact the servers in the server group.

Step 8 **source-interface** *interface*

Example:

```
switch(config-radius)#source-interface mgmt 0
switch(config-radius)#exit
```

Configures the global source interface for all RADIUS server groups configured on the device.

Step 9 **authentication event server dead action authorize**

Example:

```
switch(config)#authentication event server dead action authorize
```

Authorizes all the clients when the RADIUS server is unreachable.

Enable periodic reauthentication for an interface

You can enable periodic 802.1X reauthentication on an interface and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication defaults to the global value.

Note During the reauthentication process, the status of an already authenticated supplicant is not disrupted.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Selects the interface to configure and enters interface configuration mode.

Step 3 **dot1x re-authentication**

Example:

```
switch(config-if)#dot1x re-authentication
```

Enables periodic reauthentication of the supplicants connected to the interface. By default, periodic authentication is disabled.

Step 4 (Optional) **dot1x timeout re-authperiod** *seconds*

Example:

```
switch(config-if)#dot1x timeout re-authperiod 3300
```

Sets the number of seconds between reauthentication attempts. The default is 3600 seconds. The range is from 1 to 65535.

Note
This command affects the behavior of the Cisco NX-OS device only if you enable periodic reauthentication on the interface.

Step 5 **exit**

Example:

```
switch(config-if)#exit
switch(config)#
```

Exits configuration mode.

Step 6 (Optional) **show dot1x all**

Example:

```
switch#show dot1x all
```

Displays all 802.1X feature status and configuration information.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Manually reauthenticate supplicants

You can manually reauthenticate the supplicants for the entire Cisco NX-OS device or for an interface.

Note During the reauthentication process, the status of an already authenticated supplicant is not disrupted.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

dot1x re-authenticate [**interface** *slot* / *port*]

Example:

```
switch#dot1x re-authenticate interface 2/1
```

Reauthenticates the supplicants on the Cisco NX-OS device or on an interface.

Change 802.1x authentication timers for an interface

Use this procedure to change the 802.1X authentication timers for an interface on a Cisco NX-OS device.

You can change the following 802.1X authentication timers on the Cisco NX-OS device interfaces:

- **Quiet-period timer** : When the Cisco NX-OS device cannot authenticate the supplicant, the switch remains idle for a set period of time and then tries again. The quiet-period timer value determines the idle period. An authentication failure might occur because the supplicant provided an invalid password. You can provide a faster response time to the user by entering a smaller number than the default. The default is the value of the global quiet period timer. The range is from 1 to 65535 seconds.
- **Rate-limit timer** : The rate-limit period throttles EAPOL-Start packets from supplicants that are sending too many EAPOL-Start packets. The authenticator ignores EAPOL-Start packets from supplicants that

have successfully authenticated for the rate-limit period duration. The default value is 0 seconds and the authenticator processes all EAPOL-Start packets. The range is from 1 to 65535 seconds.

- **Switch-to-authentication-server retransmission timer for Layer 4 packets** : The authentication server notifies the switch each time that it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the Cisco NX-OS device waits a set period of time and then retransmits the packet. The default is 30 seconds. The range is from 1 to 65535 seconds.
- **Switch-to-supplicant retransmission timer for EAP response frames** : The supplicant responds to the EAP-request/identity frame from the Cisco NX-OS device with an EAP-response/identity frame. If the Cisco NX-OS device does not receive this response, it waits a set period of time (known as the retransmission time) and then retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.
- **Switch-to-supplicant retransmission timer for EAP request frames** : The supplicant notifies the Cisco NX-OS device that it received the EAP request frame. If the authenticator does not receive this notification, it waits a set period of time and then retransmits the frame. The default is the value of the global retransmission period timer. The range is from 1 to 65535 seconds.
- **Inactive period timeout** : When the Cisco NX-OS device remains inactive for a set period of time. The timeout inactivity-period value determines the inactive period. The recommended minimum value is 1800 seconds. You must ensure that the value is less than the value of the re-authentication time.

Note You should change the default values only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain supplicants and authentication servers.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **interface ethernet** *slot* / *port*

Example:

```
switch#configure terminal
switch(config)#interface ethernet 2/1
switch(config-if)
```

Selects the interface to configure and enters interface configuration mode.

Step 2 (Optional) **dot1x timeout quiet-period** *seconds*

Example:

```
switch(config-if)#dot1x timeout quiet-period 25
```

Sets the number of seconds that the authenticator waits for a response to an EAP-request/identity frame from the supplicant before retransmitting the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

Step 3 (Optional) **dot1x timeout ratelimit-period** *seconds*

Example:

```
switch(config-if)#dot1x timeout ratelimit-period 10
```

Sets the number of seconds that the authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated. The default value is 0 seconds. The range is from 1 to 65535 seconds.

Step 4 (Optional) **dot1x timeout server-timeout** *seconds*

Example:

```
switch(config-if)#dot1x timeout server-timeout 60
```

Sets the number of seconds that the Cisco NX-OS device waits before retransmitting a packet to the authentication server. The default is 30 seconds. The range is from 1 to 65535 seconds.

Step 5 (Optional) **dot1x timeout supp-timeout** *seconds*

Example:

```
switch(config-if)#dot1x timeout supp-timeout 20
```

Sets the number of seconds that the Cisco NX-OS device waits for the supplicant to respond to an EAP request frame before the Cisco NX-OS device retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.

Step 6 (Optional) **dot1x timeout tx-period** *seconds*

Example:

```
switch(config-if)#dot1x timeout tx-period 40
```

Sets the number of seconds between the retransmission of EAP request frames when the supplicant does not send notification that it received the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

Step 7 (Optional) **dot1x timeout inactivity-period** *seconds*

Example:

```
switch(config-if)#dot1x timeout inactivity-period 1800
switch(config)#exit
```

Sets the number of seconds the switch can remain inactive. The recommended minimum value is1800 seconds.

Step 8 (Optional) **show dot1x all**

Example:

```
switch#show dot1x all
```

Displays the 802.1X configuration.

Step 9 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable MAC authentication bypass

You can enable MAC authentication bypass on an interface that has no supplicant connected.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)
```

Selects the interface to configure and enters interface configuration mode.

Step 3 **dot1x mac-auth-bypass** [**eap**]

Example:

```
switch(config-if)#dot1x mac-auth-bypass
```

Enables MAC authentication bypass. The default is bypass disabled. Use the **eap** keyword to configure the Cisco NX-OS device to use EAP for authorization.

Step 4 **exit**

Example:

```
switch(config-if)#exit
switch(config)#
```

Exits configuration mode.

Step 5 (Optional) **show dot1x all**

Example:

```
switch#show dot1x all
```

Displays all 802.1X feature status and configuration information.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure the default 802.1x authentication method using MAB

Configure the default 802.1X authentication method to use MAC Authentication Bypass (MAB) on Cisco NX-OS devices.

Beginning with Cisco NX-OS Release 9.3(5), all traffic that is received on the 802.1X enabled ports can be authenticated only by MAC authentication bypass (MAB). Prior to Cisco NX-OS Release 9.3(5), all traffic was first authenticated by EAPOL and authentication by MAB occurred only after the EAPOL authentication session timed out.

Before you begin

Enable the MAB feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)
```

Selects the interface and enters interface configuration mode.

Step 3 **dot1x mac-auth-bypass**

Example:

```
switch(config-if)#dot1x mac-auth-bypass
```

Enables MAC authentication bypass. The default is bypass disabled.

Step 4 [**no**] **dot1x authentication order mab**

Example:

```
switch(config-if)#dot1x authentication order mab
```

Enables MAB for the authentication of the data traffic with the radius server. The no form of this command changes the default authentication method to EAPOL.

Step 5 **exit**

Example:

```
switch(config-if)#exit
switch(config)#
```

Exits configuration mode.

Step 6 (Optional) **show dot1x all**

Example:

```
switch#show dot1x all
```

Displays the 802.1X feature status and configuration information.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Create a dynamic access list

Before you begin

Ensure the following:

- Pre-program the ACL name (acl-name) with all the ACEs to allow or block specific traffic class for the 802.1X MAB client. The configured ACL name (acl-name) on the device must match the acl-name received from the ISE Server.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **hardware access-list tcam region ing-dacl** *tcam size*

Example:

```
switch#hardware access-list tcam region ing-dacl 256
switch(config)#
```

Specifies the TCAM size. The range is between 0 to 2147483647.

Note
This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

Step 3 **ip access-list** *blacklist*

Example:

```
switch(config)#ip access-list creative_blacklist
```

Configures the defined blacklist and applies it based on the configured TCAM size.

Step 4 (Optional) **show ip access-list**

Example:

```
switch#show ip access-list creative_blacklist
```

Displays the configured IP access list.

Step 5 (Optional) **show ip access-list dynamic**

Example:

```
switch#show ip access-list dynamic
IP access list creative_blacklist1_new_Ethernet1/1 statistics per-entry
10 permit udp 0000.1b40.ff13 0000.0000.0000 any range bootps bootpc vlan 100 [match=123]
20 permit udp 0000.1b40.ff13 0000.0000.0000 any eq domain vlan 100 [match=456]
30 deny ip 0000.1b40.ff13 0000.0000.0000 any [match=789]
```

Displays the configured IP access list.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure Per-User DACLs

You can configure per-user DACLs in the Cisco ISE server. You can then implement it in your authorization policies for control of how different users and groups of users access the network.

Procedure

Step 1 **hardware access-list tcam region ing-dacl**

Example:

```
switch#hardware access-list tcam region ing-dacl
```

Configures TCAM on the switch to create a new DACL-TCAM region.

Note
This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

Step 2 **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Step 3 **reload**

Example:

```
switch#reload
```

Reloads the Cisco NX-OS device.

What to do next

Configure the DACL for the blocklisted clients on ISE.

Note The ACEs on ISE shouldn't have a deny rule for IP because an implicit deny is internally added for every DACL client.

The blocklist client connects to the 802.1X port and downloads the ACL AV-Pair as part of the radius access-accept message. The received ACL is then applied on the port for the particular client.

For more information about how to configure the DACLs, see the *Configure Permissions for Downloadable ACLs* section in the *Segmentation* chapter of the *Cisco Identity Services Engine Administrator Guide, Release 3.0* .

Enable single host or multiple hosts mode

You can enable single host or multiple hosts mode on an interface.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)
```

Selects the interface to configure and enters interface configuration mode.

Step 3 **dot1x host-mode** {**multi-host** | **single-host**}

Example:

```
switch(config-if)#dot1x host-mode multi-host
```

Configures the host mode. The default is single-host.

Note

Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.

Step 4 **dot1x host-mode multi-auth**

Example:

```
switch(config-if)#dot1x host-mode multi-auth
```

Configures the multiple authentication mode. The port is authorized only on a successful authentication of either EAP or MAB or a combination of both. Failure to authenticate will restrict network access.

authentication either EAP or MAB

Step 5 **exit**

Example:

```
switch(config-if)#exit
switch(config)#
```

Exits configuration mode.

Step 6 (Optional) **show dot1x all**

Example:

```
switch#show dot1x all
```

Displays all 802.1X feature status and configuration information.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Disable 802.1x authentication on the Cisco NX-OS device

You can disable 802.1X authentication on the Cisco NX-OS device. By default, the Cisco NX-OS software enables 802.1X authentication after you enable the 802.1X feature. However, when you disable the 802.1X feature, the configuration is removed from the Cisco NX-OS device. The Cisco NX-OS software allows you to disable 802.1X authentication without losing the 802.1X configuration.

Note When you disable 802.1X authentication, the port mode for all interfaces defaults to force-authorized regardless of the configured port mode. When you reenable 802.1X authentication, the Cisco NX-OS software restores the configured port mode on the interfaces.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no dot1x system-auth-control**

Example:

```
switch(config)#no dot1x system-auth-control
```

Disables 802.1X authentication on the Cisco NX-OS device. The default is enabled.

Note
Use the **dot1x system-auth-control** command to enable 802.1X authentication on the Cisco NX-OS device.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show dot1x**

Example:

```
switch#show dot1x
```

Displays the 802.1X feature status.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Disable the 802.1x feature

Use this procedure to disable the 802.1X feature on a Cisco NX-OS device.

You can disable the 802.1X feature on the Cisco NX-OS device.

When you disable 802.1X, all related configurations are automatically discarded. The Cisco NX-OS software creates an automatic checkpoint that you can use if you reenable 802.1X and want to recover the configuration. For more information, see the *Cisco NX-OS System Management Configuration Guide* for your platform.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no feature dot1x**

Example:

```
switch(config)#no feature dot1x
```

Disables 802.1X.

Caution
Disabling the 802.1X feature removes all 802.1X configuration.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Reset the 802.1x interface configuration to the default values

You can reset the 802.1X configuration for an interface to the default values.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch#configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot* / *port*
	Example:
	`switch(config)#interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x default** **Example:** `switch(config-if)#dot1x default`	Reverts to the 802.1X configuration default values for the interface.

Step 4	**exit**
	Example:
	`switch(config-if)#exit`
	`switch(config)#`
		Exits configuration mode.

Step 5	(Optional) **show dot1x all** **Example:** `switch#show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch#copy running-config startup-config`	Copies the running configuration to the startup configuration.

Set the maximum Authenticator-to-Supplicant frame for an interface

You can set the maximum number of times that the Cisco NX-OS device retransmits authentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Selects the interface to configure and enters interface configuration mode.

Step 3 **dot1x max-req** *count*

Example:

```
switch(config-if)#dot1x max-req 3
```

Changes the maximum authorization request retry count. The default is 2 times and the range is from 1 to 10.

Note
Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.

Step 4 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits interface configuration mode.

Step 5 (Optional) **show dot1x all**

Example:

```
switch#show dot1x all
```

Displays all 802.1X feature status and configuration information.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable RADIUS accounting for 802.1x authentication

You can enable RADIUS accounting for the 802.1X authentication activity.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **dot1x radius-accounting**

Example:

```
switch(config)#dot1x radius-accounting
```

Enables RADIUS accounting for 802.1X. The default is disabled.

Step 3 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits configuration mode.

Step 4 (Optional) **show dot1x**

Example:

```
switch#show dot1x
```

Displays the 802.1X configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure AAA accounting methods for 802.1x

You can enable AAA accounting methods for the 802.1X feature.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Enters global configuration mode.

Step 2 **aaa accounting dot1x default group** *group-list*

Configures AAA accounting for 802.1X. The default is disabled.

The *group-list* argument consists of a space-delimited list of group names. The group names are the following:

- **radius** —For all configured RADIUS servers.
- *named-group* —Any configured RADIUS server group name.

Step 3 **exit**

Exits configuration mode.

Step 4 (Optional) **show aaa accounting**

Displays the AAA accounting configuration.

Step 5 (Optional) **copy running-config startup-config**

Copies the running configuration to the startup configuration.

This example shows how to configure AAA accounting:

```
switch#configure terminal
switch(config)#aaa accounting dot1x default group radius
switch(config)#exit
switch#show aaa accounting
switch#copy running-config startup-config
```

Set the maximum reauthentication retry count on an interface

You can set the maximum number of times that the Cisco NX-OS device retransmits reauthentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Selects the interface to configure and enters interface configuration mode.

Step 3 **dot1x max-reauth-req** *retry-count*

Example:

```
switch(config-if)#dot1x max-reauth-req 3
```

Changes the maximum reauthentication request retry count. The default is 2 times and the range is from 1 to 10.

Step 4 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits interface configuration mode.

Step 5 (Optional) **show dot1x all**

Example:

```
switch#show dot1x all
```

Displays all 802.1X feature status and configuration information.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

802.1X configuration verification commands

Use this reference to verify the 802.1X configuration.

Verifying the 802.1X configuration involves using specific show commands to display feature status and configuration information for 802.1X on Cisco NX-OS devices.

Table 50: 802.1X show commands and purpose

Command	Purpose
show dot1x	Displays the 802.1X feature status.
show dot1x all [**details** | **statistics** | **summary**]	Displays all 802.1X feature status and configuration information.
show dot1x interface ethernet *slot* / *port* [**details** | **statistics** | **summary**]	Displays the 802.1X feature status and configuration information for an Ethernet interface.

Command	Purpose
show running-config dot1x [**all**]	Displays the 802.1X feature configuration in the running configuration.
show startup-config dot1x	Displays the 802.1X feature configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see the *Cisco NX-OS Security Command Reference* for your platform.

Displays EAP-TLS configuration on a port

The following example displays information about the EAP-TLS configuration on the port as both authenticator and supplicant in authorized state:

```
switch#show dot1x int eth 5/6 details
Dot1x Info for Ethernet5/6
-----------------------------------
PAE = AUTHENTICATOR
PortControl = AUTO
HostMode = MULTI HOST
ReAuthentication = Disabled
QuietPeriod = 60
ServerTimeout = 30
SuppTimeout = 30
ReAuthPeriod = 3600 (Locally configured)
ReAuthMax = 2
MaxReq = 2
TxPeriod = 30
RateLimitPeriod = 0
InactivityPeriod = 0
Mac-Auth-Bypass = Disabled

Dot1x Info for Ethernet5/6
-----------------------------------
PAE = SUPPLICANT
StartPeriod = 30
AuthPeriod = 30
HeldPeriod = 60
MaxStart = 3

Dot1x Authenticator Client List
-------------------------------
Supplicant = 00:00:5E:00:53:01
Domain = DATA
Auth SM State = AUTHENTICATED
Auth BEND SM State = IDLE
Port Status = AUTHORIZED
Authentication Method = EAP
Authenticated By = Remote Server
Auth-Vlan = 0
DACL-Applied = False
Dot1x Supplicant Client List

-------------------------------
Authenticator = 00:00:5E:00:53:01
Supp SM State = AUTHENTICATED
Supp Bend SM State = IDLE
Port Status = AUTHORIZED
```

802.1X support for VXLAN EVPN

This section describes how to configure 802.1X for VXLAN EVPN.

Guidelines and limitations for 802.1x support for VXLAN EVPN

This section provides the guidelines and limitations for 802.1X support for VXLAN EVPN.

Platform and interface support

- Beginning with Cisco NX-OS Release 9.3(7), 802.1X support for VXLAN EVPN feature is supported for Cisco Nexus 9300-GX platform switches.
- Port channel interfaces or the member ports of the port channel are not supported.
- vPC ports are not supported.
- Cisco Nexus 9504 and Cisco Nexus 9508 platform switches with -R line cards does not support multi-authentication and multi-authentication with VXLAN.
- 802.1X is not supported with Fabric Peering.

MAC move and VLAN behavior

- The current support of the feature uses regular and dynamic EVPN updates on the BGP-EVPN control plane for 802.1X secure MAC updates. As a result, the move across EVPN cannot be prevented even if the global policy is **dot1x mac-move deny**.
- Ensure that the **dot1x mac-move** policy is configured the same across the fabric. There is no configuration validation across the nodes, hence it could lead to unexpected behavior if the configuration policy is not in sync.
- The local to remote MAC moves behavior for the deny and permit modes is permitted. Therefore, the MAC move is permitted even if the deny mode is enabled.
- Ensure that the 802.1X and the port-security ports use different VLANs. The same VLAN cannot be assigned to both ports.
- 802.1X is not VLAN aware and hence having the same MAC in two different VLANs is not possible. Depending on the mac-move mode that is selected, either the MAC is moved to a new VLAN or it is denied.

Authentication and feature limitations

- You cannot configure static and secure MAC together.
- RADIUS change of Authorization is supported for VXLAN EVPN.
- The recommended re-authentication time interval for a scale setup is the default value, which is 3600 seconds.

Configure 802.1x support for VXLAN EVPN

This procedure configures 802.1X for VXLAN EVPN.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 **feature dot1x**

Example:

```
switch(config)#feature dot1x
```

Enables the 802.1X feature. The default is disabled.

Step 3 **dot1x mac-move {permit | deny}**

Example:

```
switch(config)#dot1x mac-move permit
```

The **deny** parameters denies MAC moves. The **permit** parameter permits MAC moves.

Step 4 (Optional) **show running-config dot1x all**

Example:

```
switch#show running-config dot1x all
!Command: show running-config dot1x all
!No configuration change since last restart
!Time: Thu Sep 20 10:22:58 2018
version 9.2(2) Bios:version 07.64
feature dot1x
dot1x system-auth-control
dot1x mac-move deny
interface Ethernet1/1
dot1x host-mode multi-auth
dot1x pae authenticator
dot1x port-control auto
no dot1x re-authentication
dot1x max-req 1
dot1x max-reauth-req 2
dot1x timeout quiet-period 60
dot1x timeout re-authperiod 3600
dot1x timeout tx-period 1
dot1x timeout server-timeout 30
dot1x timeout ratelimit-period 0
dot1x timeout supp-timeout 30
dot1x timeout inactivity-period 0
dot1x mac-auth-bypass
interface Ethernet1/33
dot1x host-mode multi-auth
dot1x pae authenticator
dot1x port-control auto
no dot1x re-authentication
```

```
dot1x max-req 1
dot1x max-reauth-req 2
dot1x timeout quiet-period 60
dot1x timeout re-authperiod 3600
dot1x timeout tx-period 1
dot1x timeout server-timeout 30
dot1x timeout ratelimit-period 0
dot1x timeout supp-timeout 30
dot1x timeout inactivity-period 0
dot1x mac-auth-bypass
```

Displays the 802.1X configuration.

802.1X support for VXLAN EVPN verification commands

To display the 802.1X support for VXLAN EVPN configuration information, enter one of the following commands.

Table 51: 802.1X VXLAN EVPN verification commands

Command	Purpose
show running-config dot1x all	Displays 802.1X running configuration.
show dot1x all summary	Displays the interface status.
show dot1x	Displays the default settings.
show dot1x all	Displays additional interface detail.

Example of show running-config dot1x all command

```
switch#show running-config dot1x all
!Command: show running-config dot1x all
!No configuration change since last restart
!Time: Thu Sep 20 10:22:58 2018
version 9.2(2) Bios:version 07.64
feature dot1x
dot1x system-auth-control
dot1x mac-move deny
interface Ethernet1/1
dot1x host-mode multi-auth
dot1x pae authenticator
dot1x port-control auto
no dot1x re-authentication
dot1x max-req 1
dot1x max-reauth-req 2
dot1x timeout quiet-period 60
dot1x timeout re-authperiod 3600
dot1x timeout tx-period 1
dot1x timeout server-timeout 30
dot1x timeout ratelimit-period 0
dot1x timeout supp-timeout 30
dot1x timeout inactivity-period 0
dot1x mac-auth-bypass
interface Ethernet1/33
dot1x host-mode multi-auth
```

```
dot1x pae authenticator
dot1x port-control auto
no dot1x re-authentication
dot1x max-req 1
dot1x max-reauth-req 2
dot1x timeout quiet-period 60
dot1x timeout re-authperiod 3600
dot1x timeout tx-period 1
dot1x timeout server-timeout 30
dot1x timeout ratelimit-period 0
dot1x timeout supp-timeout 30
dot1x timeout inactivity-period 0
dot1x mac-auth-bypass
```

Example of the show dot1x all summary command

```
switch#show dot1x all summary
Interface     PAE              Client          Status
--------------------------------------------------------------------
Ethernet1/1    AUTH                none    UNAUTHORIZED
Interface     PAE              Client          Status
--------------------------------------------------------------------
Ethernet1/33    AUTH   00:00:5E:00:53:01      AUTHORIZED
00:00:5E:00:53:02      AUTHORIZED
00:00:5E:00:53:03      AUTHORIZED
00:00:5E:00:53:04      AUTHORIZED
switch#show mac address-table vlan 10
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*   10     0016.5a4c.0004   secure   -         T      F    Eth1/33
*   10     0016.5a4c.0005   secure   -         T      F    Eth1/33
*   10     0016.5a4c.0006   secure   -         T      F    Eth1/33
*   10     0016.5a4c.0007   secure   -         T      F    Eth1/33
switch#show mac address-table vlan 10 (VPC-PEER)
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*   10     0016.5a4c.0004   secure   -         T      F    vPC Peer-Link
*   10     0016.5a4c.0005   secure   -         T      F    vPC Peer-Link
*   10     0016.5a4c.0006   secure   -         T      F    vPC Peer-Link
*   10     0016.5a4c.0007   secure   -         T      F    vPC Peer-Link
switch#show mac address-table vlan 10 (RVTEP)
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C   10     0016.5a4c.0004   dynamic  0         F      F    nve1(192.0.2.1)
C   10     0016.5a4c.0005   dynamic  0         F      F    nve1(192.0.2.1)
C   10     0016.5a4c.0006   dynamic  0         F      F    nve1(192.0.2.1)
C   10     0016.5a4c.0007   dynamic  0         F      F    nve1(192.0.2.1)
```

Example of the show dot1x command

```
switch#show dot1x
 Sysauthcontrol Enabled
Dot1x Protocol Version 2Mac-Move Deny
```

Example of the show dot1x all command

```
switch#show dot1x all
Sysauthcontrol Enabled
Dot1x Protocol Version 2
Mac-Move Deny
Dot1x Info for Ethernet1/1
-----------------------------------
PAE = AUTHENTICATOR
PortControl = AUTO
HostMode = MULTI AUTH
ReAuthentication = Disabled
QuietPeriod = 60
ServerTimeout = 30
SuppTimeout = 30
ReAuthPeriod = 3600 (Locally configured)
ReAuthMax = 2
MaxReq = 1
TxPeriod = 1
RateLimitPeriod = 0
InactivityPeriod = 0
Mac-Auth-Bypass = Enabled
Dot1x Info for Ethernet1/33
-----------------------------------
PAE = AUTHENTICATOR
PortControl = AUTO
HostMode = MULTI AUTH
ReAuthentication = Disabled
QuietPeriod = 60
ServerTimeout = 30
SuppTimeout = 30
ReAuthPeriod = 3600 (Locally configured)
ReAuthMax = 2
MaxReq = 1
TxPeriod = 1
RateLimitPeriod = 0
InactivityPeriod = 0
Mac-Auth-Bypass = Enabled
```

Critical authentication verification

The following example shows how to view if the critical authentication feature is enabled.

The following example shows how to view if the critical authentication feature is enabled.

```
switch# show dot1x
  Sysauthcontrol               Enabled
  Dot1x Protocol Version       2
  Mac-Move Permit
  Server-Dead-Action-Authorize Enabled
```

If the value of the **Server-Dead-Action-Authorize** parameter is **Enabled** , the critical authentication feature is enabled.

Monitor 802.1x statistics

Monitor 802.1X statistics to review authentication activity and troubleshoot network access issues on Cisco NX-OS devices.

You can display the statistics that the Cisco NX-OS device maintains for the 802.1X activity.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

Procedure

show dot1x {**all** | **interface ethernet** *slot* / *port*} **statistics**

Example:

```
switch#show dot1x all statistics
```

Displays the 802.1X statistics.

802.1X port configuration commands

This reference provides the necessary commands to configure 802.1X authentication and illustrates expected system behavior through operational examples.

Configuration Commands

Configure 802.1X for an access port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
  dot1x pae authenticator
  dot1x port-control auto
```

Configure 802.1X for a trunk port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
  dot1x pae authenticator
  dot1x port-control auto
  dot1x host-mode multi-host
```

Note Repeat the **dot1x pae authenticator** and **dot1x port-control auto** commands for all interfaces that require 802.1X authentication.

Operational examples - open mode authentication

Example output from **show dot1x all details** on a port operating in open mode:

```
switch#show dot1x all details
Dot1x Info for Ethernet1/26
-----------------------------------
PAE = AUTHENTICATOR
PortControl = AUTO
HostMode = MULTI-DOMAIN
Mac-Auth-Bypass = Enabled
       AuthenticationOpen = Enabled         <-- Open Mode is Enabled
ViolationRestrict = Enabled

Dot1x Authenticator Client List
-------------------------------
Supplicant = 00:00:5E:00:53:01
Domain = VOICE
Auth SM State = AUTHENTICATED
Port Status = AUTHORIZED
    Authentication Method = MAB
Authenticated By = Open Mode       <-- Authenticated via Open Mode
Voice-Vlan = 111

-------------------------------
Supplicant = 00:00:5E:00:53:02
Domain = DATA
Auth SM State = AUTHENTICATED
Port Status = AUTHORIZED
    Authentication Method = MAB
Authenticated By = Open Mode       <-- Authenticated via Open Mode
Auth-Vlan = 110
```

Example output from **show mac address-table** in open mode:

```
switch#show mac address-table
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan,
(NA)- Not Applicable A â-’ ESI Active Path, S â-’ ESI Standby Path
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*  110     0030.3030.3030   secure   -         T      F    Eth1/26      <-- Data client on
 access VLAN
*  110     c414.3c97.2246   static   -         F      F    Drop         <-- Phone has drop
 entry on access VLAN
*  111     c414.3c97.2246   secure   -         T      F    Eth1/26      <-- Phone learnt
on voice VLAN
G    -     b44c.90a4.f907   static   -         F      F    sup-eth1(R)
```

Example output from **show mac address-table** when a security violation occurs:

```
switch#show mac address-table
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*  110     0030.3030.3030   secure   -         T      F    Eth1/26
*  110     0030.3030.3040   static   -         F      F    Drop         <-- Drop entry on
second data client
*  110     c414.3c97.2246   static   -         F      F    Drop
*  111     c414.3c97.2246   secure   -         T      F    Eth1/26
G    -     b44c.90a4.f907   static   -         F      F    sup-eth1(R)
```

COA Examples

COA reauthentication triggered from ISE for data client:

```
switch#show aaa client radius statistics 10.76.80.245
Dynamic Author Client 10.76.80.245
COA Statistics
    Requests: 1
    Transactions: 1
    Retransmissions: 0
    Active Transactions: 0
    Ack Responses: 1                    ! COA ACK sent to ISE
    Nak Responses: 0
    Invalid Requests: 0
    Errors: 0
```

Session state before COA disconnect:

```
switch#show dot1x all summary
           Interface     PAE              Client          Status
-----------------------------------------------------------------------
        Ethernet1/26    AUTH   00:00:5E:00:53:01      AUTHORIZED
                               00:00:5E:00:53:02      AUTHORIZED
```

Session state after COA disconnect:

```
switch#show dot1x all summary
           Interface     PAE              Client          Status
-----------------------------------------------------------------------
        Ethernet1/26    AUTH   00:00:5E:00:53:01      AUTHORIZED

switch#show aaa client radius statistics 10.76.80.245
COA Statistics
    Requests: 3
    Ack Responses: 3
```

Configuration example for Per-User DACL

Use this reference to review a Per-User DACL configuration example.

Per-User DACL behavior

The per-user DACL is a security feature that allows dynamic application of access control lists to individual clients on a switch port, enabling granular traffic filtering based on client authentication status.

- Filters blocklist traffic by applying DACLs to authenticated clients.
- Identifies blocklist clients when the DACL-Applied parameter is set to true.
- Integrates with ISE to deliver ACLs dynamically to clients.

Viewing Per-User DACL configuration and blocklisted traffic

The following information describes how to verify per-user DACL configuration and view blocklisted traffic on a switch port.

- Use the **show dot1x all summary** and **show dot1x all details** commands to check client authentication and DACL application status.
- Use the **show ip access-list dynamic** command to view blocklisted traffic entries.

Example: Verifying Per-User DACL and blocklisted traffic

The following example shows the per-user DACL configured on one of the ports. When the DACL is applied, the blocklist traffic is filtered out. If the value of the DACL-Applied parameter is true, the client is a blocklist client, which has received an ACL from ISE.

```
switch#show dot1x all summary
Interface       PAE        Client               Status
Ethernet1/1     AUTH       00:00:5E:00:53:01    AUTHORIZED
                           00:00:5E:00:53:02    AUTHORIZED

switch#show dot1x all details
-----------------------------
Supplicant = 00:00:5E:00:53:01
Domain = DATA
Auth SM State = AUTHENTICATED
DACL-Applied = False
-------------------------------

Supplicant = 00:00:5E:00:53:02
Domain = DATA
Auth SM State = AUTHENTICATED
DACL-Applied = True
```

The following example shows how to view the blocklisted traffic.

```
switch#show ip access-list dynamic
IP access list DOT1X_Restricted_base_acl_Ethernet1/1_new statistics per-entry fragments
deny-all
  10 permit udp any 3612.6151.2153 0000.0000.0000 any eq 5555 vlan 100 [match=0]
  20 permit udp any 3612.6151.2153 0000.0000.0000 any eq 6666 vlan 100 [match=0]
  30 deny ip any 3612.6151.2153 0000.0000.0000 any vlan 100 [match=0]
```

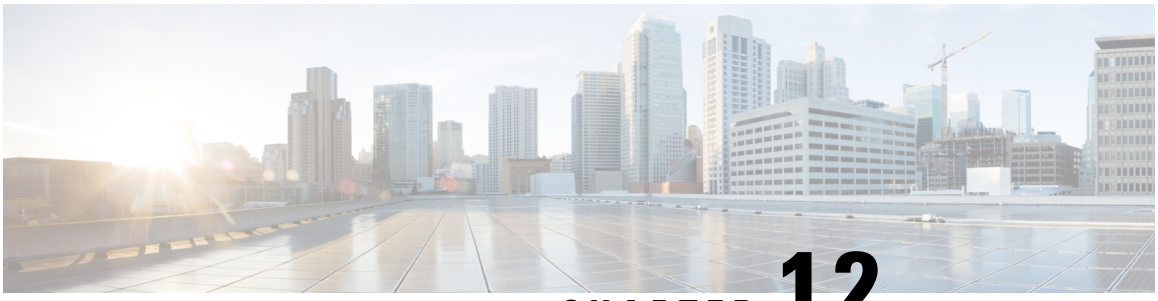
Additional references for 802.1x

This section includes additional information related to implementing 802.1X.

Standards

Table 52: Standards

Standards	Title
IEEE Std 802.1X- 2004 (Revision of IEEE Std 802.1X-2001)	*802.1X IEEE Standard for Local and Metropolitan Area Networks Port-Based Network Access Control*
RFC 2284	*PPP Extensible Authentication Protocol (EAP)*
RFC 3580	*IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*



CHAPTER 12

Configure IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on Cisco NX-OS devices.

Unless otherwise specified, the term IP ACL refers to IPv4 and IPv6 ACLs.

- About ACLs, on page 307
- Prerequisites for IP ACLs, on page 326
- Guidelines and limitations for IP ACLs, on page 326
- Default settings for IP ACLs, on page 337
- Configuring IP ACLs, on page 337
- IP ACL configuration verification commands, on page 373
- Monitoring and clearing IP ACL statistics, on page 375
- Configuration examples for IP ACLs, on page 376
- About system ACLs, on page 377
- Configuring object groups, on page 381
- Object-group configuration verification commands, on page 386
- Configuring Time-Ranges, on page 386
- Time-range configuration verification commands, on page 391
- Additional references for IP ACLs, on page 391

About ACLs

An ACL is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the device determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first matching rule determines whether the packet is permitted or denied. If there is no match, the device applies the applicable implicit rule. The device continues processing packets that are permitted and drops packets that are denied.

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you could use ACLs to disallow HTTP traffic from a high-security network to the Internet. You could also use ACLs to allow HTTP traffic but only to specific sites, using the IP address of the site to identify it in an IP ACL.

ACL types and applications

The device supports the following types of ACLs for security traffic filtering:

- IPv4 ACLs

  The device applies IPv4 ACLs only to IPv4 traffic.

- IPv6 ACLs

  The device applies IPv6 ACLs only to IPv6 traffic.

- MAC ACLs

  The device applies MAC ACLs only to non-IP traffic.

IP and MAC ACLs have the following types of applications:

- Port ACL

  Filters Layer 2 traffic

- MAC ACL with UDF-based match

  Filters MAC ACLs with UDF-based match

- Router ACL

  Filters Layer 3 traffic

- VLAN ACL

  Filters VLAN traffic

- VTY ACL

  Filters virtual teletype (VTY) traffic

Table 53: Security ACL applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	• Layer 2 interfaces • Layer 2 Ethernet port-channel interfaces When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	• IPv4 ACLs • IPv4 ACLs with UDF-based match • IPv6 ACLs • IPv6 ACLs with UDF-based match • MAC ACLs • MAC ACLs with UDF-based match

Application	Supported Interfaces	Types of ACLs Supported
Router ACL	• VLAN interfaces • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Management interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface.	• IPv4 ACLs • IPv6 ACLs **Note** MAC ACLs are supported on Layer 3 interfaces only if you enable MAC packet classification.
VLAN ACL	• VLANs	• IPv4 ACLs • IPv6 ACLs • MAC ACLs
VTY ACL	• VTYs	• IPv4 ACLs • IPv6 ACLs

Related Topics

VLAN ACLs, on page 407
MAC ACLs, on page 393

Order of ACL application

When the device processes a packet, it determines the forwarding path of the packet. The path determines which ACLs that the device applies to the traffic. The device applies the ACLs in the following order:

1. Port ACL
2. Ingress VACL
3. Ingress router ACL
4. Ingress VTY ACL
5. Egress VTY ACL
6. Egress router ACL
7. Egress VACL

If the packet is bridged within the ingress VLAN, the device does not apply router ACLs.

Figure 8: Order of ACL application

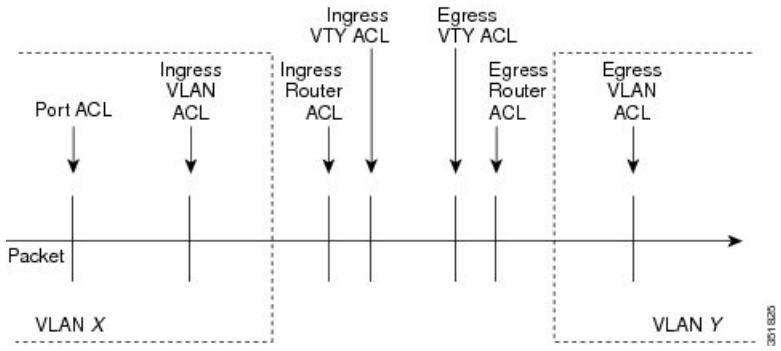
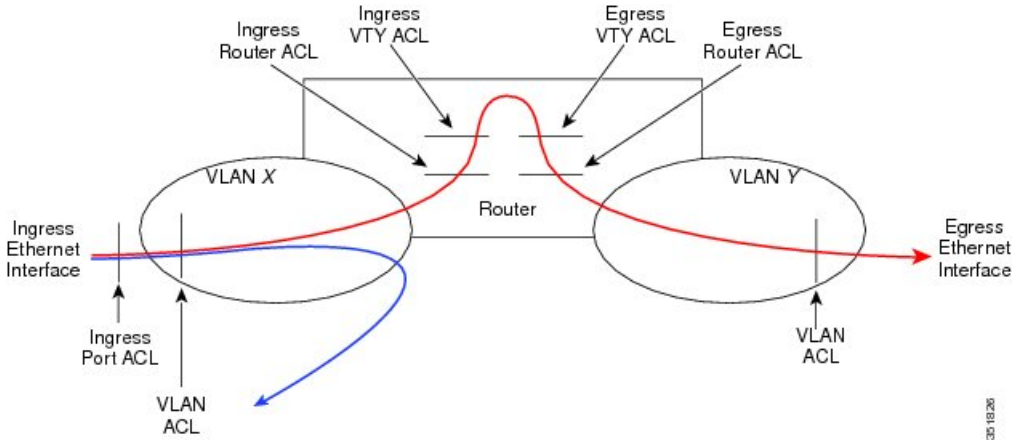


Figure 9: Acls and packet flow



About rules

Rules are what you create, modify, and remove when you configure how an ACL filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to the applicable I/O module. Depending upon how you configure the ACL, there may be more ACL entries than rules, especially if you implement policy-based ACLs by using object groups when you configure rules.

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The device allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

This section describes some of the options that you can use when you configure a rule.

Protocols for IP ACLs and MAC ACLs

IPv4, IPv6, and MAC ACLs allow you to identify traffic by protocol. For your convenience, you can specify some protocols by name. For example, in an IPv4 or IPv6 ACL, you can specify ICMP by name.

You can specify any protocol by number. In MAC ACLs, you can specify protocols by the EtherType number of the protocol, which is a hexadecimal number. For example, you can use 0x0800 to specify IP traffic in a MAC ACL rule.

In IPv4 and IPv6 ACLs, you can specify protocols by the integer that represents the Internet protocol number.

Source and destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host. How you specify the source and destination depends on whether you are configuring IPv4 ACLs, IPv6 ACLs, or MAC ACLs.

Implicit rules for IP and MAC ACLs

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the device applies them to traffic when no other rules in an ACL match. When you configure the device to maintain per-rule statistics for an ACL, the device does not maintain statistics for implicit rules.

IPv4 implicit rule

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the device denies unmatched IP traffic.

IPv6 implicit rule

All IPv6 ACLs include the following implicit rule:

```
deny ipv6 any any
```

This implicit rule ensures that the device denies unmatched IPv6 traffic.

IPv6 neighbor discovery traffic



Note

- IPv6 Neighbor Discovery packets (Router Solicitation, and Router Advertisement) will not be permitted due to the implicit **deny ipv6 any any** rule of an IPv6 ACL.
- You must add the following rules explicitly to allow IPv6 Neighbor Discovery packets in the Cisco Nexus 93180YC-EX, Nexus 93180YC-FX, Nexus 93240YC-FX2, Nexus 93360YC-FX2, Nexus 9336C-FX2, Nexus 9336C-FX2-E, Nexus 93180YC-FX3, N9K-C9316D-GX, N9K-C93600CD-GX, Nexus 9364C-GX, N9K-C9332D-GX2B, Nexus 9364C and Nexus 9332C platform switches:
  - **permit icmp any any router-advertisement**
  - **permit icmp any any router-solicitation**
- Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages do not match under the implicit rule. The following commands are required to match the NS or NA IPv6 traffic.
  - **permit/deny icmp any any nd-na**
  - **permit/deny icmp any any nd-ns**

MAC implicit rule

All MAC ACLs include the following implicit rule:

```
deny any any protocol
```

This implicit rule ensures that the device denies the unmatched traffic, regardless of the protocol specified in the Layer 2 header of the traffic.

Additional filtering options

You can identify traffic by using additional options. These options differ by ACL type. The following list includes most but not all additional filtering options:

- IPv4 ACLs support the following additional filtering options:
  - Layer 4 protocol
  - TCP and UDP ports
  - ICMP types and codes
  - IGMP types
  - Precedence level
  - Differentiated Services Code Point (DSCP) value
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
  - Established TCP connections
  - Packet length
- IPv6 ACLs support the following additional filtering options:
  - Layer 4 protocol
  - Encapsulating Security Payload
  - Payload Compression Protocol
  - Stream Control Transmission Protocol (SCTP) , TCP, and UDP ports
  - ICMP types and codes
  - DSCP value
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
  - Established TCP connections
  - Packet length
- MAC ACLs support the following additional filtering options:
  - Layer 3 protocol (Ethertype)
  - VLAN ID
  - Class of Service (CoS)

- Beginning Cisco NX-OS Release 9.2(4), IPv4 ACLs and IPv6 in Cisco Nexus 9500 platform switches with N9K-X96136YC-R, N9K-X9636C-R, and N9K-X9636C-RX line cards and N9K-C9504-FM-R fabric module support the following additional filtering options:
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
  - Established TCP connections



Note

- TCP flag options are correctly processed by Netstack rather than the kernel (KStack), due to the kernel's lack of support for TCP flags. Additionally, the following syslog message is generated:

  ```
  <HOSTNAME> %NPACL-2-IPT_WARNING: npacl [<#>] WARNING: Mgmt ACL: <ACL>
  Seq:<Seq#> has ACL option: tcp-flags that is not supported in kernel
  stack. Hence that option is not added in its filter rule.
  ```

- The **tcp-flags-mask** option is not supported.

Sequence numbers

The device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. Sequence numbers simplify the following ACL tasks:

- **Adding new rules between existing rules**

  By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.

- **Removing a rule**

  Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

  ```
  switch(config-acl)#no permit tcp 10.0.0.0/8 any
  ```

  However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

  ```
  switch(config-acl)#no 101
  ```

- **Moving a rule**

  With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the device adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the device assigns the sequence number 235 to the new rule.

In addition, Cisco NX-OS allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical operators and logical operation units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers. Cisco NX-OS supports logical operators in only the ingress direction.

The device stores operator-operand couples in registers called logical operator units (LOUs). The LOU usage for each type of operator is as follows:

- eq

  Is never stored in an LOU

- gt

  Uses 1 LOU

- lt

  Uses 1 LOU

- neq

  Uses 1 LOU

- range

  Uses 1 LOU

Note For range operators, LOU threshold configuration is used to control how the port range is expanded when configuring an ACL entry. If you want to use the LOU operator when the number of the ACL rules exceed the configured threshold value, run the following command: **hardware access-list lou resource threshold** *<x>*, wherein *<x>* denotes the number of ACL rules to be used before the LOU threshold is reached. The range value for *<x>* is 1 to 50, and the default value for LOU threshold is 5.

ACL logging

The ACL logging feature monitors ACL flows and logs statistics.

A flow is defined by the source interface, protocol, source IP address, source port, destination IP address, and destination port values. The statistics maintained for a flow include the number of forwarded packets (for each flow that matches the permit conditions of the ACL entry) and dropped packets (for each flow that matches the deny conditions of the ACL entry).

Beginning with Cisco NX-OS Release 10.4(3)F, ACL logging for Security Group ACL (SGACL) is provided on Cisco Nexus 9300-FX3/GX/GX2/H2R platform switches.

For SGACL, a flow is defined by the security group tag (SGT), destination group tag (DGT), source MAC (SMAC), destination MAC (DMAC), SGACL permit/deny information, physical interface on which packet arrived, and hit counts for that particular SGACL flow apart from the basic 5 tuples. To enable the SGACL logging, see Configure IPv4 ACL logging, on page 367.

Time ranges

You can use time ranges to control when an ACL rule is in effect. For example, if the device determines that a particular ACL applies to traffic arriving on an interface, and a rule in the ACL uses a time range that is not in effect, the device does not compare the traffic to that rule. The device evaluates time ranges based on its clock.

When you apply an ACL that uses time ranges, the device updates the affected I/O module whenever a time range referenced in the ACL starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.

IPv4, IPv6, and MAC ACLs support time ranges. When the device applies an ACL to traffic, the rules in effect are as follows:

- All rules without a time range specified
- Rules with a time range that includes the second when the device applies the ACL to traffic

The device supports named, reusable time ranges, which allows you to configure a time range once and specify it by name when you configure many ACL rules. Time range names have a maximum length of 64 alphanumeric characters.

A time range contains one or more rules. The two types of rules are as follows:

- **Absolute**

  A rule with a specific start date and time, specific end date and time, both, or neither. The following items describe how the presence or absence of a start or end date and time affect whether an absolute time range rule is active:

  - Start and end date and time both specified—The time range rule is active when the current time is later than the start date and time and earlier than the end date and time.
  - Start date and time specified with no end date and time—The time range rule is active when the current time is later than the start date and time.
  - No start date and time with end date and time specified—The time range rule is active when the current time is earlier than the end date and time.
  - No start or end date and time specified—The time range rule is always active.

  For example, you could prepare your network to allow access to a new subnet by specifying a time range that allows access beginning at midnight of the day that you plan to place the subnet online. You can use that time range in ACL rules that apply to the subnet. After the start time and date have passed, the device automatically begins applying the rules that use this time range when it applies the ACLs that contain the rules.

- **Periodic**

  A rule that is active one or more times per week. For example, you could use a periodic time range to allow access to a lab subnet only during work hours on weekdays. The device automatically applies ACL rules that use this time range only when the range is active and when it applies the ACLs that contain the rules.

Note The order of rules in a time range does not affect how a device evaluates whether a time range is active. Cisco NX-OS includes sequence numbers in time ranges to make editing the time range easier.

Time ranges also allow you to include remarks, which you can use to insert comments into a time range. Remarks have a maximum length of 100 alphanumeric characters.

The device determines whether a time range is active as follows:

- The time range contains one or more absolute rules—The time range is active if the current time is within one or more absolute rules.
- The time range contains one or more periodic rules—The time range is active if the current time is within one or more periodic rules.
- The time range contains both absolute and periodic rules—The time range is active if the current time is within one or more absolute rules and within one or more periodic rules.

  When a time range contains both absolute and periodic rules, the periodic rules can only be active when at least one absolute rule is active.

Policy-based ACLs

The device supports policy-based ACLs (PBACLs), which allow you to apply access control policies across object groups. An object group is a group of IP addresses or a group of TCP or UDP ports. When you create a rule, you specify the object groups rather than specifying IP addresses or ports.

Using object groups when you configure IPv4 or IPv6 ACLs can help reduce the complexity of updating ACLs when you need to add or remove addresses or ports from the source or destination of rules. For example, if three rules reference the same IP address group object, you can add an IP address to the object instead of changing all three rules.

PBACLs do not reduce the resources required by an ACL when you apply it to an interface. When you apply a PBACL or update a PBACL that is already applied, the device expands each rule that refers to object groups into one ACL entry per object within the group. If a rule specifies the source and destination both with object groups, the number of ACL entries created on the I/O module when you apply the PBACL is equal to the number of objects in the source group multiplied by the number of objects in the destination group.

The following object group types apply to port, router, policy-based routing (PBR), and VLAN ACLs:

- IPv4 Address Object Groups

  Can be used with IPv4 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

- IPv6 Address Object Groups

  Can be used with IPv6 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

- Protocol Port Object Groups

Can be used with IPv4 and IPv6 TCP and UDP rules to specify source or destination ports. When you use the **permit** or **deny** command to configure a rule, the **portgroup** keyword allows you to specify an object group for the source or destination.

Note Policy-based routing (PBR) ACLs do not support deny access control entries (ACEs) or **deny** commands to configure a rule.

Kernel Stack ACL

The Kernel Stack ACL is a common CLI infrastructure to configure ACLs for management of inband and outband components.

The Kernel Stack ACL uses NX-OS ACL CLI to secure management applications on management and front panel ports. Configuring a single ACL must be able to secure all management applications on NX-OS.

Kernel Stack ACL is the component that fixes the manual intervention of the user and automatically programs iptable entries when the ACL is applied to mgmt0 interface.

The following is an example for configuring Kernel Stack ACL:

```
swtich# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ip access-list kacl1
switch(config-acl)# statistics per-entry
switch(config-acl)# 10 deny tcp any any eq 443
switch(config-acl)# 20 permit ip any any
switch(config-acl)# end
switch#
```

```
switch(config-if)# interface mgmt0
switch(config-if)#   ip access-group acl1 in
switch(config-if)#   ipv6 traffic-filter acl6 in
switch(config-if)#
```

```
switch# sh ip access-lists kacl1
IP access list kacl1
statistics per-entry
10 deny tcp any any eq 443 [match=136]
20 permit ip any any [match=44952]
switch(config)#
```

The following is the Kernel Stack filtering for iptables entries based on the configuration:

```
bash-4.4# ip netns exec management iptables -L -n -v --line-numbers
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination
1 9 576 DROP tcp -- * * 0.0.0.0/0 0.0.0.0/0 tcp dpt:443
2 0 0 ACCEPT all -- * * 0.0.0.0/0 0.0.0.0/0
3 0 0 DROP all -- * * 0.0.0.0/0 0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination
bash-4.4#
```

The following are the limitations for the Kernel Stack ACL support:

- This feature is supported only on mgmt0 interface and not on other inband interfaces.
- Five tuples (protocol, source-ip, destination-ip, source-port, and destination-port) of the ACL entry are programmed in the iptables. Rest of the options provided in the ACL entry are not programmed in the iptables and throws a warning syslog in such instances.

  For example, "WARNING: Some ACL options are not supported in kstack. Only partial rule will be installed".
- If the device user has host bash access, then the user can manually update the iptables. This update could potentially corrupt the iptable rules for which they are programmed.
- The verified maximum number of ACEs is 100 for IPv4 traffic and an additional 100 for IPv6 traffic. Throughput may be impacted if more than this scale is applied.

Statistics and ACLs

The device can maintain global statistics for each rule that you configure in IPv4, IPv6, and MAC ACLs. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

Note The device does not support interface-level ACL statistics.

For each ACL that you configure, you can specify whether the device maintains statistics for that ACL, which allows you to turn ACL statistics on or off as needed to monitor traffic filtered by an ACL or to help troubleshoot the configuration of an ACL.

The device does not maintain statistics for implicit rules in an ACL. For example, the device does not maintain a count of packets that match the implicit **deny ip any any** rule at the end of all IPv4 ACLs. If you want to maintain statistics for implicit rules, you must explicitly configure the ACL with rules that are identical to the implicit rules.

Related Topics

Monitoring and clearing IP ACL statistics, on page 375
Implicit rules for IP and MAC ACLs, on page 311

Atomic ACL updates

By default, Cisco NX-OS uses atomic programming when it updates an access control list (ACL) in hardware. An atomic update maintains the existing ACL while programming the updated ACL. Therefore, the I/O module must have sufficient TCAM resources for both versions of the ACL.

The required TCAM capacity depends on the number of hardware entries that the ACL uses. Some access control entries can expand into multiple hardware entries.

When atomic updates are enabled and sufficient TCAM resources are unavailable, the device generates an error, and the ACL update fails.

When atomic updates are disabled, Cisco NX-OS first attempts the update atomically. If sufficient resources are unavailable, the device performs a non-atomic update. During a non-atomic update, the device removes the existing hardware entries before programming the updated entries. This process can briefly affect traffic.

To disable atomic updates, use this command:

```
switch#configure terminal
switch(config)#no hardware access-list update atomic
```

By default, the device drops affected traffic during a non-atomic ACL update. To permit affected traffic during the update, use this command:

```
switch(config)#hardware access-list update default-result permit
```

Note The **hardware access-list update default-result permit** command applies to supported router ACL and port ACL updates. The command does not control the programming result for policy-based routing redirects, ePBR redirects, or other policy types.

To restore atomic ACL updates, use these commands:

```
switch(config)#no hardware access-list update default-result permit
switch(config)#hardware access-list update atomic
```

Session manager support for IP ACLs

Session Manager supports the configuration of IP and MAC ACLs. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.

ACL TCAM regions

Use this reference to review ACL TCAM regions.

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

On Cisco Nexus 9300 and 9500 Series switches, the egress TCAM size is 1K, divided into four 256 entries. On other Cisco Nexus 9300 and 9500 Series switches, the ingress TCAM size is 4K, divided into eight 256 slices and four 512 slices. A slice is the unit of allocation. A slice can be allocated to one region only. For example, a 512-size slice cannot be used to configure two features of size 256 each. Similarly, a 256-size slice cannot be used to configure two features of size 128 each. The IPv4 TCAM regions are single wide. The IPv6, QoS, MAC, CoPP, and system TCAM regions are double wide and consume double the physical TCAM entries. For example, a logical region size of 256 entries actually consumes 512 physical TCAM entries.

You can create IPv6, port ACLs, VLAN ACLs, and router ACLs, and you can match IPv6 and MAC addresses for QoS. However, Cisco NX-OS cannot support all of them simultaneously. You must remove or reduce the size of the existing TCAM regions (TCAM carving) to enable the IPv6, MAC, or other desired TCAM regions. For every TCAM region configuration command, the system evaluates if the new change can be fit in the TCAM. If not, it reports an error, and the command is rejected. You must remove or reduce the size of existing TCAM regions to make room for new requirements.

On Cisco Nexus 9200 Series switches, the egress TCAM size is 2K, and the ingress TCAM size is 4K. The concepts of TCAM slices and single- and double-wide regions do not apply to these switches. For example, the ing-ifacl region can host IPv4, IPv6, or MAC type entries. IPv4 and MAC types occupy one TCAM entry whereas IPv6 types occupy two TCAM entries.

For N9K-X9636C-RX, when PACL uses external TCAM region, the internal TCAM needs to take 2K for ifacl and the ingress RACL-IPv4 can take up to 2044. Additional four entries are required when egress PACL external TCAM region is used.

Guidelines and limitations for ACL TCAM regions

ACL TCAM region sizes have the following guidelines and limitations:

- On Cisco Nexus 9500 Series switches, the default ingress TCAM region configuration has one free 256-entry slice in Cisco NX-OS Release 6.1(2)I1(1). This slice is allocated to the SPAN region in Cisco NX-OS Release 6.1(2)I2(1). Similarly, the RACL region is reduced from 2K to 1.5K in Cisco NX-OS Release 6.1(2)I2(1) to make room for the vPC convergence region with 512 entries.
- To enable RACL or PACL on existing TCAM regions, you must carve the TCAM region beyond 12, 288.
- On Cisco Nexus 9300 Series switches, the X9536PQ, X9564PX, and X9564TX line cards are used to enforce the QoS classification policies applied on 40G ports. It has 768 TCAM entries available for carving in 256-entry granularity. These region names are prefixed with "ns-".
- For the X9536PQ, X9564PX, and X9564TX line cards, only the IPv6 TCAM regions consume double-wide entries. The rest of the TCAM regions consume single-wide entries.
- When a VACL region is configured, it is configured with the same size in both the ingress and egress directions. If the region size cannot fit in either direction, the configuration is rejected.
- On Cisco Nexus 9200 Series switches, the ing-sup region occupies a minimum size of 512 entries, and the egr-sup region occupies a minimum size of 256 entries. These regions cannot be configured to lesser values. Any region size can be carved with a value only in multiples of 256 entries (with the exception of the span region, which can be carved only in multiples of 512 entries).
- RACL v6, CoPP, and multicast have default TCAM sizes and these TCAM sizes must be non-zero on the following Cisco Nexus 9504 and Cisco Nexus 9508 line cards to avoid line card failure during reload:
  - N9K-X96136YC-R
  - N9K-X9636C-RX
  - N9K-X9636Q-R
  - N9K-X9636C-R
- When the egress RACL is beyond 4K, the TCAM carving configuration has to be ingress RACL (RACL) + egress RACL (e-racl) summing to 20480. See the following TCAM carving example:

```
hardware access-list tcam region ifacl 0
hardware access-list tcam region ipv6-ifacl 0
hardware access-list tcam region mac-ifacl 0
hardware access-list tcam region racl 0
hardware access-list tcam region ipv6-racl 0
hardware access-list tcam region span 0
hardware access-list tcam region redirect_v4 0
hardware access-list tcam region redirect_v6 0
hardware access-list tcam region e-racl 20480
```

- You can partially use IPv6 RACL with IPv6 IFCAL. This is applicable to Cisco Nexus N9K-C9508 and N9K-C9504 with N9K-X96136YC-R, N9K-X9636C-R, N9K-X9636Q-R, and N9K-X9636C-RX line cards.

Line card TCAM region support

- The N9K-X9636C-R and N9K-X9636Q-R line cards support a maximum TCAM region size of 12K. If you configure a greater number, the TCAM region is set to 12K.
- The N9K-X96136YC-R and N9K-X9636C-R line cards support egress RACL of 2K.
- The N9K-X9636C-RX line card supports a TCAM region size beyond 12K. If you configure the RACL IPv4 TCAM region to 100K, the TCAM region is set to 12K for the N9K-X9636C-R and N9K-X9636Q-R line cards and to 100K for the N9K-X9636C-RX line card, provided you have set all of the other TCAM regions and made space for the N9K-X9636C-R and N9K-X9636Q-R line cards to accommodate 12K.
- Beginning with Cisco NX-OS Release 10.2(2)F, The N9K-X9636C-R and N9K-X9636Q-R line cards support a maximum TCAM region size of 20K. If you configure a greater number, the TCAM region is re-set to 20K.
- In addition to the internal TCAM, an external TCAM of 128K is available on the N9K-X9636C-RX line card.
- Post reload or upgrade switch operational behavior may be inconsistent if the TCAM utilization for the specific "ing-racl" region exceeds 50% before reload.

Feature-to-region mapping

The following table summarizes the regions that need to be configured for a given feature to work. The region sizes should be selected based on the scale requirements of a given feature.

Table 54: Features per ACL TCAM region

Feature Name	Region Name
Port ACL	ifacl: For IPv4 port ACLs ifacl-udf: For UDFs on IPv4 port ACLs ing-ifacl: For ingress IPv4, IPv6, and MAC port ACLs ing-ifacl: For ingress IPv4, IPv6, MAC port ACLs, and MAC port ACLs with UDF ipv6-ifacl: For IPv6 port ACLs mac-ifacl: For MAC port ACLs

Feature Name	Region Name
Port QoS (QoS classification policy applied on Layer 2 ports or port channels)	qos, qos-lite, rp-qos, rp-qos-lite, ns-qos, e-qos, or e-qos-lite: For classifying IPv4 packets ing-l2-qos: For classifying ingress Layer 2 packets ipv6-qos, rp-ipv6-qos, ns-ipv6-qos, or e-ipv6-qos: For classifying IPv6 packets mac-qos, rp-mac-qos, ns-mac-qos, or e-mac-qos: For classifying non-IP packets **Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve the qos regions and the corresponding ns-*qos regions.
VACL	vacl: For IPv4 packets ipv6-vacl: For IPv6 packets mac-vacl: For non-IP packets
VLAN QoS (QoS classification policy applied on a VLAN)	vqos or ns-vqos: For classifying IPv4 packets ipv6-vqos or ns-ipv6-vqos: For classifying IPv6 packets ing-l3-vlan-qos: For classifying ingress Layer 3, VLAN, and SVI QoS packets mac-vqos or ns-mac-vqos: For classifying non-IP packets **Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve the qos regions and the corresponding ns-*qos regions.
RACL	egr-racl: For egress IPv4 and IPv6 RACLs e-racl: For egress IPv4 RACLs e-ipv6-racl: For egress IPv6 RACLs ing-racl: For ingress IPv4 and IPv6 RACLs racl: For IPv4 RACLs racl-lite: For IPv4 RACLs racl-udf: For UDFs on IPv4 RACLs ipv6-racl: For IPv6 RACLs

Feature Name	Region Name
Layer 3 QoS (QoS classification policy applied on Layer 3 ports or port channels)	l3qos, l3qos-lite, or ns-l3qos: For classifying IPv4 packets ipv6-l3qos or ns-ipv6-l3qos: For classifying IPv6 packets **Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve qos regions and the corresponding ns-*qos regions.
VLAN source or VLAN filter SPAN (for Cisco Nexus 9500 or 9300 Series switches) Rx SPAN on 40G ports (for Cisco Nexus 9300 Series switches only)	span
SPAN filters	ifacl: For filtering IPv4 traffic on Layer 2 (switch port) source interfaces. ifacl-udf: For UDFs on IPv4 port ACLs ipv6-ifacl: For filtering IPv6 traffic on Layer 2 (switch port) source interfaces. mac-ifacl: For filtering Layer 2 traffic on Layer 2 (switch port) source interfaces. racl-udf: For UDFs on IPv4 RACLs vacl: For filtering IPv4 traffic on VLAN sources. ipv6-vacl: For filtering IPv6 traffic on VLAN sources. mac-vacl: For filtering Layer 2 traffic on VLAN sources. racl: For filtering IPv4 traffic on Layer 3 interfaces. ipv6-racl: For filtering IPv6 traffic on Layer 3 interfaces. ing-l2-span-filter: For filtering ingress Layer 2 SPAN traffic ing-l3-span-filter: For filtering ingress Layer 3 and VLAN SPAN traffic
SVI counters **Note** This region enables the packet counters for Layer 3 SVI interfaces.	svi

Feature Name	Region Name
BFD, DHCP relay, or DHCPv6 relay	redirect **Note** BFD uses the ing-sup region while DHCPv4 relay, DHCPv4 snooping, and DHCPv4 client use the ing-redirect region.
CoPP	copp **Note** The region size cannot be 0.
System-managed ACLs	system **Note** The region size cannot be changed.
vPC convergence **Note** This region boosts the convergence times when a vPC link goes down and traffic needs to be redirected to the peer link.	vpc-convergence **Note** Setting this region size to 0 might affect the convergence times of vPC link failures.
Fabric extender (FEX)	fex-ifacl, fex-ipv6-ifacl, fex-ipv6-qos, fex-mac-ifacl, fex-mac-qos, fex-qos, fex-qos-lite
Dynamic ARP inspection (DAI)	arp-ether
IP source guard (IPSG)	ipsg
Multicast PIM Bidir	mcast_bidir
Static MPLS	mpls
Network address translation (NAT)	nat
NetFlow	ing-netflow
OpenFlow	openflow
sFlow	sflow
Supervisor modules	egr-sup: Egress supervisor ing-sup: Ingress supervisor
Policy-Based Routing (PBR)	ing-racl: For matching ingress L3 traffic for PBR.
Layer 2 Intelligent Traffic Director (ITD)	vacl: Programs L2 redirect ACLs at the VLAN level.

Feature Name	Region Name
Layer 3 Intelligent Traffic Director (ITD)	ing-racl: Programs L3 redirect ACLs for ITD.
Enhanced Policy-Based Redirect at L2 (ePBR)	ing-ifacl: Programs L2 redirect ACLs for ePBR L2.
Enhanced Policy-Based Redirect at L3 (ePBR)	ing-racl: Programs L3 redirect ACLs for ePBR L3.

Related Topics

Configure ACL TCAM region sizes, on page 344
Configure TCAM carving, on page 353
Configure TCAM carving - for Cisco NX-OS release 6.1(2)I1(1)

Maximum label sizes supported for ACL types

Use this reference to review maximum label sizes supported for ACL types.

Cisco NX-OS switches support the following label sizes for the corresponding ACL types:

Table 55: ACL types and maximum label sizes

ACL Types	Direction	Label	Label Type
RACL/PBR/VACL/ L3-VLAN QoS/L3-VLAN SPAN ACL	Ingress	62	BD
PACL/L2 QoS/L2 SPAN ACL	Ingress	62[1] **Note** This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.	IF
RACL/VACL/L3-VLAN QoS	Egress	254	BD
L2 QoS	Egress	31	IF
RACL	Ingress	510	L3

[1] The label size can be increased to 62 when you enter the **hardware access-list tcam label ing-ifacl 6** command and reload the switch.

Beginning with Cisco NX-OS Release 9.3(6), the **hardware access-list tcam label ing-ifacl 6** command is introduced and is applicable only for Cisco Nexus 9300-FX platform switches.

Beginning with Cisco NX-OS Release 10.1(2), the **hardware access-list tcam label ing-ifacl 6** command is also supported on Cisco Nexus 9300-FX2 platform switches.

Beginning with Cisco NX-OS Release 10.4(3)F, the **hardware access-list tcam label ing-ifacl 6** command is also supported on Cisco Nexus 9300-FX3, GX, GX2, H2R, H1 platform switches.

Prerequisites for IP ACLs

Use this reference to review the prerequisites for IP ACLs.

IP ACL prerequisites

IP ACLs have the following prerequisites:

- You must be familiar with IP addressing and protocols to configure IP ACLs.
- You must be familiar with the interface types that you want to configure with ACLs.

Guidelines and limitations for IP ACLs

Use this reference to review the guidelines and limitations for IP ACLs.

IP ACL configuration guidelines and limitations

IP ACLs have the following configuration guidelines and limitations:

Note For more information about the Cisco Nexus 9000 series platform switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.

- Beginning with Cisco NX-OS Release 10.2(1)F, Egress PACL is supported on the Cisco Nexus 9364D-GX2A, and 9332D-GX2B switches.
- If you configure egress PACL and egress VACL on the same interface, only egress VACL is enabled.
- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This recommendation is especially useful for ACLs that include more than 1000 rules. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.
- Configuring a IPv4 PACL in the range of 12K to 64K is supported on Cisco Nexus 9500 Series switches with -RX line cards.
- Duplicate ACL entries with different sequence numbers are allowed in the configuration. However, these duplicate entries are not programmed in the hardware access-list.
- Only 62 unique ACLs can be configured. Each ACL takes one label. If the same ACL is configured on multiple interfaces, the same label is shared. If each ACL has unique entries, the ACL labels are not shared, and the label limit is 62. This is not applicable to Cisco Nexus 9500 Series switches.
- IPv4 and IPv6 RACLs and PBR policies applied to multiple Layer 3 interfaces share the same label space. If the exact same set of ACLs and PBR policies are applied across multiple interfaces, a shared

label is used, optimizing TCAM utilization. However, modifying the ACL or PBR on any one of these interfaces breaks the label sharing for that interface, requiring additional TCAM resources to allocate a new label space. This can lead to TCAM exhaustion errors if sufficient resources are not available.

- The IPv6 extension header hop-by-hop filter is not supported in IPv6 ACLs.

ACL processing, VTY, and VXLAN guidelines

- Usually, ACL processing for IP packets occurs on the I/O modules, which use hardware that accelerates ACL processing. In some circumstances, processing occurs on the supervisor module, which can result in slower ACL processing, especially during processing that involves an ACL with many rules. Management interface traffic is always processed on the supervisor module. If IP packets in any of the following categories are exiting a Layer 3 interface, they are sent to the supervisor module for processing:
  - Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.
  - IPv4 packets that have IP options (other IP packet header fields following the destination address field).
  - IPv6 packets that have extended IPv6 header fields.

  Rate limiters prevent redirected packets from overwhelming the supervisor module.
- When you apply an ACL that uses time ranges, the device updates the ACL entries whenever a time range that is referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.
- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally. For more information about VLAN interfaces, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.
- The VTY ACL feature restricts all traffic for all VTY lines. You cannot specify different traffic restrictions for different VTY lines. Any router ACL can be configured as a VTY ACL.
- An egress VTY ACL (an IP ACL applied to the VTY line in the outbound direction) prevents the switch from copying files using a file transfer protocol (TFTP, FTP, SCP, SFTP, etc.) unless the file transfer protocol is explicitly permitted within the egress VTY ACL.
- When you apply an undefined ACL to an interface, the system treats the ACL as empty and permits all traffic.
- IP tunnels do not support ACLs or QoS policies.
- The following guidelines apply to ACLs for VXLANs:
  - Ingress port ACLs applied on a Layer 2 port for traffic in the access to a network direction (Layer 2 to Layer 3 encapsulation path) are supported on the inner payload.
  - We recommend using port ACLs on the access side to filter out traffic entering the overlay network.
  - Ingress router ACLs applied on an uplink Layer 3 interface matching on the inner or outer payload in the network to access direction (Layer 3 to Layer 2 decapsulation path) are not supported.
  - Egress router ACLs applied on an uplink Layer 3 interface matching on the inner or outer payload in the access to a network direction (encapsulation path) are not supported.

- Cisco Nexus 9300 and 9500 Series switches, and Cisco Nexus 9200 and 9300-EX Series switches have the following limitations for ACL options that can be used on VXLAN traffic:
  - Does not support egress port ACLs applied on a Layer 2 port for traffic in the network to access direction (decapsulation path).
  - Supports ingress VACLs applied on a VLAN for traffic in the access to a network direction (encapsulation path).
  - Supports egress VACLs applied on a VLAN for traffic in the network to access direction (decapsulation path).
  - Supports ingress RACLs applied on a tenant or server facing SVI for traffic in the access to network direction (encapsulation path).
  - Supports egress RACLs applied on a tenant or server facing SVI for traffic in the network to access direction (decapsulation path).

ACL logging and egress RACL limitations

- IPv6 ACL logging is not supported for egress PACL.
- IPv4 ACL logging in the egress direction is not supported.
- ACL logging for VACLs is not supported.
- ACL logging applies to port ACLs configured by the **ip port access-group** command and to router ACLs configured by the **ip access-group** command only.
- The total number of IPv4 ACL flows is limited to a user-defined maximum value to prevent DoS attacks. If this limit is reached, no new logs are created until an existing flow finishes.
- The number of syslog entries that are generated by IPv4 ACL logging is limited by the configured logging level of the ACL logging process. If the number of syslog entries exceeds this limit, the logging facility might drop some logging messages. Therefore, IPv4 ACL logging should not be used as a billing tool or as an accurate source of the number of matches to an ACL.
- Egress router ACLs are not supported on Cisco Nexus 9300 Series switch uplink ports.

TCAM resource sharing and HTTP method limitations

- For Network Forwarding Engine (NFE)-enabled switches, ingress RACLs matching the outer header of the tunnel interface are not supported.
- If the same QoS policy and ACL are applied to multiple interfaces, the label is shared only when the QoS policy is applied with the no-stats option.
- The switch hardware does not support range checks (Layer 4 operations) in the egress TCAM. Therefore, ACL and QoS policies with a Layer 4 operations-based classification must be expanded to multiple entries in the egress TCAM.

  The switch hardware supports only up to 16 Layer 4 operands. Make sure to consider this limitation for egress TCAM space planning. For more information see the Logical operators and logical operation units, on page 314 section.

- For Cisco Nexus X96136YC-R, X9636C-RX, X9636C-RX, and X9636Q-R line cards, run the **hardware profile acl-eg-ext module all** command before applying **eg-racl-v6** configuration on a SVI or port object on an EoR switch.
- TCAM resources are shared in the following scenarios:
  - When a routed ACL is applied to multiple switched virtual interfaces (SVIs) in the ingress direction.
  - When a routed ACL is applied to multiple layer 2 interfaces in the ingress or egress direction.
- TCAM resources are not shared in the following scenarios:
  - VACL (VLAN ACL) is applied to multiple VLANs.
  - Routed ACL is applied to multiple SVIs in the egress direction.
- Access-lists based on HTTP methods are not supported on the Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2, 9300-FXP, 9300-GX platform switches and the 9500 switches with the X9700-EX and X9700-FX line cards. For all these switches, you must use UDF-based ACLs.
- HTTP methods are not supported on FEX ports.

Platform-specific ACL and TCAM limitations

- The following guidelines and limitations apply to Cisco Nexus 9200 and 9300-EX Series switches:
  - Egress MAC ACLs are not supported.
  - Egress RACLs are not supported on an interface if the packet matches the outer header of the tunnel interface on the device where the tunnel is originating the traffic.
  - Ingress RACLs matching the outer header of the tunnel interface are not supported.
  - IP length-based matches are not supported.
  - All ACL-based features cannot be enabled at the same time.
  - 16 Layer 4 operations are supported.
  - Layer 4 operations are not supported on egress TCAM regions.
  - The MAC compression table size is 4096 + 512 overflow TCAM.
  - An overlap of MAC addresses and MAC masks is rejected.
  - The ACL log rate limiter does not have any hardware counters for transmitted or dropped packets.
  - The ACL log rate limiter is implemented at the per-TCAM entry level (instead of using aggregated rate limiting), and the default is 1 pps.
  - The Network Address Translation (NAT) exception counters are zero.
  - Only PACL redirects are supported for TAP aggregation. VACL redirects are not supported.
  - Only three of the following four features can be supported at a time: DHCPv4 snooping or relay, DHCPv6 relay, ARP snooping, VXLAN. The first three configured features take effect, but the fourth one will fail because all three bridge domain label bits are already in use.

- The following guidelines and limitations apply to Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX series switches:
  - The MAC compression table size is 4096 + 512 overflow TCAM.
  - An overlap of MAC addresses and MAC masks is rejected.
- Cisco Nexus 9504 and Cisco Nexus 9508 switches with -R line cards do not support the following TCAM:
  - All FEX related TCAM
  - All xxx-lite related TCAM region
  - Ranger related TCAM
  - All FCoE related TCAM
- TCAM carving configuration of the ing-netflow region can be performed on -FX line cards. -EX line cards have a default ing-netflow region TCAM carving of 1024 and cannot be configured otherwise. For ports on the -EX and -FX line cards, the suggested maximum for the ing-netflow region is 1024.
- On the Cisco Nexus 9200 and 9300-EX platform switches, router ACL with the ACL log option will not take into effect as the sup-redirect ACLs have higher priority for the traffic that is destined to SUP.
- On the Cisco Nexus 9300-GX platform switches, dot1q VLAN with ACL redirect supports only the VLAN IDs from 1 to 509.

  If PACL redirect or TapAgg is configured, the **switchport access vlan** *vlan-id* command supports only the vlan IDs from 1 to 509.
- For traffic destined to the FHRP VIP and ingressing on FHRP standby which matches an ACL log enabled ACE designed to permit the traffic, the Cisco Nexus 9000 Series switch drops this packet.

Cisco Nexus 9300 and 9500 ACL support guidelines

- Cisco Nexus 9364D-GX2A, and 9332D-GX2B switches do not support the following on egress router ACL:
  - UDF to support ICMP Type Match.
  - ACL log-on egress
  - Egress IPv4 router ACL with additional filter option tcp/udp ports with lt/gt
  - Egress IPv4 router ACL with additional filter option tcp/udp ports with neq
  - Egress IPv4 router ACL with extra filter option tcp/udp ports with range
  - Egress IPv4 router ACL with a flag
  - Egress router ACL on an external TCAM
  - Egress PACL support
  - Statistics support
  - Label sharing

- Cisco Nexus 9500 platform switches with -R and -RX line cards have the following guidelines:
  - Atomic ACL update is supported for all the ingress ACL features except for the Multihop BFD and CoPP features.
  - Atomic ACL update is not supported for the egress ACL features.
  - Label sharing is supported only for the same policy on different interfaces within the same ASIC.
  - In Cisco NX-OS Release 9.2(3), ACL statistics are supported for the following:
    - PACL - IPv4 (including system ACL for both, internal, and external TCAM)
    - Router ACL - IPv4 (internal TCAM for both, ingress RACL-IPv4 and egress RACL-IPv4)
    - Only 2K counters are supported in the egress.
  - ACL statistics are not supported for the following:
    - BFD
    - DHCP - IPv4 and IPv6
    - PACL-MAC
    - PACL- IPv6
    - PBR - IPv4 and IPv6
    - RACL-IPv6
    - RACL-IPv4 when using an external TCAM
- ACL label sharing works on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 and 9300C platform switches with following limitations:
  - ACL statistics are disabled by default. However, statistics are enabled by default only for QoS policies.
  - ACL target (port / VLAN / SVI etc.,) must be on the same slice, and port.
  - Additionally, label space is shared with following features:
    - Ingress RACL, PBR and Ingress L3 QoS
    - Ingress PACL, Ingress L2 QoS
    - Egress RACL, Egress QoS



Note For label sharing to work, ensure that the same set of features are supported on interfaces.

- When you enable the counters for the ACL TCAM entries using the hardware profile acl-stats module *xx* command, the input discard field in the show interface is always zero. This limitation is applicable only to the Cisco Nexus 9500 platform switches with -R and -RX line cards.

- Cisco Nexus 9500 platform switches with -R and -RX line cards do not support the following:
  - Egress atomic updates
  - Egress router ACL on external TCAM
  - Egress router ACL with UDF
  - Router ACL v6 counters for both egress and ingress
  - Egress and ingress router ACL IPv6 with l4 ops
  - Egress router ACL on subinterface
  - Egress and ingress router ACL with IPv6 ICMP Type and Code
  - IPv6 ingress router ACL with tcp-flag
  - IPv4 router ACL with extra option
- In Cisco NX-OS Release 9.3(3), egress IPv4 RACLs support the following on Cisco Nexus 9504 and 9508 switches with -R and -RX line cards:
  - TCP flags
  - ICMP Type and Code
  - ACL logs

IPv6 egress ACL and PACL redirect guidelines

- IPv6 Egress ACL support the following on Cisco Nexus 9504 and 9508 switches with -R and -RX line cards:
  - Layer 4 Protocol
  - TCP flags
  - Fragment
  - ACL logs for IPv4
  - IPv6 header fields

  The following limitations are applicable for the IPv6 egress ACL:
  - Port groups and Layer 4 Operations are not supported. The ranges expand to multiple ACE entries for eg-racl-ipv6.
  - Address group defined host is not supported.
  - Counters are not supported.
  - Egress IPv6 RACL is not supported on sub-interfaces and external TCAM.
  - Atomic updates are not supported.
  - VXLAN is not supported when acl-eg-ext is enabled.

- PACL redirects are supported on Cisco Nexus 9300-GX switches. The following limitations are applicable:
  - To support PACL redirects, you must run the **mode tap-agg** command on the ingress tap interface.
  - To support the MPLS strip feature, the **mpls strip** and the **hardware acl tap-agg** commands must be configured and the switch reloaded.
  - For double tag VLAN, the range of the second VLAN is 2-510.
  - MPLS strip with dot1q VLAN is not supported.
  - The redirect port carries the tag if the incoming packet is tagged, even when the redirect port is configured as an access port.
  - TapAgg redirect is not supported for deny ACE.
- In Cisco NX-OS Release 10.1(2), PACL redirect feature is not supported in mixed mode on Cisco Nexus X9736C-FX, X9788TC-FX, and X97160YC-EX line cards.
- Egress ACL does not support traffic that is destined to the IP address of the second VLAN in inter-VLAN routing flow.
- In Cisco Nexus 9300-EX/FX/FX2/FX3/GX platform switches and 93180YC-FX switches, RACLs cannot match on packets with multicast MAC destination addresses on Layer-3 interfaces. Use the **ignore routable** command when you configure the ACL to remove the routable qualifier. However, when you add ignore-routable to a RACL and apply on SVI, RACL will match with the bridged packets too.
- The Get operation provides incomplete data/no sequence number when wildcard bits are in A.B.C.D format. This is a known behavior. The Open Config model does not have srcPrefixMask/dstPrefixMask. Also, no meaningful value can be returned for prefix length because it is not possible to convert the mask to prefix length for non-contiguous mask.
- The ing-sup region occupies a minimum size of 512 entries, and the egr-sup region occupies a minimum size of 256 entries. These regions cannot be configured to lesser values. Any region size can be carved with a value only in multiples of 256 entries (with the exception of the span region, which can be carved only in multiples of 512 entries).
- Beginning with Cisco NX-OS Release 9.3(9), the Layer 3 subinterface egress router ACL feature is supported on Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 platform switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, the Layer 3 subinterface egress router ACL feature is supported on Cisco Nexus 9300 Series platform switches.

Release 10.x ACL feature support and limitations

- For egress RACL V6 region, you need to configure **hw profile mdb-balanced-exem** .
- From Cisco NX-OS Release 10.2(2)F, the egress PACL feature is supported on egress router ACL on Cisco Nexus 9300-GX platform switches and 93108TC-FX3P, and 93180YC-FX3 switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, the egress filtering on subinterfaces feature supports Layer 3 subinterface egress router ACL on Cisco Nexus 9300-FX/FX2/FX3/GX/GX2 platform switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, the increase ACL LOU threshold feature supports configurable LOU threshold limit for ACL configuration on Cisco Nexus 9500-R platform switches.

- Beginning with Cisco NX-OS Release 10.3(1)F, ITD NAT VRF configuration is provided on Cisco Nexus 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 10.3(1)F, ACL Consistency Checker support is provided on Cisco Nexus 9808 switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, ACL Consistency Checker is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, ACL Consistency Checker support is provided on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.
- Cisco Nexus 9808/9804 switches have the following limitations for ACL SUP support:
  - In ACE, match COS and match VLAN are not supported.
  - Ensure nd-na and nd-ns packets matches with IPv6 ACE.
  - TCAM carving is not supported. However, you can view the currently allocated TCAM for each individual feature. To view the currently allocated TCAM, use the **show hardware access-list resource utilization** command.
  - Central TCAM is supported. However, it is shared among both ingress and egress ACLs.
  - UDF is not supported.
  - LOUs is not supported
  - IPv6 fragments are not matched in egress RACL.
  - L2 ACL features are not supported.
  - ODM merge is not supported.
  - IPv6 next header match will match the innermost next header, the pipeline is able to parse.
  - Only 16 unique burst values are supported. Due to this, user configured burst values are mapped to nearest 2 power value (min 64 to max 65536).
  - Each IPv6 ACL is limited to 1,000 ACEs. This applies to all IPv6 ACLs (RACL, QoS or SPAN filter). No such limitation applies for IPv4 ACL.
- Beginning with Cisco NX-OS Release 10.3(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus 9808 switches. However, UDF is not supported.
  - Beginning with Cisco NX-OS Release 10.4(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus X98900CD-A, and X9836DM-A line cards with Cisco Nexus 9808 switches.

ACL statistics, CoPP, and redirect limitations

- Beginning with Cisco NX-OS Release 10.4(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards. However, UDF is not supported.

- To display ACL statistics on Cisco Nexus 9808/9804 switches, the **hardware access-list tcam per-entry-stats template racl** command has to be enabled and reload of switch is required after configuring the **hardware access-list tcam per-entry-stats template racl** command.
- Cisco Nexus 9808/9804 switches have the following limitations for CoPP support:
  - CoPP policer stats for Stage-1, Stage-2, and Stage-3 are in PPS.



Note CoPP Stage3 stats gets reset to zero after system switchover.

  - Stage-2 output is at LC/Module level, and Stage-3 output is at SUP/CPU level.
  - Policer rates and CoS changes are supported in Custom CoPP.
  - Fabrics/FMs are not involved in in-band path.
  - CoPP Consistency checker is not supported.
  - Supported CIR minimum value is 125 PPS.
  - CIR 0 is supported.
  - There are no per entry statistics for CoPP ACL entries.
  - MACsec packets are mapped to BPDU queue.
  - Only 16 unique burst values are supported. Due to this, user configured burst values are mapped to nearest 2 power value (min 64 to max 65536).
- Beginning with Cisco NX-OS Release 10.4(3)F, Cisco Nexus 9364C-H1 switches have the following limitations for CoPP support:
  - There is only one stage of policing and CoPP policer stats for Stage-1 are in PPS
  - Policer rate and CoS changes are supported in Custom CoPP
  - Policer rates are in multiples of 572
  - CoPP Consistency checker is not supported
- Deny ACE in MAC ACL or PACL (Port ACL) with redirect option is not supported on Cisco Nexus 9000 Series switches.
- Beginning with Cisco NX-OS Release 10.3(3)F, ACL auto name completion feature is supported on Cisco Nexus 9000 Series platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is provided for applying the IP or IPv6 ACL rule priority over SUP rule on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, C9364C, C9332C, and Cisco Nexus 9500 with 9700-EX/ FX/GX line cards.

Security ACL and flexible TCAM guidelines

- Beginning with Cisco NX-OS Release 10.4(1)F, Security ACL is supported on the Cisco Nexus 9332D-H2R switches.

- Beginning with Cisco NX-OS Release 10.4(2)F, Security ACL is supported on the Cisco Nexus 93400LD-H1 switches.
- Beginning with Cisco Nexus Release 10.4(3)F, Security ACL is supported on the Cisco Nexus 9364C-H1 switches.
- The flexible TCAM configuration is supported on the Cisco Nexus 9332D-H2R , 9364C-H1, and 93400LD-H1 switches. Ingress and Egress regions uses 14K shared TCAM on this platform switch.



Note

- In Cisco Nexus 9332D-H2R , 9364C-H1, and 93400LD-H1 switches, there is no per-direction total TCAM size limit for the shared TCAM model as in Cisco Nexus 9300-FX3, GX2 switches. Limit 14K applies to the sum of both direction TCAM sizes.
- 14K is split into 28 blocks of 512 entries each. Allocation to ingress or egress direction TCAM happens in block size granularity. If the sum of ingress region sizes configured is multiple of 256, additionally 256 is allocated (and unused) for block level allocation granularity. Same applies for the sum of egress region sizes configured.

- Beginning with Cisco NX-OS Release 10.4(2)F, the new ACE configuration filter route-tag default-route is supported. This configuration enables the filtering of traffic for QoS classification when traffic matches the default route. The following guidelines and limitations are applicable for this enhancement:
  - ACL using **route-tag default-route** command is supported for class-map, QoS type only. This feature is supported for following switches:
    - Cisco Nexus 9300-FX3
    - Cisco Nexus 9300-GX
    - Cisco Nexus 9300-GX2
    - Cisco Nexus 93400LD-H1
    - Cisco Nexus 9332D-H2R
- The **route-tag default-route** configuration is supported for IPv4 and IPv6 access-list ACEs only.
- You cannot configure both PBR ACLs and default-route on a switch. Both configurations cannot co-exist.
- If you have configured the **hardware access-list tcam pbr match-default-route** command on a switch, you cannot configure the **route-tag default-route** command in PBR policy configuration.
- Ensure that you do not have **FabricPath to VXLAN** feature configured to use the **hardware access-list tcam label ing-ifacl 6** command.

Security group ACL support guidelines

- Beginning with Cisco NX-OS Release 10.4(3)F, Class E IP address is supported for Security Group ACL with Endpoint Security Group (ESG).

ACL guidelines and limitations for Cisco Nexus 9364E-SG2 Series switches

Default settings for IP ACLs

Use this reference to review the default settings for IP ACLs.

Table 56: Default IP ACL parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default
IP ACL entries	1024
ACL rules	Implicit rules apply to all ACLs
Object groups	No object groups exist by default
Time ranges	No time ranges exist by default

Related Topics

Implicit rules for IP and MAC ACLs, on page 311

Configuring IP ACLs

Explains configuring IP ACLs.

Create an IP ACL

You can create an IPv4 access control list (ACL) or IPv6 access control list (ACL) on the device and add rules to it.

Before you begin

Use the Session Manager to perform the ACL configuration. This feature helps you verify the ACL configuration and check that all required resources are available before you commit them to the running configuration. The Session Manager is especially useful if your access control list (ACL) includes more than 1,000 rules.

Procedure

Step 1 Create the IP ACL in global configuration mode. Then, enter the IP ACL configuration mode using one of these commands:

- **ip access-list** *name*
- **ipv6 access-list** *name*

Example:

```
switch#configure terminal
switch(config)#ip access-list acl-01
switch(config-acl)#
```

The *name* argument can be up to 64 characters.

Notice
The names *dynamic* , *expanded* , and *summary* are reserved for system-defined access lists. Do not use these names for user-defined ACLs, as this can cause conflicts when displaying or verifying your configuration.

Step 2 (Optional) Optimize fragment handling for noninitial fragments using the **fragments** {**permit-all** | **deny-all**} command.

When a device applies an ACL containing the **fragments** command to traffic, the **fragments** command only matches noninitial fragments that do not match any explicit **permit** or **deny** commands in the ACL.

Example:

```
switch(config-acl)#fragments permit-all
```

Step 3 Create a rule in the IP ACL using the [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*} command.

You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4,294,967,295.

The **permit** and **deny** commands support many ways of identifying traffic.

For IPv4 and IPv6 access lists, you can specify a source and destination IPv4 or IPv6 prefix, which matches only on the first contiguous bits, or you can specify a source and destination IPv4 or IPv6 wildcard mask, which matches on any bit in the address. IPv6 wildcard masks are supported for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FXP switches and the Cisco Nexus 9364C switch .

Example:

```
switch(config-acl)#permit ip 192.168.2.0/24 any
```

Example:

```
switch(config-acl)#10 permit ipv6 2001:db8::1 2001:db8::2 2001:db8::3 2001:db8::4
```

Step 4 (Optional) Specify whether the device maintains global statistics for packets that match the rules in the ACL by using **statistics per-entry** command.

Example:

```
switch(config-acl)#statistics per-entry
```

Note
Beginning Cisco NX-OS Release 9.2(3), ACL statistics is supported on Cisco Nexus 9500 platform switches with -R line cards. You must complete this step if you are using Cisco Nexus 9500 platform switches.

Step 5 Enable counters for the ACL TCAM entries on both the internal and external TCAMs using the **hardware profile acl-stats module** *xx* command.

Example:

```
switch(config-acl)#exit
switch(config)#hardware profile acl-stats module 10
```

Note
This command is applicable only for Cisco Nexus 9500 platform switches with -R and -RX line cards . VLAN and SVI statistics are lost when you enable the counters.

Step 6 Reload the switch using the **reload module** *xx* command.

Note
For Cisco Nexus 9500 platform switches, this command is optional. Only the modules where the **hardware profile ac-stats** is applied must be reloaded.

Example:

```
switch#reload module 10
```

Step 7 Enable the filtering of multicast traffic on Cisco Nexus 9300-EX and 9300-FX platform switches using the **ignore routeable** command.

Example:

```
switch#configure terminal
switch(config)#ignore routeable
```

Step 8 (Optional) Enter one of the following commands:

- **show ip access-lists** *name*
- **show ipv6 access-lists** *name*

Example:

```
switch#show ip access-lists acl-01
```

Displays the IP ACL configuration.

Step 9 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Modify an IP ACL

You can add or remove rules in an existing IPv4 or IPv6 ACL. However, you cannot change existing rules. To modify a rule, first remove it, then recreate it with the desired changes.

To add additional rules between existing ones when the current sequence numbering does not allow it, use the **resequence** command to reassign sequence numbers.

Before you begin

We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the necessary resources are available before committing changes. It is especially helpful for ACLs containing more than 1,000 rules.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter IP ACL configuration mode for the ACL that you specify by name using one of the following commands:

- **ip access-list** *name*
- **ipv6 access-list** *name*

Example:

```
switch(config)#ip access-list acl-01
switch(config-acl)#
```

Step 3 (Optional) Create a rule in the IP ACL. Specify a sequence number to assign the rule's position in the ACL with the [*sequence-number*] {**permit** | **deny**} *protocol source destination* command.

Example:

```
switch(config-acl)#100 permit ip 192.168.2.0/24 any
```

Without a sequence number, the rule is added to the end of the ACL. The *sequence-number* argument accepts whole numbers from 1 to 4,294,967,295.

The **permit** and **deny** commands support many ways of identifying traffic.

Step 4 (Optional) Optimizes fragment handling for noninitial fragments using the fragments [**no**] **fragments** {**permit-all** | **deny-all**} command.

The no option removes fragment-handling optimization.

Example:

```
switch(config-acl)#fragments permit-all
```

When a device applies to traffic an ACL that contains the **fragments** command, the **fragments** command only matches noninitial fragments that do not match any explicit **permit** or **deny** commands in the ACL.

Step 5 (Optional) Remove the rule from the IP ACL using the **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*} command.

Example:

```
switch(config-acl)#no 80
```

The **permit** and **deny** commands support many ways of identifying traffic.

Step 6 (Optional) Specify that the device maintains global statistics for packets that match the rules in the ACL using the [**no**] **statistics per-entry** command.

The no option stops the device from maintaining global statistics for the ACL.

Example:

```
switch(config-acl)#statistics per-entry
```

Step 7 (Optional) Display the IP ACL configuration using one of the following commands:

- **show ip access-lists** *name*
- **show ipv6 access-lists** *name*

Example:

```
switch#show ip access-lists acl-01
```

Step 8 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Change sequence numbers in an IP ACL, on page 342

Create a VTY ACL

You can configure a VTY ACL to control access to all IPv4 or IPv6 traffic over all VTY lines in the ingress or egress direction.

Before you begin

Set identical restrictions on all the virtual terminal lines because a user can connect to any of them.

We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration, which is especially useful for ACLs that include more than about 1000 rules.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 { **ip** | **ipv6** } **access-list** *name*

Example:

```
switch(config)#ip access-list vtyacl
```

Creates an ACL and enters IP access list configuration mode for that ACL. The maximum length for the *name* argument is 64 characters.

Step 3 { **permit** | **deny** } *protocol source destination* [**log**] [**time-range** *time*]

Example:

```
switch(config-ip-acl)#permit tcp any any
```

Creates an ACL rule that permits TCP traffic from and to the specified sources.

Step 4 **exit**

Example:

```
switch(config-ip-acl)#exit
switch(config)#
```

Exits IP access list configuration mode.

Step 5 **line vty**

Example:

```
switch(config)#line vty
switch(config-line)#
```

Specifies the virtual terminal and enters line configuration mode.

Step 6 { **ip** | **ipv6** } **access-class** *name* { **in** | **out** }

Example:

```
switch(config-line)#ip access-class vtyacl out
```

Restricts incoming or outgoing connections to and from all VTY lines using the specified ACL. The maximum length for the *name* argument is 64 characters.

Step 7 (Optional) **show** { **ip** | **ipv6** } **access-lists**

Example:

```
switch#show ip access-lists
```

Displays the configured ACLs, including any VTY ACLs.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Change sequence numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

Before you begin

We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **resequence** {**ip** | **ipv6**} **access-list** *name starting-sequence-number increment*

Example:

```
switch(config)#resequence access-list ip acl-01 100 10
```

Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295.

Step 3 (Optional) **show ip access-lists** *name*

Example:

```
switch#show ip access-lists acl-01
```

Displays the IP ACL configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Remove an IP ACL

You can remove an IP ACL from the device.

Before you begin

Ensure that you know whether the ACL is applied to an interface. The device allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the removed ACL to be empty. Use the **show ip access-lists** command or the **show ipv6 access-lists** command with the summary keyword to find the interfaces that an IP ACL is configured on.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **no ip access-list** *name*
- **no ipv6 access-list** *name*

Example:

```
switch(config)#no ip access-list acl-01
```

Removes the IP ACL that you specified by name from the running configuration.

Step 3 (Optional) Enter one of the following commands:

- **show ip access-lists** *name* **summary**
- **show ipv6 access-lists** *name* **summary**

Example:

```
switch#show ip access-lists acl-01 summary
```

Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure ACL TCAM region sizes

You can change the size of the ACL TCAM regions in the hardware. After TCAM carving, for the TCAM to qualify, you must save the configuration and reload the switch. If the switch has a faulty module, saving the configuration will take a longer time.

You can use this procedure for all Cisco Nexus 9200, 9300, and 9500 Series switches , except for NFE2-enabled devices (such as the X9432C-S 100G line card and the C9508-FM-S fabric module), which must use TCAM templates to configure ACL TCAM region sizes. For more information on using TCAM templates, see "Using Templates to Configure ACL TCAM Region Sizes."



Note

- Once you apply a template, the **hardware access-list tcam region** command in this section will not work. You must uncommit the template in order to use the command.
- The **hardware access-list tcam region** command for the Multicast PIM Bidir feature is applicable only to the Broadcom-based Cisco Nexus 9000 Series switches.
- For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide* .

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **hardware access-list tcam region** *region tcam-size*

Example:

```
switch(config)#hardware access-list tcam region mpls 256
```

Note
This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

Changes the ACL TCAM region size. These are the available regions:

Table 57: Configure ACL TCAM region sizes

TCAM Region / Argument	Description
n9k-arp-acl	Configures the rate limit for ARP packets entering an interface on their way to the CPU.
arp-ether	Configures the size of the ARP/Layer 2 Ethertype TCAM region.
copp	Configures the size of the CoPP TCAM region.
e-flow	Configures the size of the egress flow counters TCAM region.
egr-copp	Configures the size of the egress CoPP TCAM region.
egr-racl	Configures the size of the egress IPv4 or IPv6 RACL TCAM region. Cisco Nexus 9200 switches only.
egr-sup	Configures the size of the egress supervisor TCAM region. Cisco Nexus 9200 switches only.
e-ipv6-qos	Configures the size of the IPv6 egress QoS TCAM region.
e-ipv6-racl	Configures the size of the IPv6 egress RACL TCAM region.
e-mac-qos	Configures the size of the egress MAC QoS TCAM region.
e-qos	Configures the size of the IPv4 egress QoS TCAM region.
e-qos-lite	Configures the size of the IPv4 egress QoS lite TCAM region.
e-racl	Configures the size of the IPv4 egress RACL TCAM region.
fex-ifacl	Configures the size of the FEX IPv4 port ACL TCAM region.
fex-ipv6-ifacl	Configures the size of the FEX IPv6 port ACL TCAM region.
fex-ipv6-qos	Configures the size of the FEX IPv6 port QoS TCAM region.
fex-mac-ifacl	Configures the size of the FEX MAC port ACL TCAM region.
fex-mac-qos	Configures the size of the FEX MAC port QoS TCAM region.
fex-qos	Configures the size of the FEX IPv4 port QoS TCAM region.
fex-qos-lite	Configures the size of the FEX IPv4 port QoS lite TCAM region.
fhs	Configures the size of the FHS TCAM region on Cisco Nexus 9300 and 9500 Series switches.
flow	Configures the size of the ingress flow counters TCAM region.

TCAM Region / Argument	Description
ifacl	Configures the size of the IPv4 port ACL TCAM region.
ifacl-udf	Configures the size of the IPv4 port ACL UDF TCAM region. Cisco Nexus 3232C and 3264Q switches only.
ing-ifacl	Configures the size of the ingress IPv4, IPv6, or MAC port ACL TCAM region. Cisco Nexus 9200, 9300, and 9300-EX switches only.
ing-l2qos	Configures the size of the ingress Layer 2 QoS TCAM region. Cisco Nexus 9200 switches only.
ing-l2-span-filter	Configures the size of the ingress Layer 2 SPAN filter TCAM region. Cisco Nexus 9200 and 9300-EX switches only.
ing-l3-span-filter	Configures the size of the ingress Layer 3 and VLAN SPAN filter TCAM region. Cisco Nexus 9200 and 9300-EX switches only.
ing-l3-vlan-qos	Configures the size of the ingress Layer 3, VLAN, and SVI QoS TCAM region. Cisco Nexus 9200 switches only.
ing-netflow	Configures the size of the NetFlow TCAM region.
ing-racl	Configures the size of the IPv4 or IPv6 ingress RACL TCAM region. Cisco Nexus 9200 switches only.
ing-redirect	Configures the size of the redirect TCAM region for DHCPv4 relay, DHCPv4 snooping, and DHCPv4 client. Cisco Nexus 9200 switches only.
ing-sup	Configures the size of the ingress supervisor TCAM region. Cisco Nexus 9200 switches only.
ipsg	Configures the size of the IP Source Guard SMAC-IP binding TCAM region.
ipv6-ifacl	Configures the size of the IPv6 port ACL TCAM region.
ipv6-l3qos	Configures the size of the IPv6 Layer 3 QoS TCAM region.
ipv6-qos	Configures the size of the IPv6 port QoS TCAM region.
ipv6-racl	Configures the size of the IPv6 RACL TCAM region.
ipv6-vacl	Configures the size of the IPv6 VACL TCAM region.
ipv6-vqos	Configures the size of the IPv6 VLAN QoS TCAM region.
l3qos	Configures the size of the IPv4 Layer 3 QoS TCAM region.
l3qos-lite	Configures the size of the IPv4 Layer 3 QoS lite TCAM region.
mac-ifacl	Configures the size of the MAC port ACL TCAM region.
mac-l3qos	Configures the size of the MAC Layer 3 QoS TCAM region.
mac-qos	Configures the size of the MAC port QoS TCAM region.
mac-vacl	Configures the size of the MAC VACL TCAM region.
mac-vqos	Configures the size of the MAC VLAN QoS TCAM region.
mcast_bidir	Configures the size of the multicast PIM Bidir TCAM region.

TCAM Region / Argument	Description
`mpls`	Configures the size of the static MPLS TCAM region.
`nat`	Configures the size of the NAT TCAM region.
`openflow`	Configures the size of the OpenFlow TCAM region.
`qos`	Configures the size of the IPv4 port QoS TCAM region.
`qos-lite`	Configures the size of the IPv4 port QoS lite TCAM region.
`racl`	Configures the size of the IPv4 RACL TCAM region.
`racl-lite`	Configures the size of the IPv4 RACL lite TCAM region. Cisco Nexus 3232C and 3264Q switches only.
`racl-udf`	Configures the size of the IPv4 RACL UDF TCAM region. Cisco Nexus 3232C and 3264Q switches only.
`redirect`	Configures the size of the redirect TCAM region.
`redirect-tunnel`	Configures the size of the redirect-tunnel TCAM region used for BFD over VXLAN. **Note** This command is supported only if the TP_SERVICES_PKG license is installed.
`sflow`	Configures the size of the sFlow TCAM region.
`span`	Configures the size of the SPAN TCAM region.
`svi`	Configures the size of the ingress SVI counters TCAM region.
`vacl`	Configures the size of the IPv4 VACL TCAM region.
`vpc-convergence`	Configures the size of the vPC convergence TCAM region.
`vqos`	Configures the size of the IPv4 VLAN QoS TCAM region.
`vqos-lite`	Configures the size of the IPv4 VLAN QoS lite TCAM region.
`tcam-size`	Specifies the TCAM size. The size must be a multiple of 256. If the size is more than 256, it must be a multiple of 512. For FHS, the range is 0 to 4096.

You can use the **no** form of this command to revert to the default TCAM region size.

Note
You can attach IPv4 user-defined fields (UDFs) to the racl, ifacl, and vacl TCAM regions using the **hardware access-list tcam region** {**racl** | **ifacl** | **vacl**} **qualify udf** *udf-names* command to configure IPv4 UDF-based SPAN or ERSPAN.You can attach IPv6 UDFs to the ing-l2-span-filter and ing-l3-span-filter TCAM regions using the **hardware access-list tcam region** {**ing-ifacl** | **ing-l2-span-filter** | **ing-l3-span-filter**} **qualify v6udf** *v6udf-names* commands to configure IPv6 UDF-based ERSPAN. For more information and configuration instructions, see the latest *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Step 3 **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Step 4 (Optional) **show hardware access-list tcam region**

Example:

```
switch#show hardware access-list tcam region
```

Displays the TCAM sizes that will be applicable on the next reload of the device.

Step 5 **hardware access-list tcam label vrf-nat**

Example:

```
switch#configure terminal
switch(config)#hardware access-list tcam label vrf-nat
```

Configures the ITD NAT with VRF.

Note
Beginning with Cisco NX-OS Release 10.3(1)F, this command is supported on Cisco Nexus 9300-GX switches.

Step 6 **reload**

Example:

```
switch#reload
```

Reloads the device.

Note
The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules.

The following example shows how to change the size of the n9k-arp-acl TCAM region on a Cisco Nexus NFE-enabled device:

```
switch#hardware access-list tcam region n9k-arp-acl 256
switch#copy running-config startup-config
switch#reload

Configuring storm-control-cpu:
switch#configure terminal
switch(config)#interface ethernet 1/10
switch(config-if)#storm-control-cpu arp rate 150
switch(config-if)#exit
switch#show access-list storm-control-cpu arp-stats interface ethernet 1/10
```

The following example shows how to change the size of the RACL TCAM region on a Cisco Nexus 9500 Series switch:

```
switchhardware access-list tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy running-config startup-config' and 'reload'
switch#copy running-config startup-config
switchreload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

This example shows how to display the TCAM region sizes to verify your changes:

```
switch#show hardware access-list tcam region
TCAM Region Sizes:

IPV4 PACL [ifacl] size =    512
IPV6 PACL [ipv6-ifacl] size =    0
MAC PACL [mac-ifacl] size =    0
IPV4 Port QoS [qos] size =    256
IPV6 Port QoS [ipv6-qos] size =    0
MAC Port QoS [mac-qos] size =    0
FEX IPV4 PACL [fex-ifacl] size =    0
FEX IPV6 PACL [fex-ipv6-ifacl] size =    0
FEX MAC PACL [fex-mac-ifacl] size =    0
FEX IPV4 Port QoS [fex-qos] size =    0
FEX IPV6 Port QoS [fex-ipv6-qos] size =    0
FEX MAC Port QoS [fex-mac-qos] size =    0
IPV4 VACL [vacl] size =    512
IPV6 VACL [ipv6-vacl] size =    0
MAC VACL [mac-vacl] size =    0
IPV4 VLAN QoS [vqos] size =    0
IPV6 VLAN QoS [ipv6-vqos] size =    0
MAC VLAN QoS [mac-vqos] size =    0
IPV4 RACL [racl] size =    512
IPV6 RACL [ipv6-racl] size =    0
IPV4 Port QoS Lite [qos-lite] size =    0
FEX IPV4 Port QoS Lite [fex-qos-lite] size =    0
IPV4 VLAN QoS Lite [vqos-lite] size =    0
IPV4 L3 QoS Lite [l3qos-lite] size =    0
Egress IPV4 QoS [e-qos] size =    0
Egress IPV6 QoS [e-ipv6-qos] size =    0
Egress MAC QoS [e-mac-qos] size =    0
Egress IPV4 VACL [vacl] size =    512
Egress IPV6 VACL [ipv6-vacl] size =    0
Egress MAC VACL [mac-vacl] size =    0
Egress IPV4 RACL [e-racl] size =    256
Egress IPV6 RACL [e-ipv6-racl] size =    0
Egress IPV4 QoS Lite [e-qos-lite] size =    0
IPV4 L3 QoS [l3qos] size =    0
IPV6 L3 QoS [ipv6-l3qos] size =    0
MAC L3 QoS [mac-l3qos] size =    0
Ingress System size =  256
Egress System size =  256
SPAN [span] size =  256
Ingress COPP [copp] size =  256
Ingress Flow Counters [flow] size =    0
Egress Flow Counters [e-flow] size = 0
Ingress SVI Counters [svi] size =    0
Redirect [redirect] size =  512
NS IPV4 Port QoS [ns-qos] size =  256
NS IPV6 Port QoS [ns-ipv6-qos] size =    0
NS MAC Port QoS [ns-mac-qos] size =    0
NS IPV4 VLAN QoS [ns-vqos] size =  256
NS IPV6 VLAN QoS [ns-ipv6-vqos] size =    0
NS MAC VLAN QoS [ns-mac-vqos] size =    0
NS IPV4 L3 QoS [ns-l3qos] size =  256
NS IPV6 L3 QoS [ns-ipv6-l3qos] size =    0
NS MAC L3 QoS [ns-mac-l3qos] size =    0
VPC Convergence [vpc-convergence] size =  256
IPSG SMAC-IP bind table [ipsg] size =    0
Ingress ARP-Ether ACL [arp-ether] size =    0
ranger+ IPV4 QoS Lite [rp-qos-lite] size =    0
ranger+ IPV4 QoS [rp-qos] size =  256
ranger+ IPV6 QoS [rp-ipv6-qos] size =  256
ranger+ MAC QoS [rp-mac-qos] size =  256
NAT ACL[nat] size =    0
```

```
Mpls ACL size =    0
Ingress IPv4 N3K QoS size =    0
Ingress IPv6 N3K QoS size =    0
MOD RSVD size =    0
sFlow ACL [sflow] size =    0
mcast bidir ACL size =    0
Openflow size =    0
```

```
switch#show hardware access-list tcam region
TCAM Region Sizes:

IPV4 PACL [ifacl] size =    0
IPV6 PACL [ipv6-ifacl] size =    0
MAC PACL [mac-ifacl] size =    0
IPV4 Port QoS [qos] size =    0
IPV6 Port QoS [ipv6-qos] size =    0
MAC Port QoS [mac-qos] size =    0
FEX IPV4 PACL [fex-ifacl] size =    0
FEX IPV6 PACL [fex-ipv6-ifacl] size =    0
FEX MAC PACL [fex-mac-ifacl] size =    0
FEX IPV4 Port QoS [fex-qos] size =    0
FEX IPV6 Port QoS [fex-ipv6-qos] size =    0
FEX MAC Port QoS [fex-mac-qos] size =    0
IPV4 VACL [vacl] size =    0
IPV6 VACL [ipv6-vacl] size =    0
MAC VACL [mac-vacl] size =    0
IPV4 VLAN QoS [vqos] size =    0
IPV6 VLAN QoS [ipv6-vqos] size =    0
MAC VLAN QoS [mac-vqos] size =    0
IPV4 RACL [racl] size = 1536
IPV6 RACL [ipv6-racl] size =    0
IPV4 Port QoS Lite [qos-lite] size =    0
FEX IPV4 Port QoS Lite [fex-qos-lite] size =    0
IPV4 VLAN QoS Lite [vqos-lite] size =    0
IPV4 L3 QoS Lite [l3qos-lite] size =    0
Egress IPV4 QoS [e-qos] size =    0
Egress IPV6 QoS [e-ipv6-qos] size =    0
Egress MAC QoS [e-mac-qos] size =    0
Egress IPV4 VACL [vacl] size =    0
Egress IPV6 VACL [ipv6-vacl] size =    0
Egress MAC VACL [mac-vacl] size =    0
Egress IPV4 RACL [e-racl] size =  768
Egress IPV6 RACL [e-ipv6-racl] size =    0
Egress IPV4 QoS Lite [e-qos-lite] size =    0
IPV4 L3 QoS [l3qos] size =  256
IPV6 L3 QoS [ipv6-l3qos] size =    0
MAC L3 QoS [mac-l3qos] size =    0
Ingress System size =  256
Egress System size =  256
SPAN [span] size =  256
Ingress COPP [copp] size =  256
Ingress Flow Counters [flow] size =    0
Egress Flow Counters [e-flow] size =    0
Ingress SVI Counters [svi] size =    0
Redirect [redirect] size =  256
NS IPV4 Port QoS [ns-qos] size =  256
NS IPV6 Port QoS [ns-ipv6-qos] size =    0
NS MAC Port QoS [ns-mac-qos] size =    0
NS IPV4 VLAN QoS [ns-vqos] size =  256
NS IPV6 VLAN QoS [ns-ipv6-vqos] size =    0
NS MAC VLAN QoS [ns-mac-vqos] size =    0
NS IPV4 L3 QoS [ns-l3qos] size =  256
NS IPV6 L3 QoS [ns-ipv6-l3qos] size =    0
NS MAC L3 QoS [ns-mac-l3qos] size =    0
VPC Convergence [vpc-convergence] size =  512
```

```
IPSG SMAC-IP bind table [ipsg] size =    0
Ingress ARP-Ether ACL [arp-ether] size =    0
```

This example shows how to revert to the default RACL TCAM region size:

```
switch#no hardware access-list tcam region racl
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy running-config startup-config' and 'reload'
switch#exit
switch#copy running-config startup-config
switch#reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

Configure ACL TCAM region sizes using templates

You can use create and apply custom templates to configure ACL TCAM region sizes.

For all Cisco Nexus 9200, 9300, and 9500 Series switches, you can use this procedure or the Configuring ACL TCAM Region Sizes procedure to configure ACL TCAM region sizes. However, NFE2-enabled devices (such as the X9432C-S 100G line card and the C9508-FM-S fabric module) do not support the **hardware access-list tcam region** command and must use a template to configure the ACL TCAM region size.

Note

- Once you apply a TCAM template, the **hardware access-list tcam region** command will not work. You must uncommit the template in order to use the command.
- For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.
- The TCAM profile template is not supported on the C9508-FM-S fabric module.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **hardware profile tcam resource template** *template-name* **ref-template** {**nfe** | **nfe2** | {**l2-l3** | **l3**}}

Example:

```
switch(config)#hardware profile tcam resource template SR_MPLS_CARVE ref-template nfe2
switch(config-tcam-temp)#
```

Creates a template for configuring ACL TCAM region sizes.

nfe —The default TCAM template for Network Forwarding Engine (NFE)-enabled Cisco Nexus 9300 and 9500 Series.

nfe2 —The default TCAM template for NFE2-enabled Cisco Nexus 9500 Series devices.

l2-l3 —The default TCAM template for Layer 2 and Layer 3 configurations.

l3 —The default TCAM template for Layer 3 configurations.

Note
This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

Step 3 (Optional) *region tcam-size*

Example:

```
switch(config-tcam-temp)#mpls 256
```

Adds any desired TCAM regions and their sizes to the template. Enter this command for each region you want to add to the template. For the list of available regions, see Configuring ACL TCAM Region Sizes.

Step 4 **exit**

Example:

```
switch(config-tcam-temp)#exit
switch(config#)
```

Exits the TCAM template configuration mode.

Step 5 [**no**] **hardware profile tcam resource service-template** *template-name*

Example:

```
switch(config)#hardware profile tcam resource service-template SR_MPLS_CARVE
```

Applies the custom template to all line cards and fabric modules.

Note
This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

Step 6 (Optional) **show hardware access-list tcam template** {**all** | **nfe** | **nfe2** | **l2-l3** | **l3** | *template-name*}

Example:

```
switch#show hardware access-list tcam template SR_MPLS_CARVE
```

Displays the configuration for all TCAM templates or for a specific template.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Step 8 **reload**

Example:

```
switch#reload
```

Reloads the device.

Note

The configuration is effective only after you enter **copy running-config startup-config** + **reload** .

Configure TCAM carving

The default TCAM region configuration varies by platform and does not accommodate all TCAM regions. To enable any desired regions, you must decrease the TCAM size of one region and then increase the TCAM size for the desired region.

This configuration is not supported on Cisco N9300 Smart switches.

Note For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide* .

Note Beginning with Cisco NX-OS Release 10.3(1)F , the following TCAM limitation applies on Cisco Nexus 9800 platform switches:

- TCAM carving is not supported. However you can view the currently allocated TCAM for each individual feature. To view the currently allocated TCAM, use the **show hardware access-list resource utilization** command.
- Central TCAM is supported. However it is shared among both ingress and egress ACLs.

Table 58: Default TCAM region configuration (ingress) - for Cisco Nexus 9500 series switches

Region Name	Size	Width	Total Size
IPv4 RACL	1536	1	1536
IPv4 Layer 3 QoS	256	2	512
SPAN	256	1	256
CoPP	256	2	512
System	256	2	512
Redirect	256	1	256
vPC convergence	512	1	512

Table 59: Default TCAM region configuration (egress) - for Cisco Nexus 9500 series switches

Region Name	Size	Width	Total Size
IPv4 RACL	768	1	768
System	256	1	256

Region Name	Size	Width	Total Size
			1K

Table 60: Default TCAM size - for Cisco Nexus 9504 and 9508 platform switches

Region	Size
MAC PACL [mac-ifacl]	1952
IPV6 Port QoS [ipv6-qos]	256
PV6 L3 QoS [ipv6-l3qos]	256
SPAN [span]	96
Ingress CoPP [copp]	128
Redirect IPv4	2048
Redirect IPv6	2048

Table 61: Default TCAM region configuration (ingress) - for Cisco Nexus 9300-FX series switches

Region Name	Size	Width	Total Size
IPv4 RACL	2304	1	2304
Layer 2 QoS	256	1	256
Layer 3/VLAN QoS	512	1	512
System	512	1	512
Layer 2 SPAN filter	256	1	256
Layer 3 SPAN filter	256	1	256
SPAN	512	1	512
NetFlow/Analytics filter	512	1	512
			5K

Table 62: Default TCAM region configuration (egress) - for Cisco Nexus 9300-FX series switches

Region Name	Size	Width	Total Size
IPv4 RACL	1792	1	1792
System	256	1	256
			2K

Table 63: Default TCAM region configuration (ingress) - for Cisco Nexus 9300-EX series switches

Region Name	Size	Width	Total Size
IPv4 RACL	1792	1	1792
Layer 2 QoS	256	1	256
Layer 3/VLAN QoS	512	1	512
System	512	1	512
Layer 2 SPAN ACL	256	1	256
Layer 3/VLAN SPAN ACL	256	1	256
SPAN	512	1	512
			4K

Table 64: Default TCAM region configuration (egress) - for Cisco Nexus 9300-EX series switches

Region Name	Size	Width	Total Size
IPv4 RACL	1792	1	1792
System	256	1	256
			2K

Table 65: Default TCAM region configuration (ingress) - for Cisco Nexus 9300 series switches

Region Name	Size	Width	Total Size
IPv4 port ACL	512	1	512
IPv4 port QoS	256	2	512
IPv4 VACL	512	1	512
IPv4 RACL	512	1	512
SPAN	256	1	256
CoPP	256	2	512
IPv4 port QoS for ACI leaf line card	256	1	256
IPv4 VLAN QoS for ACI leaf line card	256	1	256
IPv4 Layer 3 QoS for ACI leaf line card	256	1	256
System	256	2	512
Redirect	512	1	512
vPC convergence	256	1	256

Region Name	Size	Width	Total Size
			4K

Table 66: Default TCAM region configuration (egress) - for Cisco Nexus 9300 series switches

Region Name	Size	Width	Total Size
IPv4 VACL	512	1	512
IPv4 RACL	256	1	256
System	256	1	256
			1K

Table 67: Default TCAM region configuration (ingress) - for layer 2-to-Layer 3 configurations on Cisco Nexus 9200 series switches

Region Name	Size	Width	Total Size
Ingress NAT	0	1	0
Ingress port ACL	256	1	256
Ingress VACL	256	1	256
Ingress RACL	1536	1	1536
Ingress Layer 2 QoS	256	1	256
Ingress Layer 3 VLAN QoS	256	1	256
Ingress supervisor	512	1	512
Ingress Layer 2 ACL SPAN	256	1	256
Ingress Layer 3 ACL SPAN	256	1	256
Port-based SPAN	512	1	512
			4096

Table 68: Default TCAM region configuration (egress) - for layer 2-to-Layer 3 configurations on Cisco Nexus 9200 series switches

Region Name	Size	Width	Total Size
Egress VACL	256	1	256
Egress RACL	1536	1	1536
Egress supervisor	256	1	256
			2048

Table 69: Default TCAM region configuration (ingress) - for layer 3 configurations on Cisco Nexus 9200 series switches

Region Name	Size	Width	Total Size
Ingress NAT	0	1	0
Ingress port ACL	0	1	0
Ingress VACL	0	1	0
Ingress RACL	1792	1	1792
Ingress Layer 2 QoS	256	1	256
Ingress Layer 3 VLAN QoS	512	1	512
Ingress supervisor	512	1	512
Ingress Layer 2 ACL SPAN	256	1	256
Ingress Layer 3 ACL SPAN	256	1	256
Port-based SPAN	512	1	512
			4096

Table 70: Default TCAM region configuration (egress) - for layer 3 configurations on Cisco Nexus 9200 series switches

Region Name	Size	Width	Total Size
Egress VACL	0	1	0
Egress RACL	1792	1	1792
Egress supervisor	256	1	256
			2048

The following example sets the IPv6 RACL TCAM size to 256 on a Cisco Nexus 9500 Series switch. An IPv6 RACL of size 256 takes 512 entries because IPv6 is double wide.



Note Follow a similar procedure to modify the TCAM settings for a different region or to modify the TCAM settings on a different device.

To set the size of the ingress IPv6 RACL TCAM region on a Cisco Nexus 9500 Series switch, perform one of two options.

Option #1

Reduce the ingress IPv4 RACL by 1024 entries (1536 - 1024 = 512) and add an ingress IPv6 RACL with 512 entries—This option is preferred.

```
switch#hardware access-list tcam region racl 512
Warning: Please reload the linecard for the configuration to take effect
```

```
switch#hardware access-list tcam region ipv6-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

Table 71: Updated TCAM region configuration after reducing the IPv4 RACL (ingress)

Region Name	Size	Width	Total Size
IPv4 RACL	1024	1	1024
IPv6 RACL	256	2	1024[2]
IPv4 Layer 3 QoS	256	2	512
SPAN	256	1	256
CoPP	256	2	512
System	256	2	512
Redirect	256	1	256
vPC convergence	512	1	512
			4K

[2] 2 x 512 entry slices are allocated due to the non-availability of 256 entry slices.

Option #2

Remove IPv4 Layer 3 QoS by reducing its size to 0 and add an ingress IPv6 RACL—This option is available if you are not using IPv4 Layer 3 QoS.

```
switch#hardware access-list tcam region l3qos 0
Warning: Please reload the linecard for the configuration to take effect
switch#hardware access-list tcam region ipv6-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

Table 72: Updated TCAM region configuration after removing layer 3 QoS (ingress)

Region Name	Size	Width	Total Size
IPv4 RACL	1536	1	1536
IPv6 RACL	256	2	512
IPv4 Layer 3 QoS	0	2	0
SPAN	256	1	256
CoPP	256	2	512
System	256	2	512
Redirect	256	1	256
vPC convergence	512	1	512
			4K

To enable an egress IPv6 RACL of size 256, reduce the egress IPv4 RACL to 256 and add the egress IPv6 RACL:

```
switch#hardware access-list tcam region e-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

```
switch#hardware access-list tcam region e-ipv6-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

Table 73: Default TCAM region configuration after reducing the IPv4 RACL (egress)

Region Name	Size	Width	Total Size
IPv4 RACL	256	1	256
IPv6 RACL	256	2	512
System	256	1	256
			1K

Table 74: Default TCAM size - for Cisco Nexus 9800 platform switches

Feature name	Size (Unidimensional)
Ingress RACLv4	9216
Ingress QoSv4	
Ingress SPAN filter v4	
Egress RACLv4	
Ingress SUP	
Ingress RACLv6	4608
Ingress QoSv6	
Ingress SPAN filter v6	
Egress RACL v6	

Note Each IPv6 ACL is limited to 1,000 ACEs. This applies to all IPv6 ACLs (RACL, QoS or SPAN filter). No such limitation applies for IPv4 ACL.

After you adjust the TCAM region sizes, enter the **show hardware access-list tcam region** command to display the TCAM sizes that will be applicable on the next reload of the device.

Attention To keep all modules synchronized, you must reload all line card modules or enter **copy running-config startup-config** + **reload** to reload the device. Multiple TCAM region configurations require only a single reload. You can wait until you complete all of your TCAM region configurations before you reload the device.

Depending on the configuration, you might exceed the TCAM size or run out of slices.

If you exceed the 4K ingress limit for all TCAM regions when you configure a TCAM region, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Ingress TCAM space. Please
 re-configure.
```

If you exceed the number of slices, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Ingress TCAM slices.
Please re-configure.
```

If you exceed the 1K egress limit for all TCAM regions when you configure a TCAM region, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Egress TCAM space. Please
 re-configure.
```

If TCAM for a particular feature is not configured and you try to apply a feature that requires TCAM carving, the following message appears:

```
ERROR: Module x returned status: TCAM region is not configured. Please configure TCAM
region and retry the command.
```

Note The default redirect TCAM region size of 256 might not be sufficient if you are running many BFD or DHCP relay sessions. To accommodate more BFD or DHCP relay sessions, you might need to increase the TCAM size to 512 or greater.

Note "e-racl" tcam region size can be maximum of 16K when we have at least one "N9K-X9624D-R2" line card on a N9K-C9508 (Fretta) system.

Related Topics

Configure ACL TCAM region sizes, on page 344

Configure UDF-Based port ACLs

You can configure UDF-based port ACLs for Cisco Nexus 9200, 9300, and 9300-EX Series switches. This feature enables the device to match on user-defined fields (UDFs) and to apply the matching packets to an IPv4 port ACL.

You can configure UDF-based port IPv6 ACLs for Cisco Nexus 9300-EX switches. This feature enables the device to match on the new UDFs and to apply the matching packets to an IPv6 port ACL.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **udf** *udf-name offset-base offset length*

Example:

```
switch(config)#udf pktoff10 packet-start 10 2
```

Example:

```
switch(config)#udf pktoff10 header outer l3 20 2
```

Defines the UDF as follows:

- *udf-name* —Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.
- *offset-base* —Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: {**packet-start** | **header** {**outer** | **inner** {**l3** | **l4**}}}.
- *offset* —Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0.
- *length* —Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.

You can define multiple UDFs, but Cisco recommends defining only required UDFs.

Step 3 **hardware access-list tcam region ing-ifacl qualify** {**udf** *udf-name*} | **v6udf** *v6udf-name* }

Example:

```
switch#hardware access-list tcam region ing-ifacl qualify udf pktoff10
```

Attaches the UDFs to the ing-ifacl TCAM region, which applies to IPv4 or IPv6 port ACLs.

Note
This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

The number of UDFs that can be attached to a TCAM region varies by platform. You can attach up to 2 UDFs for Cisco Nexus 9200 switches, up to 8 UDFs for Cisco Nexus 9300 switches, and up to 18 UDFs for IPv4 port ACLs or 7 UDFs for IPv6 port ACLs for Cisco Nexus 9300-EX switches.

Note
When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see Configuring ACL TCAM Region Sizes.

Note
The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.

Step 4 Required: **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Step 5 Required: **reload**

Example:

```
switch#reload
```

Reloads the device.

Note
Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload** .

Step 6 **ip access-list** *udf-acl*

Example:

```
switch(config)#ip access-list udfacl
switch(config-acl)#
```

Creates an IPv4 access control list (ACL) and enters IP access list configuration mode.

Step 7 Enter one of the following commands:

- **permit udf** *udf-name value mask*
- **permit ip** *source destination* **udf** *udf-name value mask*

Example:

```
switch(config-acl)#permit udf pktoff10 0x1234 0xffff
```

Example:

```
switch(config-acl)#permit ip any any udf pktoff10 0x1234 0xffff
```

Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). The range for the *value* and *mask* arguments is from 0x0 to 0xffff.

A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Applying an IP ACL as a router ACL

You can apply an IPv4 or IPv6 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces
- VLAN interfaces
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

Note Egress router ACLs are not supported on Cisco Nexus 9300 Series switch uplink ports.

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot*/*port*[**.** *number*]
- **interface port-channel** *channel-number*
- **interface vlan** *vlan-id*
- **interface mgmt** *port*

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

```
switch(config)#interface ethernet 2/3.1
switch(config-if)#
```

Enters configuration mode for the interface type that you specified.

Step 3 (Optional) **encapsulation dot1q 21**

Example:

```
switch(config-if)#encapsulation dot1q 21
switch(config-if)#
```

Note
This command is required only for Layer 3 subinterfaces.

Step 4 Enter one of the following commands:

- **ip access-group** *access-list* {**in** | **out**}
- **ipv6 traffic-filter** *access-list* {**in** | **out**}

Example:

```
switch(config-if)#ip access-group acl1 in
```

Applies an IPv4 or IPv6 ACL to the Layer 3 interface and subinterfaces for traffic flowing in the direction specified. You can apply one router ACL per direction.

Step 5 **ip access-list match-local-traffic**

Example:

```
switch(config-if)#ip access-list match-local-traffic
```

Lists the matching traffic which is generated locally. It does not affect transit traffic through the switch.

Step 6 (Optional) **show running-config aclmgr**

Example:

```
switch#show running-config aclmgr
```

Displays the ACL configuration.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Create an IP ACL, on page 337

Applying an IP ACL as a port ACL

You can apply an IPv4 or IPv6 ACL to a Layer 2 interface, which can be a physical port or a port channel. ACLs applied to these interface types are considered port ACLs.

Note If the interface is configured with the **mac packet-classify** command, you cannot apply an IP port ACL to the interface until you remove the **mac packet-classify** command from the interface configuration.

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot/port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

Enters configuration mode for the interface type that you specified.

Step 3 Enter one of the following commands:

- **ip port access-group** *access-list* **in**
- **ipv6 port traffic-filter** *access-list* **in**

Example:

```
switch(config-if)#ip port access-group acl-l2-marketing-group in
```

Applies an IPv4 or IPv6 ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.

Step 4 (Optional) **show running-config aclmgr**

Example:

```
switch#show running-config aclmgr
```

Displays the ACL configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Create an IP ACL, on page 337
Enable or disable MAC packet classification, on page 402

Applying an IP ACL as a VACL

You can apply an IP ACL as a VACL.

Related Topics

Configuring VACLs, on page 410

Apply an IP ACL rule prioritization over SUP rule

Beginning with Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is supported for IP or IPv6 ACL which would increase the priority of ACL rule to 0 (highest) over any other SUP ACL rule that also matches on the same criteria.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **ip access-list** *name*
- **ipv6 access-list** *name*

Example:

```
switch(config)#ip access-list acl-01
switch(config-acl)#
```

Creates the IP or IPv6 ACL and enters the ACL configuration mode. The *name* argument can be up to 64 characters.

Step 3 [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*} **all**

Example:

For IP

```
switch(config-acl)#permit ip 192.168.2.0/24 any all
```

```
switch(config-ipv6-acl)#10 permit ipv6 2001:db8::1 2001:db8::2 2001:db8::3 2001:db8::4 all
```

Creates a rule in the IP or IPv6 ACL with an **all** keyword to prioritize the ACL rule over the SUP rule.

Step 4 Enter one of the following commands:

- **interface ethernet** *slot*/*port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

Enters configuration mode for the interface type that you specified.

Step 5 Enter one of the following commands:

- **ip port access-group** *access-list* **in**
- **ipv6 port traffic-filter** *access-list* **in**

Example:

For IP

```
switch(config-if)#ip port access-group acl-01 in
```

Applies an IPv4 or IPv6 ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.

Step 6 (Optional) **show running-config aclmgr**

Example:

```
switch#show running-config aclmgr
```

Displays the ACL configuration.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure IPv4 ACL logging

To configure the ACL logging process, you first create the access list, then enable filtering of traffic on an interface using the specified ACL, and finally configure the ACL logging process parameters.

Procedure

Step 1 Enter global configuration mode.

configure terminal

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Create and configure the IPv4 ACL.

a) Create an IPv4 ACL and enters IP ACL configuration mode.

ip access-list *name*

Example:

```
switch(config)#ip access-list logging-test
switch(config-acl)#
```

The *name* argument can be up to 64 characters.

b) Add permit or deny rules and enable logging.

{**permit** | **deny**} **ip** *source-address destination-address* **log**

Example:

```
switch(config-acl)#permit ip any 10.30.30.0/24 log
```

To enable the system to generate an informational logging message about each packet that matches the rule, you must include the **log** keyword.

The *source-address* and *destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, or **any** to designate any address.

c) Exit ACL configuration mode.

exit

Example:

```
switch(config-acl)#exit
switch(config)#
```

Step 3 Assign the ACL to an interface.

a) Enter interface config mode.

interface ethernet *slot/port*

Example:

```
switch(config)#interface ethernet 1/1
switch(config-if)#
```

b) Apply the ACL.

ip access-group *name* **in**

Example:

```
switch(config-if)#ip access-group logging-test in
```

Enables the filtering of IPv4 traffic on an interface using the specified ACL. You can apply an ACL to inbound traffic.

c) Exit interface configuration mode.

exit

Example:

```
switch(config-if)#exit
switch(config)#
```

Step 4 Set logging parameters.

a) Configure log-update interval.

logging ip access-list cache interval *interval*

Example:

```
switch(config)#logging ip access-list cache interval 490
```

Configures the log-update interval (in seconds) for the ACL logging process. The default value is 300 seconds. The range is from 5 to 86400 seconds.

b) Define max number of flows monitored.

logging ip access-list cache entries *number-of-flows*

Example:

```
switch(config)#logging ip access-list cache entries 8001
```

Specifies the maximum number of flows to be monitored by the ACL logging process. The default value is 8000. The range of values supported is from 0 to 1048576.

c) Set cache threshold for syslog alerts.

logging ip access-list cache threshold *threshold*

Example:

```
switch(config)#logging ip access-list cache threshold 490
```

If the specified number of packets is logged before the expiry of the alert interval, the system generates a syslog message.

Step 5 (Optional) Enable detailed logging output.

logging ip access-list detailed

Example:

```
switch(config)#logging ip access-list detailed
```

Enables the following information to be displayed in the output of the **show logging ip access-list cache** command: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface.

Step 6 (Optional) Include additional details in logs.

a) Add security group tag information.

logging ip access-list include sgt

Example:

```
switch(config)#logging ip access-list include sgt
```

Enables the following SGACL information to be displayed in the output of the **show logging ip access-list cache** command: the security group tag (SGT), destination group tag (DGT), source MAC (SMAC) and destination MAC (DMAC).

b) Add MAC address details.

logging ip access-list include mac

Example:

```
switch(config)#logging ip access-list include mac
```

Enables to include MAC address as part of the ACLLOG entry to be displayed in the output of the **show logging ip access-list cache** command: source MAC (SMAC) and destination MAC (DMAC).

This command can be configured along with detailed option (**logging ip access-list detailed**), or without detailed option.

Step 7 (Optional) Configure hardware rate limiting for ACL logging.

hardware rate-limiter access-list-log *packets*

Example:

```
switch(config)#hardware rate-limiter access-list-log 200
```

Configures rate limits in packets per second for packets copied to the supervisor module for ACL logging. The range is from 0 to 30000.

Step 8 Set the minimum severity level for logging matches

acllog match-log-level *severity-level*

Example:

```
switch(config)#acllog match-log-level 5
```

Specifies the minimum severity level to log ACL matches. The default is 6 (informational). The range is from 0 (emergency) to 7 (debugging).

Step 9 Verify and review logged traffic.

a) (Optional) **show logging ip access-list cache** [**detail**]

Example:

```
switch#show logging ip access-list cache
```

Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, source interfaces. No other information of active flows will be displayed specifically all the unsupported options.

If you entered the **logging ip access-list detailed** command, the output also includes the following information: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface.

Note
The output format of **show logging ip access-list cache** [**detail**] will be based on the chosen optional configurations:

- If you configure two or all optional configurations, the **include sgt** option takes the precedence over the other optional configurations.
- For example, if the **logging ip access-list detailed** command is used along with the **logging ip access-list include sgt** command or vice-versa, the configured **include sgt** display format will be used for printing the flow information.

Configure ACLs using HTTP methods to redirect requests

You can configure ACLs to intercept and redirect specific HTTP methods to a server that is connected to a specific port.

The following HTTP methods can be redirected:

- connect
- delete
- get
- head
- post
- put
- trace

Before you begin

Enable the double-wide TCAM for the IFACL region using the **hardware access-list tcam region ifacl 512 double-wide** command. This command applies to the global configuration. Reload the switch for this configuration to take into effect.

Note This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **ip access-list** *name*

Example:

```
switch(config)#ip access-list acl-01
switch(config-acl)#
```

Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.

Step 3 [*sequence-number*] **permit** *protocol source destination* **http-method** *method* [**tcp-option-length** *length*] [**redirect** *interface*]

Example:

```
switch(config-acl)#permit tcp 192.0.2.1/32 any http-method get
```

Configures the ACL to redirect specific HTTP methods to a server.

The following HTTP methods are supported:

- connect—Matches HTTP packets with the CONNECT method [0x434f4e4e]
- delete—Matches HTTP packets with the DELETE method [0x44454c45]
- get—Matches HTTP packets with the GET method [0x47455420]
- head—Matches HTTP packets with the HEAD method [0x48454144]
- post—Matches HTTP packets with the POST method [0x504f5354]
- put—Matches HTTP packets with the PUT method [0x50555420]
- trace—Matches HTTP packets with the TRACE method [0x54524143]

The **tcp-option-length** option specifies the length of the TCP options header in the packets. You can configure up to four TCP option lengths (in multiples of four bytes) in the access control entries (ACEs). The *length* range is from 0 to 40. If you do not configure this option, the length is specified as 0, and only packets without the TCP options header can match the ACE. This option allows the HTTP method to be matched even on packets that have a variable-length TCP options header.

The **redirect** option redirects an HTTP method to a server that is connected to a specific port. The HTTP redirect feature does not work on Layer 3 ports.

Step 4 (Optional) **show ip access-lists** *name*

Example:

```
switch#show ip access-lists acl-01
```

Displays the IP ACL configuration.

Step 5 (Optional) **show running-config interface** *interface slot/port*

Example:

```
switch#show running-config interface ethernet 2/2
```

Displays the interface configuration.

The following example specifies a length for the TCP options header in the packets and redirects the post HTTP method to a server that is connected to port channel 4001:

```
switch(config)#ip access-list http-redirect-acl
switch(config-acl)#10 permit tcp any any http-method get tcp-option-length 4 redirect
port-channel4001
switch(config-acl)#20 permit tcp any any http-method post redirect port-channel4001
switch(config-acl)#statistics per-entry
switch(config)#interface Ethernet 1/33
switch(config-if)#ip port access-group http-redirect-acl in
```

Configure an ACL for IPv6 extension headers

This procedure applies only to the following devices:

- Cisco Nexus 9504 and 9508 modular chassis with these line cards: N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, and N9K-X96136YC-R
- Cisco Nexus 3600 Platform Switches (N3K-C36180YC-R and N3K-C3636C-R)

Beginning with Cisco NX-OS Release 9.3(7), if you configure an IPv6 ACL on the devices listed here, you must include a new rule for the disposition of IPv6 packets that include extension headers. For more information about IPv6 extension headers, see "Simplified IPv6 Packet Header" in NX-OS Release 9.3(x) or later of the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

Note The permit or deny rule that you choose in this procedure is applied to any IPv6 packet with at least one extension header regardless of any other ACL rule that matches the packet's other fields.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **ipv6 access-list** *name*

Example:

```
switch(config)#ipv6 access-list acl-01
switch(config-acl)#
```

Creates the IPv6 ACL and enters ACL configuration mode.

Step 3 **extension-header** {**permit-all** | **deny-all**}

Example:

```
switch(config-acl)#extension-header permit-all
switch(config-acl)#
```

Choose the desired action for matched packets:

- **permit-all** — Any IPv6 packet with at least one extension header is permitted.
- **deny-all** — Any IPv6 packet with at least one extension header is dropped.

IP ACL configuration verification commands

Use this reference to verify the IP ACL configuration.

Table 75: IP ACL configuration verification commands

Command	Purpose
show hardware access-list tcam region	Displays the TCAM sizes that will be applicable on the next reload of the device.
show hardware access-list tcam template {**all** | **nfe** | **nfe2** | **l2-l3** | **l3** | *template-name*}	Displays the configuration for all TCAM templates or for a specific template. **nfe** —The default TCAM template for Network Forwarding Engine (NFE)-enabled Cisco Nexus 9300 and 9500 Series devices. **nfe2** —The default TCAM template for NFE2-enabled Cisco Nexus 9500 devices. **l2-l3** —The default TCAM template for Layer 2 and Layer 3 configurations. **l3** —The default TCAM template for Layer 3 configurations.
show ip access-lists	Displays the IPv4 ACL configuration.

Command	Purpose
show ipv6 access-lists	Displays the IPv6 ACL configuration.
show logging ip access-list cache [**detail**]	Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, and source interfaces. No other information of active flows will be displayed specifically all the unsupported options. If you entered the **logging ip access-list detailed** command, the output also includes the following information: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface.
show logging ip access-list status	Displays the deny maximum flow count, the current effective log interval, and the current effective threshold value.
show running-config acllog	Displays the ACL log running configuration.
show running-config aclmgr [**all**]	Displays the ACL running configuration, including the IP ACL configuration and the interfaces to which IP ACLs are applied. **Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config acllog	Displays the ACL log startup configuration.

Command	Purpose
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

show hardware access-list interface *ethernet X/Y* **input entries detail**	Displays the hardware ACL interface input entries' detail.
	Note
	On platforms other than 9500-R, even when the entry is expanded, the display shows the range as x y.
	Sample output for 9500-R:
	`permit tcp 192.0.2.1/24 eq 10006 192.0.2.1/24 eq 0x4e24/fffe [0]`
	Sample output for 9300-FX3S:
	`permit tcp 192.0.2.1/24 eq 10006 192.0.2.1/24 range 20004 20005 routeable 0x1 [0]`

Monitoring and clearing IP ACL statistics

Use this reference to review monitoring and clearing IP ACL statistics.

To monitor or clear IP ACL statistics, use one of the commands in this table

Table 76: To monitor or clear IP ACL statistics, use one of the commands in this table

Command	Purpose
show ip access-lists	Displays the IPv4 ACL configuration. If the IPv4 ACL includes the **statistics per-entry** command, the **show ip access-lists** command output includes the number of packets that have matched each rule.
show ipv6 access-lists	Displays IPv6 ACL configuration. If the IPv6 ACL includes the **statistics per-entry** command, then the **show ipv6 access-lists** command output includes the number of packets that have matched each rule.
clear ip access-list counters	Clears statistics for all IPv4 ACLs or for a specific IPv4 ACL.

Command	Purpose
clear ipv6 access-list counters	Clears statistics for all IPv6 ACLs or for a specific IPv6 ACL.

Configuration examples for IP ACLs

These examples show common IP ACL configurations.

IPv4 port ACL example

The following example shows how to create an IPv4 ACL named acl-01 and apply it as a port ACL to Ethernet interface 2/1, which is a Layer 2 interface:

```
ip access-list acl-01
  permit ip 192.168.2.0/24 any
interface ethernet 2/1
  ip port access-group acl-01 in
```

IPv6 router ACL example

The following example shows how to create an IPv6 ACL named acl-120 and apply it as a router ACL to Ethernet interface 2/3, which is a Layer 3 interface:

```
ipv6 access-list acl-120
  permit tcp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit tcp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
interface ethernet 2/3
  ipv6 traffic-filter acl-120 in
```

VTY ACL example

The following example shows how to create a VTY ACL named single-source and apply it on input IP traffic over the VTY line. This ACL allows all TCP traffic through and drops all other IP traffic:

```
ip access-list single-source
  permit tcp 192.168.7.5/24 any
  exit
  line vty
ip access-class single-source in
show ip access-lists
```

IPv4 ACL logging example

The following example shows how to configure IPv4 ACL logging:

```
switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)#ip access-list logging-test
switch(config-acl)#permit ip any 2001:DB8:1::1/64 log
switch(config-acl)#exit
switch(config)#interface ethernet 1/1
switch(config-if)#ip access-group logging-test in
switch(config-if)#exit
switch(config)#logging ip access-list cache interval 400
switch(config)#logging ip access-list cache entries 100
switch(config)#logging ip access-list cache threshold 900
```

```
switch(config)#hardware rate-limiter access-list-log 200
switch(config)#acllog match-log-level 5
```

UDF-based port ACL example

The following example shows how to configure a UDF-based port ACL:

```
switch#configure terminal
switch#hardware access-list tcam region ing-ifacl 256
switch(config)#udf pktoff10 packet-start 10 2
switch(config)#udf pktoff20 packet-start 10 1
switch#hardware access-list tcam region ing-ifacl qualify udf pktoff10 pktoff20
switch#configure terminal
switch(config)#ip access-list udfacl
switch(config-acl)#statistics per-entry
switch(config-acl)#10 permit ip any any udf pktoff10 0x1234 0xffff
switch#configure terminal
switch(config)#interface Ethernet1/1
switch(config-if)#ip port access-group udfacl in
switch(config-if)#switchport
switch(config-if)#no shutdown
```

Route-tag default-route example

The following example shows the configuration of the **route-tag default-route**:

```
switch(config)#ip access-list global-acl
switch(config-acl)#10 permit ip any any route-tag default-route
switch(config-acl)#exit
switch(config)#class-map type qos global
switch(config-cmap-qos)#match access-group name global-acl
switch(config)#class-map type qos domestic
switch(config-cmap-qos)#match access-group name domestic-acl
switch(config)#policy-map type qos pmap
switch(config-pmap)#class global
switch(config-pmap-c)#police cir 100 mbps bc 200 ms conform transmit violate drop
switch(config-pmap)#class domestic
switch(config-pmap-c)#police cir 200 mbps bc 200 ms conform transmit violate drop
switch(config)#interface ethernet1/12
switch(config-if)#service-policy type qos input pmap
switch#show running-config ipqos
!Running configuration last done at: Tue Jun 13 10:08:38 2023
!Time: Tue Jun 13 10:10:05 2023
version 10.4(2) Bios:version 01.08
class-map type qos match-all global
match access-group name global-acl
class-map type qos match-all domestic
match access-group name domestic-acl
policy-map type qos pmap
class global
police cir 100 mbps bc 200 ms conform transmit violate drop
class domestic
police cir 200 mbps bc 200 ms conform transmit violate drop
```

About system ACLs

Use this reference to review system ACLs.

System ACL support

You can configure system ACLs on Cisco Nexus 9500 Series switches with -R and -RX line cards. With system ACLs, you can now configure a Layer 2 port ACL (PACL) on all the ports with the same access-list in the switch. Configuring system ACLs reduces the TCAM usage and also brings down the time and memory usage while the policy is being applied or modified.

Guidelines and limitations for system ACLs

See the following guidelines and limitations for configuring system ACLs:

- The system PACL is supported for Layer 2 interface only.
- Up to 10K ACEs are supported with all other basic features for the switch to come up on Cisco Nexus 9500 Series switches with -R line cards. The hardware capacity on Cisco Nexus 9500 Series switches with -RX line cards is 64K ACEs.
- You can also configure system ACLs on Cisco Nexus 3600 platform switches with N3K-C3636C-R and N3K-C36180YC-R line cards.
- Configuring IPv4 PACL TCAM region (ifacl) with anything more than the total physical TCAM capacity of -R line cards of 12k results in the power down of -R line cards only.
- ACE statistics are not yet supported for the system ACLs.
- IPv6 is not yet supported in the system ACLs.
- System ACLs are not supported on the breakout port.
- For quality of service, ACL, or TCAM carving configuration on Cisco Nexus Series switches with -R series line cards, see the Cisco Nexus 3600 NX-OS Quality of Service Configuration Guide, Release 7.x for more information.
- The non-atomic update either drops or permits all the traffic. By default, the non-atomic update drops all the traffic until the ACL update completes. The non-atomic ACL update behavior can be controlled using the **hardware access-list update default-result permit** CLI command. This CLI works only for physical ports. See the following example:

```
hardware access-list update default-result permit     => #Allows all the traffic during
 ACL updates. There may be < 10secs traffic drop.
no hardware access-list update default-result permit   => #This is the default behavior.
 It denies all the traffic during ACL updates.
```

- In Cisco NX-OS Release 9.2(2) and earlier releases, although the atomic ACL update is not supported on Cisco Nexus -R series line cards, the non-atomic update **hardware access-list update default-result** is supported on the Cisco Nexus -R series line cards.

Carving a TCAM region

Before configuring the system ACLs, carve the TCAM region first. Note that for configuring the ACLs less than 1k, you do not need to carve the TCAM region. See the

Configure ACL TCAM region sizes, on page 344 section for more information.

Configure system ACLs

After an IPv4 ACL is created, configure the system ACL.

Before you begin

Create an IPv4 ACL on the device. See Create an IP ACL, on page 337 for more information.

Procedure

Step 1 **configure terminal**

Enters the configuration mode.

Step 2 **system acl**

Configures the system ACL.

Step 3 **ip port access-group** <*pacl name*> **in**

Applies a Layer 2 PACL to the interface. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.

Configuration and show command examples for the system ACLs

See the following configuration examples for the system ACL show commands.

Configuring system PACL with 1K scale [using default TCAM]

See the following example for configuring system PACL with 1K scale [Using default TCAM].

Step 1: Create PACL.

```
config t
ip access-list PACL-DNA
    10 permit ip 192.0.2.1/32 any
    20 permit tcp 192.0.2.2/8 192.0.2.3 eq 1500
    25 deny udp any any eq 500
    26 deny tcp any eq 490 any
    …·· ….

    1000 deny any any
```

Step 2: Apply PACL into system level.

```
configuration terminal
system acl
    ip port access-group PACL-DNA in
```

Configuration and show command examples for the system ACL

To validate the system ACLs that are configured on the switch, use the **show running-config aclmgr | sec system** command:

```
switch#show running-config aclmgr | sec system
system acl
```

```
 ip port access-group test in
switch#
```

Configuration and show command examples for the system PACL

To validate the PACLs that are configured on the switch, use the **sh ip access-lists <name> [summary]** command:

```
switch#sh ip access-lists test
IP access list test
10 deny udp any any eq 27
20 permit ip 192.0.2.1/32 192.0.2.4/32
30 permit ip 192.0.2.5/32 192.0.2.4/32
40 permit ip 192.0.2.6/32 192.0.2.4/32
50 permit ip 192.0.2.7/32 192.0.2.4/32
60 permit ip 192.0.2.8/32 192.0.2.4/32
70 permit ip 192.0.2.9/32 192.0.2.4/32
80 permit ip 192.0.2.10/32 192.0.2.4/32
90 permit ip 192.0.2.11/32 192.0.2.4/32

switch#sh ip access-lists test summary
IPV4 ACL test
Total ACEs Configured: 12279
Configured on interfaces:
Active on interfaces:
- ingress
- ingress

switch#
```

To validate PACL IPv4 (ifacl) TCAM region size, use the **show hardware access-list tcam region** command:

```
switch#show hardware access-list tcam region
*********************************WARNING*********************************
*****************The output shows NFE tcam region info******************
***Please refer to 'show hardware access-list tcam template' for NFE2***
*************************************************************************
IPV4 PACL [ifacl] size = 12280
IPV6 PACL [ipv6-ifacl] size =    0
MAC PACL [mac-ifacl] size =    0
IPV4 Port QoS [qos] size =  640
IPV6 Port QoS [ipv6-qos] size =  256
MAC Port QoS [mac-qos] size =    0
FEX IPV4 PACL [fex-ifacl] size =    0
FEX IPV6 PACL [fex-ipv6-ifacl] size =    0
FEX MAC PACL [fex-mac-ifacl] size =    0
FEX IPV4 Port QoS [fex-qos] size =    0
FEX IPV6 Port QoS [fex-ipv6-qos] size =    0
FEX MAC Port QoS [fex-mac-qos] size =    0
IPV4 VACL [vacl] size =    0
IPV6 VACL [ipv6-vacl] size =    0
MAC VACL [mac-vacl] size =    0
IPV4 VLAN QoS [vqos] size =    0
IPV6 VLAN QoS [ipv6-vqos] size =    0
MAC VLAN QoS [mac-vqos] size =    0
IPV4 RACL [racl] size =    0
IPV6 RACL [ipv6-racl] size =  128
IPV4 Port QoS Lite [qos-lite] size =    0
FEX IPV4 Port QoS Lite [fex-qos-lite] size =    0
IPV4 VLAN QoS Lite [vqos-lite] size =    0
IPV4 L3 QoS Lite [l3qos-lite] size =    0
Egress IPV4 QoS [e-qos] size =    0
Egress IPV6 QoS [e-ipv6-qos] size =    0
Egress MAC QoS [e-mac-qos] size =    0
```

```
Egress IPV4 VACL [vacl] size =    0
Egress IPV6 VACL [ipv6-vacl] size =    0
Egress MAC VACL [mac-vacl] size =    0
Egress IPV4 RACL [e-racl] size =    0
Egress IPV6 RACL [e-ipv6-racl] size =    0
Egress IPV4 QoS Lite [e-qos-lite] size =    0
IPV4 L3 QoS [l3qos] size =  640
IPV6 L3 QoS [ipv6-l3qos] size =  256
MAC L3 QoS [mac-l3qos] size =    0
Ingress System size =    0
Egress System size =    0
SPAN [span] size =   96
Ingress COPP [copp] size =  128
Ingress Flow Counters [flow] size =    0
switch#
```

Configuration and show command examples for ACL related tech support

To view ACL related tech support information, use the **show tech-support aclmgr** and **show tech-support aclqos** commands.

```
show tech-support aclmgr
show tech-support aclqos
```

Configuring object groups

You can use object groups to specify source and destination addresses and protocol ports in IPv4 ACL and IPv6 ACL rules.

Session manager support for object groups

Session Manager supports the configuration of object groups. This feature allows you to create a configuration session and verify your object group configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*

Create and change an IPv4 address object group

You can create and change an IPv4 address group object.



Note Beginning Cisco Nexus Release 7.0(3)I5(2), the **no host IPv4-address** command is not supported. With the DME support, deletion without the no sequence command is not supported.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **object-group ip address** *name*

Example:

```
switch(config)#object-group ip address ipv4-addr-group-13
switch(config-ipaddr-ogroup)#
```

Creates the IPv4 address object group and enters IPv4 address object-group configuration mode.

Step 3 Enter one of the following commands:

- [*sequence-number*] **host** *IPv4-address*
- [*sequence-number*] *IPv4-address*/*prefix-len*
- [*sequence-number*] *IPv4-address network-wildcard*

Example:

```
switch(config-ipaddr-ogroup)#host 10.99.32.6
```

Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host, or omit the **host** command to specify a network of hosts.

You can specify a prefix length for an IPv4 object group, which matches only on the first contiguous bits, or you can specify a wildcard mask, which matches on any bit in the address.

Step 4 Enter one of the following commands:

- **no** [*sequence-number*]
- **no host** *IPv4-address*
- **no** *IPv4-address*/*prefix-len*
- **no** *IPv4-address network-wildcard*

Example:

```
switch(config-ipaddr-ogroup)#no host 10.99.32.6
```

Removes an entry in the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.

Step 5 (Optional) **show object-group** *name*

Example:

```
switch#show object-group ipv4-addr-group-13
```

Displays the object group configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Create and change an IPv6 address object group

You can create and change an IPv6 address group object.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **object-group ipv6 address** *name*

Example:

```
switch(config)#object-group ipv6 address ipv6-addr-group-A7
switch(config-ipv6addr-ogroup)#
```

Creates the IPv6 address object group and enters IPv6 address object-group configuration mode.

Step 3 Enter one of the following commands:

- [*sequence-number*] **host** *IPv6-address*
- [*sequence-number*] *IPv6-address*/*prefix-len*
- [*sequence-number*] *IPv6-address network-wildcard*

Example:

```
switch(config-ipv6addr-ogroup)#host 2001:db8:0:3ab0::1
```

Example:

```
switch(config-ipv6addr-ogroup)#10 2001:db8::1 2001:db8::2
```

Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host, or omit the **host** command to specify a network of hosts.

You can specify a prefix length for an IPv6 object group, which matches only on the first contiguous bits , or you can specify a wildcard mask, which matches on any bit in the address. IPv6 wildcard masks are supported for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FXP switches and the Cisco Nexus 9364C switch.

Step 4 Enter one of the following commands:

- **no** *sequence-number*
- **no host** *IPv6-address*
- **no** *IPv6-address*/*prefix-len*
- **no** *IPv6-address network-wildcard*

Example:

```
switch(config-ipv6addr-ogroup)#no host 2001:db8:0:3ab0::1
```

Removes an entry from the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.

Step 5 (Optional) **show object-group** *name*

Example:

```
switch#show object-group ipv6-addr-group-A7
```

Displays the object group configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Create and change a protocol port object group

You can create and change a protocol port object group.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **object-group ip port** *name*

Example:

```
switch(config)#object-group ip port NYC-datacenter-ports
switch(config-port-ogroup)#
```

Creates the protocol port object group and enters port object-group configuration mode.

Step 3 [*sequence-number*] *operator port-number* [*port-number*]

Example:

```
switch(config-port-ogroup)#eq 80
```

Creates an entry in the object group. For each entry that you want to create, use one of the following operator commands:

- **eq** —Matches only the port number that you specify.
- **gt** —Matches port numbers that are greater than (and not equal to) the port number that you specify.
- **lt** —Matches port numbers that are less than (and not equal to) the port number that you specify.
- **neq** —Matches all port numbers except for the port number that you specify.
- **range** —Matches the range of port numbers between and including the two port numbers that you specify.

Note
The **range** command is the only operator command that requires two *port-number* arguments.

Step 4 **no** {*sequence-number* | *operator port-number* [*port-number*]}

Example:

```
switch(config-port-ogroup)#no eq 80
```

Removes an entry from the object group. For each entry that you want to remove, use the **no** form of the applicable operator command.

Step 5 (Optional) **show object-group** *name*

Example:

```
switch#show object-group NYC-datacenter-ports
```

Displays the object group configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Remove an object group

You can remove an IPv4 address object group, an IPv6 address object group, or a protocol port object group.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no object-group** {**ip address** | **ipv6 address** | **ip port**} *name*

Example:

```
switch(config)#no object-group ip address ipv4-addr-group-A7
```

Removes the specified object group.

Step 3 (Optional) **show object-group**

Example:

```
switch#show object-group
```

Displays all object groups. The removed object group should not appear.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Object-group configuration verification commands

Use this reference to verify the Object-group configuration.

Configuration information configuration example

To display object-group configuration information, enter one of the following commands:

Table 77: Object-group configuration verification commands

Command	Purpose
show object-group	Displays the object-group configuration.
show {**ip** | **ipv6**} **access-lists** *name* [**expanded**]	Displays expanded statistics for the ACL configuration.
show running-config aclmgr	Displays the ACL configuration, including object groups.

Configuring Time-Ranges

Session manager support for Time-Ranges

Session Manager supports the configuration of time ranges. This feature allows you to create a configuration session and verify your time-range configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Create a Time-Range

You can create a time range on the device and add rules to it.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **time-range** *name*

Example:

```
switch(config)#time-range workday-daytime
switch(config-time-range)#
```

Creates the time range and enters time-range configuration mode.

Step 3 (Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*

Example:

```
switch(config-time-range)#periodic monday 00:00:00 to friday 23:59:59
```

Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.

Step 4 (Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*

Example:

```
switch(config-time-range)#periodic weekdays 06:00:00 to 20:00:00
```

Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument:

- **daily** —All days of the week.
- **weekdays** —Monday through Friday.
- **weekend** —Saturday through Sunday.

Step 5 (Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]

Example:

```
switch(config-time-range)#absolute start 1:00 15 march 2013
```

Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed.

Step 6 (Optional) [*sequence-number*] **absolute** [start *time date*] **end** *time date*

Example:

```
switch(config-time-range)#absolute end 23:59:59 31 may 2013
```

Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.

Step 7 (Optional) **show time-range** *name*

Example:

```
switch#show time-range workday-daytime
```

Displays the time-range configuration.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Change a Time-Range

You can add and remove rules in an existing time range. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **time-range** *name*

Example:

```
switch(config)#time-range workday-daytime
switch(config-time-range)#
```

Enters time-range configuration mode for the specified time range.

Step 3 (Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*

Example:

```
switch(config-time-range)#periodic monday 00:00:00 to friday 23:59:59
```

Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.

Step 4 (Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*

Example:

```
switch(config-time-range)#100 periodic weekdays 05:00:00 to 22:00:00
```

Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument:

- **daily** —All days of the week.
- **weekdays** —Monday through Friday.
- **weekend** —Saturday through Sunday.

Step 5 (Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]

Example:

```
switch(config-time-range)#absolute start 1:00 15 march 2013
```

Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed.

Step 6 (Optional) [*sequence-number*] **absolute** [**start** *time date*] **end** *time date*

Example:

```
switch(config-time-range)#absolute end 23:59:59 31 may 2013
```

Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.

Step 7 (Optional) **no** {*sequence-number* | **periodic** *arguments* . . . | **absolute** *arguments*. . .}

Example:

```
switch(config-time-range)#no 80
```

Removes the specified rule from the time range.

Step 8 (Optional) **show time-range** *name*

Example:

```
switch#show time-range workday-daytime
```

Displays the time-range configuration.

Step 9 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Change sequence numbers in a time range, on page 390

Remove a Time-Range

You can remove a time range from the device.

Before you begin

Ensure that you know whether the time range is used in any ACL rules. The device allows you to remove time ranges that are used in ACL rules. Removing a time range that is in use in an ACL rule does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the ACL rule using the removed time range to be empty.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no time-range** *name*

Example:

```
switch(config)#no time-range daily-workhours
```

Removes the time range that you specified by name.

Step 3 (Optional) **show time-range**

Example:

```
switch#show time-range
```

Displays the configuration for all time ranges. The removed time range should not appear.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Change sequence numbers in a time range

You can change all the sequence numbers assigned to rules in a time range.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **resequence time-range** *name starting-sequence-number increment*

Example:

```
switch(config)#resequence time-range daily-workhours 100 10
switch(config)#
```

Assigns sequence numbers to the rules contained in the time range, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify.

Step 3 (Optional) **show time-range** *name*

Example:

```
switch#show time-range daily-workhours
```

Displays the time-range configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Time-range configuration verification commands

Use this reference to verify the Time-range configuration.

Table 78: Time-range configuration verification commands

Command	Purpose
show time-range	Displays the time-range configuration.
show running-config aclmgr	Displays ACL configuration, including all time ranges.

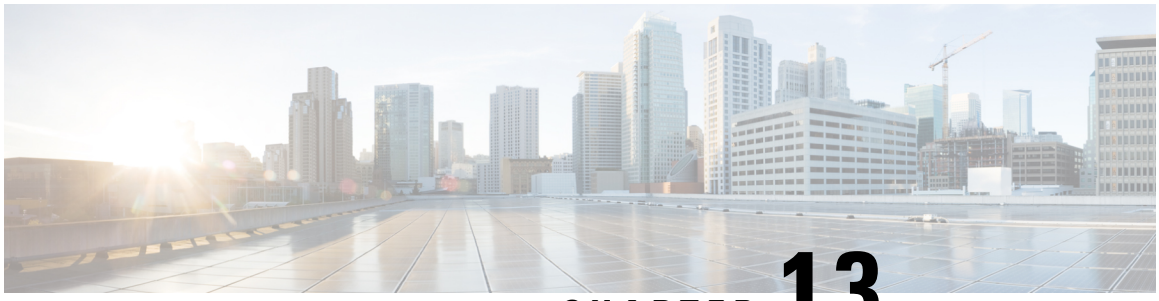
Additional references for IP ACLs

Use this reference to find additional references for IP ACLs.

Related Documents

Table 79: Related documents

Related Topic	Document Title
TAP aggregation	Configuring TAP Aggregation and MPLS Stripping



CHAPTER 13

Configuring MAC ACLs

This chapter describes how to configure MAC access lists (ACLs) on Cisco NX-OS devices.

This chapter contains the following sections:

- MAC ACLs, on page 393
- Guidelines for configuring MAC ACLs, on page 394
- Default settings for MAC ACLs, on page 395
- Configuring MAC ACLs, on page 395
- MAC ACL configuration verification commands, on page 405
- MAC ACL statistics monitoring and clearing, on page 405
- Examples of MAC ACL configuration, on page 406
- Additional references for MAC ACLs, on page 406

MAC ACLs

A MAC access control list (ACL)

- filters network traffic using information in the Layer 2 header.
- works like an IP access control list (ACL) and supports virtualization
- applies filtering based on MAC addresses instead of IP addresses.

MAC packet classification

Media Access Control packet characteristics include the following:

- is identified and filtered using Media Access Control access control lists (MAC ACLs) at Layer 2 interfaces,
- Packets can be classified to affect how traffic, including both Internet Protocol (IP) and non-Internet Protocol (non-IP), is handled.
- This enables differentiated control over interface access for security and policy enforcement.

Media Access Control (MAC) packet classification determines whether a Media Access Control access control list (MAC ACL) on a Layer 2 interface applies to all incoming traffic, including Internet Protocol (IP) traffic, or applies only to non-Internet Protocol (non-IP) traffic.

Table 80: MAC packet classification

MAC Packet Classification State	Effect on the interface
Enabled	• A Media Access Control access control list (MAC ACL) that is on the interface applies to all traffic entering the interface, including Internet Protocol (IP) traffic. • You can apply an IP port ACL to the interface.
Disabled	• A MAC ACL that is on the interface applies only to non-IP traffic entering the interface. • You can apply an Internet Protocol (IP) port access control list (ACL) to the interface.

Guidelines for configuring MAC ACLs

Follow these guidelines and review the limitations when you configure MAC ACLs.

MAC ACL best practices for the supported platforms

- MAC packet classification is not supported when MAC ACLs are used as match criteria for QoS policies on Nexus 9300 Series switches with forty-gigabit (40-gigabit) [43.6 billion bits per second] uplink ports.
- When you define a MAC ACL on the Nexus 9000 first-generation and 9300-EX switches, you must define the ethertype to match the traffic appropriately. For Nexus 9300-FX and later release switches, the **all** keyword is available, which replaces the need to specify the ethertype.

  The 9300-EX switches do not support the **all** keyword.
- Beginning NX-OS Release 9.3(2), you can configure a user-defined MAC address limit in the range of 16 to 256.
- Nexus 93600CD-GX switches do not support breakout on ports 1 through 24.
- Beginning with NX-OS Release 10.4(1)F, a new ACE keyword (all) is provided for applying the MAC ACL rule priority over SUP rule on Nexus 9300- EX/ FX/FX2/FX3/GX/GX2 , C9364C, C9332C, and Cisco Nexus 9500 with 9700-EX/ FX/GX line cards.

MAC ACL general best practices

- MAC ACLs control only ingress traffic.
- If you apply too many ACL entries, your configuration might be rejected.
- MAC packet classification is not supported when a MAC ACL is applied as part of a VACL.
- Mac-packet classify knob is partially supported on the Nexus 9300-EX platform switches. The switches match all packets with certain fields, such as src_mac, dst_mac, and vlan in the key field because there is no direct field for marking the packet as an L2 packet. They cannot match on the eth_type field. If you install two rules with identical fields except for the MAC protocol number field, the match conditions

are identical in the hardware. The first entry in the rule sequence hits for all packets for all protocol numbers. The MAC protocol number will have no effect when you configure mac-packet classify.

- When you set a user-defined MAC limit using the **mac address-table limit <16-256> user-defined** command, the FHRP group limit is automatically adjusted so that the sum of user-defined MAC limits and FHRP limits equals 490. For example, if you set the user-defined MAC limit to 100, the FHRP limit is reduced to 390.
- A MAC access list applied to an interface will not block Bridge Protocol Data Unit (BPDU) traffic, such as Spanning Tree Protocol BPDUs

Default settings for MAC ACLs

Use this table to find the default settings for MAC ACL parameters.

Table 81: Default MAC ACLs parameters

Parameters	Default
MAC ACLs	By default, no MAC ACLs are configured.
ACL rules	Implicit rules are applied to all access control lists.

Configuring MAC ACLs

Explains configuring MAC ACLs.

Create a MAC ACL

You define and configure a MAC access control list to filter traffic based on MAC addresses.

You can use this task to create and configure a MAC access control list (ACL) that includes permit or deny rules in the command-line interface (CLI).

Before you begin

Access the global configuration mode on the NX-OS device.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command .

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Create the MAC ACL and enter ACL configuration mode by using the **mac access-list** *name* command.

Example:

```
switch(config)#mac access-list acl-mac-01
switch(config-mac-acl)#
```

Notice
The names **dynamic** , expanded, and summary are reserved for system-defined access lists. Do not use these names for user-defined ACLs, as this can cause conflicts when displaying or verifying your configuration.

Step 3 Create a rule in the MAC access control list by using the {**permit** | **deny**} *source destination-protocol* command.

Example:

```
switch(config-mac-acl)#100 permit mac 00c0.4f00.0000 0000.00ff.ffff any 0x0806
```

The **permit** and **deny** commands support many ways of identifying traffic.

Step 4 (Optional) Specify that the device collects global statistics for packets matching the ACL rules by using the **statistics per-entry** command.

Example:

```
switch(config-mac-acl)#statistics per-entry
```

Step 5 (Optional) Display the MAC ACL configuration by using the **show mac access-lists** *name* command.

Example:

```
switch#show mac access-lists acl-mac-01
```

Step 6 (Optional) Save the configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

You create and configure the MAC access control list with the specified rules. Apply the list to interfaces or verify its configuration.

Configure a UDF-Based MAC ACL

You can configure UDF-based MAC access lists (ACLs) for the Cisco Nexus 9200, 9300 , and 9300-EX Series switches. This feature enables the device to match on user-defined fields (UDFs) and apply MAC ACLs to the matching packets.

Starting with Cisco NX-OS Release 9.3(3), you can configure UDF-based access lists (ACLs) on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

Procedure

Step 1 Enter global configuration mode by using the command **configure terminal** and define the UDF using the command **udf** *udf-name offset-base offset length* .

Example:

```
switch#configure terminal
switch(config)#udf pktoff10 packet-start 10 2
```

The following defines the UDF.

- *udf-name* —Specifies the name of the UDF. You can enter up to sixteen alphanumeric characters for the name.
- *offset-base* —Specifies the UDF offset base as follows: {**packet-start**} .
- *offset* —Specifies the number of bytes offset from the offset base.
- *length* —Specifies the number of bytes from the offset. Only one or two bytes are supported. To match additional bytes, you must define multiple UDFs.

You can define multiple UDFs; however, Cisco recommends defining only the required UDFs.

Step 2 Attach the UDFs to the ing-ifacl TCAM region for IPv4 or IPv6 port ACLs. To do this, use the } **hardware access-list tcam region ing-ifacl qualify** {**udf** *udf-name*} command.

Example:

Note
This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

Example:

```
switch#hardware access-list tcam region ing-ifacl qualify udf pktoff10
```

Up to eighteen UDFs are supported.

Note
When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Ensure there is enough free space available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see Configuring ACL TCAM Region Sizes .

Note
The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.

Step 3 Required: Ensure the configuration is saved after reboots and restarts, copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Step 4 Required: Reload the device by using the **reload** command.

Example:

```
switch#reload
```

Note
The UDF configuration takes effect after you enter the **copy running-config startup-config** and **reload** commands, and then reload the device.

Step 5 Create a MAC access control list (ACL) and enter MAC ACL configuration mode by using the **mac access-list** *udf-acl* command.

Example:

```
switch(config)#mac access-list udfacl
switch(config-acl)#
```

Step 6 Configure the MAC ACL to match on UDFs using the **permit mac** *source destination* **udf** *udf-name value mask* command.

Example:

```
switch(config-mac-acl)#permit mac any any udf pktoff10 0x1234 0xffff
```

Configure the MAC ACL to match on UDFs together with the current access control entries (ACEs) for the outer packet fields. The *value* and *mask* arguments can range from 0x0 to 0xffff.

A single ACL can have ACEs with or without UDFs. Each ACE can match a different UDF field, or all ACEs can match the same UDF or UDFs.

Step 7 Enter interface configuration mode for a Layer 2 port-channel interface using the **interface port-channel** *channel-number* command.

Example:

```
switch(config)#interface port-channel 5
switch(config-if)#
```

Step 8 Apply the UDF-based MAC ACL to the interface by using the **mac port access-group** *udf-access-list* command.

Example:

```
switch(config-if)#mac port access-group udfacl
```

Step 9 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

You can configure UDF-based MAC access lists (ACLs) and apply them to matching packets.

Modify a MAC ACL

You can use this task to modify a MAC ACL on your device

Modify a MAC ACL by completing these steps:

Before you begin

Use the **show mac access-lists** command with the **summary** keyword. Find the interfaces that have a MAC ACL configured.

Procedure

Step 1 Enter the global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter MAC ACL configuration mode for the target ACL by using the **mac access-list** *name* command.

Example:

```
switch(config)#mac access-list acl-mac-01
switch(config-mac-acl)#
```

Step 3 (Optional) Create a rule in the MAC ACL by using the [*sequence-number*] {**permit** | **deny**} *source destination-protocol* command.

Example:

```
switch(config-mac-acl)#100 permit mac 00c0.4f00.0000 0000.00ff.ffff any 0x0806
```

You can specify a position for the rule in the ACL by using a sequence number. If you do not provide a sequence number, the rule is added to the end of the list.

Use the **permit** and **deny** commands to identify traffic.

Step 4 (Optional) Remove the specific rule by sequence number or criteria by using the **no** {*sequence-number* | {**permit** | **deny**} *source destination-protocol*} command.

Example:

```
switch(config-mac-acl)#no 80
```

The **permit** and **deny** commands support many ways of identifying traffic.

Step 5 (Optional) Use the [**no**] **statistics per-entry** command to specify the device that maintains global statistics for packets matching the rules in the ACL

Example:

```
switch(config-mac-acl)#statistics per-entry
```

Use the **no statistics per-entry** command to stop the device from maintaining global statistics for packets that match the rules in the ACL.

Step 6 (Optional) You can display the updated MAC ACL configuration by using the **show mac access-lists** *name* command.

Example:

```
switch#show mac access-lists acl-mac-01
```

Step 7 (Optional) (Optional) Save the configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

You modified the specified MAC ACL rule on your device.

What to do next

Verify that you modified the ACL and confirm that your device functions as expected.

Change sequence numbers in a MAC ACL

You can change all the sequence numbers assigned to rules in a MAC ACL. Resequencing is useful when you need to insert rules into an ACL and there are not enough available sequence numbers.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Assign sequence numbers to the rules contained in the ACL by using the **resequence mac access-list** *name starting-sequence-number increment* command.

Example:

```
switch(config)#resequence mac access-list acl-mac-01 100 10
```

The first rule receives the starting sequence number that you specify. Each subsequent rule receives a number greater than the previous rule. The difference in numbers is determined by the increment number that you specify.

Step 3 (Optional) Display the MAC ACL configuration by using the **show mac access-lists** *name* command.

Example:

```
switch#show mac access-lists acl-mac-01
```

Step 4 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Remove a MAC ACL

Remove an Access Control List (ACL) from the running configuration to stop specific traffic filtering.

Perform this task when you no longer need an ACL or plan to replace it on the device.

Before you begin

- You need administrative privileges to modify the device configuration.
- Identify the name of the ACL you want to remove.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Remove the MAC ACL by name from the running configuration using the **no mac access-list** *name* command.

Example:

```
switch(config)#no mac access-list acl-mac-01
switch(config)#
```

Step 3 (Optional) Display the MAC ACL configuration by using the **show mac access-lists** *name* **summary** command.

Example:

```
switch#show mac access-lists acl-mac-01 summary
```

If the ACL remains applied to an interface, the command lists the interfaces.

Step 4 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

The switch removes the specified ACL from the running configuration and from the startup configuration if you save the changes.

What to do next

- Verify that network traffic behaves as expected after removing the ACL.
- Remove the ACL from any interface where it is still applied.

Apply a MAC ACL as a port ACL

You can apply a MAC ACL as a port ACL to these interface types:

- Layer 2 Ethernet interfaces
- Layer 2 port-channel interfaces

Before you begin

Confirm that the required ACL is present and configured to filter traffic appropriately for your use case.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter one of these commands:

- **interface ethernet** *slot* / *port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Example:

```
switch(config)#interface port-channel 5
switch(config-if)#
```

- Enter interface configuration mode for a Layer 2 or Layer 3 interface.
- Enter interface configuration mode for a Layer 2 or Layer 3 port-channel interface.

Step 3 Applies a MAC ACL to the interface by using the **mac port access-group** *access-list* command.

Example:

```
switch(config-if)#mac port access-group acl-01
```

Step 4 (Optional) Display the ACL configuration by using the **show running-config aclmgr** command.

Example:

```
switch#show running-config aclmgr
```

Step 5 (Optional) Copies the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Apply a MAC ACL as a VACL

You can apply a MAC ACL as a VACL.

Enable or disable MAC packet classification

You can enable or disable MAC packet classification on a Layer 2 interface.

Before you begin

The interface must be configured as a Layer 2 interface.

Note If the interface is configured with the **ip port access-group** command or the **ipv6 port traffic-filter** command, you cannot enable MAC packet classification until you remove the **ip port access-group** and **ipv6 port traffic-filter** commands from the interface configuration.

Procedure

Step 1 Enters the global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter one of the following commands:

- **interface ethernet** *slot* / *port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Example:

```
switch(config)#interface port-channel 5
switch(config-if)#
```

- Enter interface configuration mode for an Ethernet interface, or
- Enter interface configuration mode for a port-channel interface.

Step 3 Enable MAC packet classification on the interface by using the command [**no**] **mac packet-classify** .

Example:

```
switch(config-if)#mac packet-classify
```

Use the **no** form of the command to disable MAC packet classification on the interface.

Step 4 (Optional) Enter one of the following commands:

- **show running-config interface ethernet** *slot* / *port*
- **show running-config interface port-channel** *channel-number*

Example:

```
switch#show running-config interface ethernet 2/1
```

Example:

```
switch#show running-config interface port-channel 5
```

- Display the running configuration of the Ethernet interface, or
- Display the running configuration of the port-channel interface.

Step 5 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Prioritize a MAC ACL rule over a SUP rule

In Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is supported for MAC ACLs. This keyword increases the priority of the ACL rule to 0 (the highest), giving it precedence over any other SUP ACL rule that matches the same criteria.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
```

Step 2 Create a MAC ACL and enter ACL configuration mode by using the **mac access-list** *name* command.

Example:

```
switch(config)#mac access-list acl-mac-01
 switch(config-acl)#
```

The *name* argument can be up to 64 characters.

Step 3 Create a rule in a MAC ACL using the {**permit** | **deny**} *source destination-protocol* **all** command.

Example:

```
switch(config-mac-acl)#100 permit 00c0.4f00.0000 0000.00ff.ffff any all
```

The **all** keyword prioritizes the MAC ACL rule over the SUP rule.

Step 4 Enter one of the following commands:

- **interface ethernet** *slot* / *port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

Enter configuration mode for the interface type that you specified.

Step 5 Apply a MAC ACL to the interface or port channel by using the **mac port access-group** *access-list* **in** command.

Example:

```
switch(config-if)#mac port access-group acl-mac-01 in
```

Port ACLs support only inbound filtering. You can apply one port ACL to each interface.

Step 6 (Optional) Display the ACL configuration by using the **show running-config aclmgr** command.

Example:

```
switch#show running-config aclmgr
```

Step 7 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

MAC ACL configuration verification commands

To display MAC ACL configuration information, perform one of the following tasks:

Table 82: Commands to verify MAC ACL configuration

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration.
show running-config aclmgr [**all**]	Displays the ACL configuration, including MAC ACLs and the interfaces to which MAC ACLs are applied. **Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

MAC ACL statistics monitoring and clearing

Use these commands to monitor or clear MAC ACL statistics on the device.

Table 83: Commands for monitoring and clearing MAC ACL statistics

Command	Purpose
show mac access-lists	This command displays the MAC ACL configuration. If the MAC ACL includes the **statistics per-entry** command, the output of **show mac access-lists** command also includes the number of packets that have matched each rule.
clear mac access-list counters	Clears statistics for MAC ACLs.

Examples of MAC ACL configuration

MAC ACLs (Media Access Control Lists) filter network traffic based on source and destination MAC addresses and are applied to Layer 2 interfaces.

```
mac access-list acl-mac-01
permit 00c0.4f00.0000 0000.00ff.ffff any 0x0806
interface ethernet 2/1
mac port access-group acl-mac-01
```

This configuration:

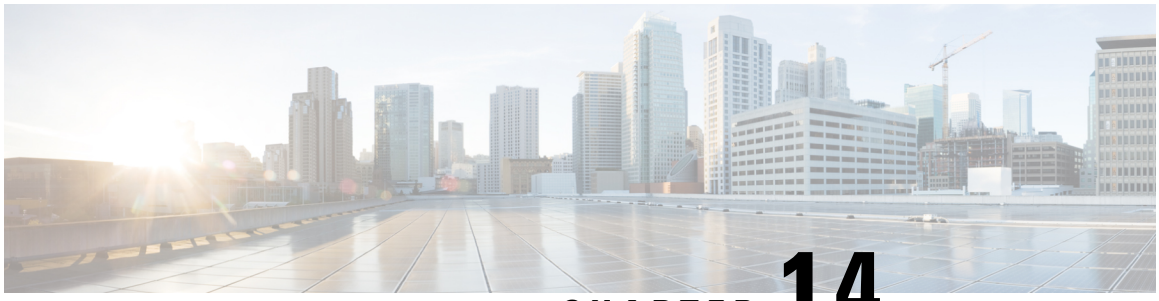
- Creates a MAC ACL named acl-mac-01.
- Permits Address Resolution Protocol (ARP) packets from the specified MAC address range.
- Applies the ACL to Ethernet interface 2/1.

Additional references for MAC ACLs

Use this table for related documents for Media Access Control (MAC) access control lists (ACLs).

Table 84: Related documents

Related Topic	Document Title
TAP aggregation	Configuring TAP Aggregation and MPLS Stripping



CHAPTER 14

Configuring VLAN ACLs

This chapter describes how to configure VLAN access lists (ACLs) on Cisco NX-OS devices.

This chapter includes the following sections:

- VLAN ACLs, on page 407
- Prerequisite requirements for VACLs, on page 408
- Best practices and limitations for VACLs, on page 408
- Default settings for VACLs, on page 409
- Configuring VACLs, on page 410
- VACL configuration verification commands, on page 413
- Commands for monitor and clear VACL statistics, on page 414
- Configuration example for VACLs, on page 414
- Additional references for VACLs, on page 414

VLAN ACLs

A VLAN ACL is a network security rule set that

- applies to all packets that are routed into, out of, or bridged within a VLAN,
- filters or redirects traffic within the VLAN for security purposes, and
- directionless and does not distinguish between ingress and egress.

VLAN access maps and entries

VACLs use access maps to contain an ordered list of map entries. Each map entry associates an IP or MAC ACL with an action. Each entry has a sequence number. The sequence number determines the precedence of entries.

When the device applies a VACL to a packet, it uses the action configured in the first access map entry with an ACL that permits the packet.

Actions in VACLs

In access map configuration mode, you use the **action** command to specify one of the following actions:

- Forward: sends the traffic to the destination determined by the normal operation of the device.
- Redirect: redirects the traffic to one or more specified interfaces.
- Drop: drops the traffic. If you specify drop as the action, you can also specify that the device logs the dropped packets.

VACL statistics

You can view global statistics for each rule in a VACL. When you apply a VACL to multiple VLANs, the rule statistics show the sum of packet matches (hits) on all interfaces where you applied the VACL.



Note You cannot view VACL statistics at the interface level.

For each VLAN access map you configure, you decide whether to keep statistics for that VACL. Turn VACL statistics on or off to monitor filtered traffic or troubleshoot VLAN access map configuration.

VACLs in session manager

A Session manager VACL is a configuration feature that

- allows you to verify ACL configurations prior to applying them,
- ensures required resources are available before committing changes, and
- helps prevent errors in running configurations.

For more information about Session Manager, see *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide* .

Prerequisite requirements for VACLs

Learn about the prerequisites you need to configure VACLs on your network devices. Make sure the IP access control list (ACL) or MAC ACL you use for a VACL exists. Configure it to filter traffic according to your specified criteria.

Best practices and limitations for VACLs

General guidelines

Follow these configuration guidelines to set up VACLs:

- Use Session Manager to configure access control lists. This feature allows you to verify ACL configuration and check that required resources are available before you commit them to the running configuration. For more information about Session Manager, see the Nexus 9000 Series NX-OS System Management Configuration Guide.

- Apply the necessary number of ACL entries. If you add too many entries, your configuration will be rejected.
- You can redirect VACLs only to supported ports. You cannot use VACL redirects on SPAN destination ports.
- You cannot configure logging with VACLs.
- When you apply a VACL to multiple VLANs, TCAM resources are not shared.
- To drop traffic, use permit statements with the action 'drop' on VACLs; deny statements are not supported.
- When you configure a VACL with the 'redirect' option, define the redirect interface as a member of the VLAN where you apply the VACL. Make sure this VLAN is in the forwarding state on the interface to enable redirection. If the VLAN is not in the forwarding state, packets matched by the VACL will be dropped.
- If a VLAN access-map entry does not include a match statement, the default behavior is to deny (drop) the traffic for that packet type. To allow traffic, you must configure a match statement under an "action forward" entry that specifies the permit ACL.
- Configure active VLAN filters to enable clearing of VACL counters.

Platform and software guidelines

These guidelines apply to the platform and software release:

- To redirect traffic on Nexus 9200 and 9300-EX Series switches, use the VACL redirect option and specify one physical or port-channel interface.
- Avoid configuring VACLs on Nexus 9500 Series switches equipped with the 36-port 40 to 100 Gigabit Ethernet QSFP28 line card (with additional TCAM) or the 36-port 40 Gigabit Ethernet QSFP line card, because these configurations are not supported.
- As of NX-OS Release 10.1(2), you can use VACLs on the 24-port 400 to 100 Gigabit Ethernet QSFP-DD line card 438.91/100 Gigabit Ethernet (438.91 or 100 billion bits per second) and on the Fabric Module for Nexus 9508 R2-Series line cards.
- On Nexus 9332D-GX2B platform switches running NX-OS Release 10.2(1q)F, you can configure VACL, DACL, and CoPP.

VACLs for VXLANs guidelines

These guidelines apply to VACLs for VXLANs:

- The inner payload supports VACLs applied on a VXLAN VLAN in the access to network direction (Layer 2 to Layer 3 encapsulation path).
- Use VACLs on the access side to filter traffic entering the overlay network.
- You cannot use egress VACLs for decapsulated VXLAN traffic.

Default settings for VACLs

This table lists the default settings for VACL parameters.

Table 85: Default VACL parameters

Parameters	Default
VACLs	No IP access control lists (ACLs) exist by default.
ACL rules	Implicit rules apply to all access control lists (ACLs).

Configuring VACLs

Explains configuring VACLs.

Create a VACL entry

You can create a VACL or add entries to an existing VACL. In both cases, you create a VACL entry, which is a VLAN access-map entry that associates one or more ACLs with an action to be applied to the matching traffic.

Before you begin

To use an ACL in the VACL, make sure it already exists and is configured to filter traffic as needed for this application.

Procedure

Step 1 Enter the global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter VLAN access-map configuration mode for the specified VLAN access map using the **vlan access-map** *map-name* [*sequence-number*] command.

Example:

```
switch(config)#vlan access-map acl-mac-map
switch(config-access-map)#
```

- If the VLAN access map does not exist, the device creates it.
- If you do not specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map.

Step 3 Specify an ACL for the access-map entry using one of the following commands:

- **match** {**ip** | **ipv6**} **address** *ip-access-list*
- **match mac address** *mac-access-list*

Example:

```
switch(config-access-map)#match mac address acl-ip-lab
```

Example:

```
switch(config-access-map)#match mac address acl-mac-01
```

Step 4 Specify the action that the device applies to traffic that matches the ACL using the **action** {**drop** | **forward** | **redirect**} command.

Example:

```
switch(config-access-map)#action forward
```

Example:

```
switch(config-access-map)#action redirect e1/1
switch(config-access-map)#action redirect po100
```

The **action** command supports the **drop** , **forward** , and **redirect** options.

Step 5 (Optional) Specify that the device maintains global statistics for packets that match the rules in the VACL using the [**no**] **statistics per-entry** command.

Example:

```
switch(config-access-map)#statistics per-entry
```

The **no** option stops the device from maintaining global statistics for the VACL.

Step 6 (Optional) Display the ACL configuration using the **show running-config aclmgr** command.

Example:

```
switch#show running-config aclmgr
```

Step 7 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

A VACL entry is created or updated. Traffic is filtered, redirected, or forwarded, according to your configuration.

What to do next

- Verify operation by using the appropriate show commands.
- Review device behavior for correct traffic processing.

Remove a VACL or VACL entry

Remove a VLAN access control list (VACL) or an individual entry from a VACL in your network switch configuration.

Perform this task if you need to delete a VACL or a specific VACL entry. You can delete the VLAN access map as part of this process.

To remove a VACL or VACL entry, complete these steps.

Before you begin

Ensure that you know whether the VACL is applied to a VLAN. The device allows you to remove VACLs that are currently applied. Removing a VACL does not affect the configuration of VLANs where the VACL was applied. After removal, the device treats the VACL as an empty configuration.

Procedure

Step 1 Enter the global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Delete the VLAN access map configuration for the specified access map using the **no vlan access-map** *map-name* [*sequence-number*] command.

Example:

```
switch(config)#no vlan access-map acl-mac-map 10
```

The command removes only the specified entry if you provide the *sequence-number* argument and the VACL contains more than one entry.

Step 3 (Optional) Display the ACL configuration using the **show running-config aclmgr** command.

Example:

```
switch#show running-config aclmgr
```

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

The specified VACL or VACL entry is removed from the device configuration.

What to do next

Verify that the intended VACL or entry is deleted and that network access policies function as expected.

Apply a VACL to your VLAN

You can apply a VACL to a VLAN.

Before you begin

Before you apply a VACL, make sure that the VACL exists and filters traffic for your application.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Apply the VACL to the specified VLANs using the [**no**] **vlan filter** *map-name* **vlan-list** *list* command.

Example:

```
switch(config)#vlan filter acl-mac-map vlan-list 1-20,26-30
switch(config)#
```

Use the **no** option to not apply the VACL.

Step 3 (Optional) Display the ACL configuration using the **show running-config aclmgr** command.

Example:

```
switch#show running-config aclmgr
```

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

VACL configuration verification commands

Use these commands to verify VACL (VLAN Access Control List) configuration.

Table 86: Commands for VACL configuration verification.

Command	Purpose
show running-config aclmgr [**all**]	Displays the ACL configuration, including the VACL-related configuration. **Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.
show vlan filter	Displays information about VACLs that are applied to a VLAN.
show vlan access-map	Displays information about VLAN access map.

Commands for monitor and clear VACL statistics

Use one of these commands to monitor or clear VACL statistics.

Table 87: Commands for monitoring and clearing VACL statistics

Command	Purpose
show vlan access-list	Displays the VACL configuration. If the VLAN access-map includes the **statistics per-entry** command, the **show vlan access-list** command output includes the number of packets that have matched each rule.
clear vlan access-list counters	Clears statistics for VACLs.

Configuration example for VACLs

Configure a VACL to forward traffic permitted by a MAC ACL named acl-mac-01 and apply the VACL to VLANs 50 to 82.

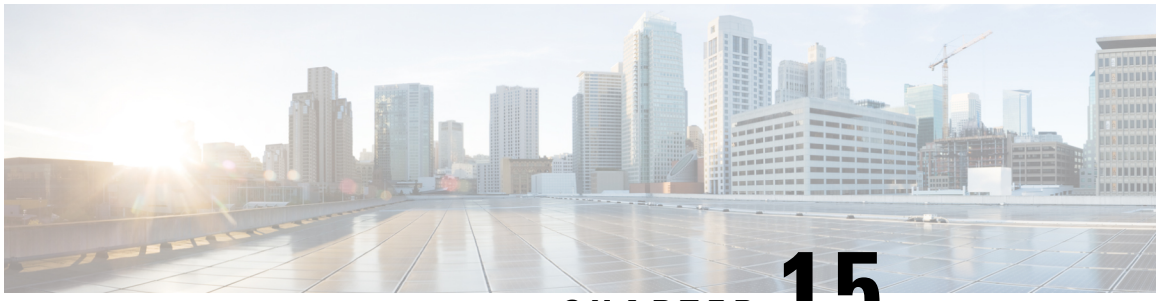
```
configure terminal
vlan access-map acl-mac-map
match mac address acl-mac-01
action forward
vlan filter acl-mac-map vlan-list 50-82
```

Additional references for VACLs

Use this table to find documents about VACL configuration and related topics.

Table 88: Related documents

Related Topic	Document Title
QoS configuration	*Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*



CHAPTER 15

Configuring Port Security

This chapter describes how to configure port security on Cisco NX-OS devices.

This chapter includes the following sections:

- About port security, on page 415
- Prerequisites for port security, on page 421
- Default settings for port security, on page 421
- Guidelines and limitations for port security, on page 422
- Guidelines and limitations for port security on vPCs, on page 423
- Configuring port security, on page 424
- Port security configuration verification commands, on page 434
- Displaying secure MAC addresses, on page 435
- Configuration example for port security, on page 435
- Configuration examples for port security in a vPC domain, on page 435
- Additional references for port security, on page 437
- Port security support for VXLAN EVPN, on page 437

About port security

Port security allows you to configure Layer 2 physical interfaces and Layer 2 port-channel interfaces to allow inbound traffic from only a restricted set of MAC addresses. The MAC addresses in the restricted set are called secure MAC addresses. In addition, the device does not allow traffic from these MAC addresses on another interface within the same VLAN. The number of MAC addresses that the device can secure is configurable per interface.

Port security information

Note Unless otherwise specified, the term *interface* refers to both physical interfaces and port-channel interfaces; likewise, the term *Layer 2 interface* refers to both Layer 2 physical interfaces and Layer 2 port-channel interfaces.

Secure MAC address learning

The process of securing a MAC address is called learning. A MAC address can be a secure MAC address on one interface only. For each interface on which you enable port security, the device can learn a limited number of MAC addresses by the static or dynamic methods. The way that the device stores secure MAC addresses varies depending upon how the device learned the secure MAC address.

Static method

The static learning method allows you to manually add or remove secure MAC addresses to the running configuration of an interface. If you copy the running configuration to the startup configuration, static secure MAC addresses are unaffected if the device restarts.

A static secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address from the configuration.
- You configure the interface to act as a Layer 3 interface.

Adding secure addresses by the static method is not affected by whether dynamic address learning is enabled.

Dynamic method

By default, when you enable port security on an interface, you enable the dynamic learning method. With this method, the device secures MAC addresses as ingress traffic passes through the interface. If the address is not yet secured and the device has not reached any applicable maximum, it secures the address and allows the traffic.

The device stores dynamic secure MAC addresses in memory. A dynamic secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- The device restarts
- The interface restarts
- The address reaches the age limit that you configured for the interface
- You explicitly remove the address
- You configure the interface to act as a Layer 3 interface

Sticky method

If you enable the sticky method, the device secures MAC addresses in the same manner as dynamic address learning, but the device stores addresses learned by this method in nonvolatile RAM (NVRAM). As a result, addresses learned by the sticky method persist through a device restart. Sticky secure MAC addresses do not appear in the running configuration of an interface.

Dynamic and sticky address learning are mutually exclusive. When you enable sticky learning on an interface, the device stops dynamic learning and performs sticky learning instead. If you disable sticky learning, the device resumes dynamic learning.

A sticky secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address
- You configure the interface to act as a Layer 3 interface

Dynamic address aging

The device ages MAC addresses learned by the dynamic method and drops them after the age limit is reached. You can configure the age limit on each interface. The range is from 0 to 1440 minutes, where 0 disables aging.

The method that the device uses to determine that the MAC address age is also configurable. The two methods of determining address age are as follows:

- **Inactivity**

  The length of time after the device last received a packet from the address on the applicable interface.

Note This feature is supported only on Cisco Nexus 9200 and 9300-EX Series switches.

- **Absolute**

  The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

Note When the absolute aging time is configured, MAC aging occurs even when the traffic from the source MAC is flowing. However, during MAC aging and re-learn, there could be a transient traffic drop.

Secure MAC address maximums

By default, an interface can have only one secure MAC address. You can configure the maximum number of MAC addresses permitted per interface or per VLAN on an interface. Maximums apply to secure MAC addresses learned by any method: static or dynamic.

Tip To ensure that an attached device has the full bandwidth of the port, set the maximum number of addresses to one and configure the MAC address of the attached device.

The following three limits can determine how many secure MAC addresses are permitted on an interface:

- **Device Maximum**

  The device has a nonconfigurable limit of 8192 secure MAC addresses. If learning a new address would violate the device maximum, the device does not permit the new address to be learned, even if the interface or VLAN maximum has not been reached.

- **Interface Maximum**

You can configure a maximum number of 1025 secure MAC addresses for each interface protected by port security. The default interface maximum is one address. Interface maximums cannot exceed the device maximum.

- **VLAN Maximum**

  You can configure the maximum number of secure MAC addresses per VLAN for each interface protected by port security. A VLAN maximum cannot exceed the configured interface maximum. VLAN maximums are useful only for trunk ports. There are no default VLAN maximums.

You can configure VLAN and interface maximums per interface, as needed; however, when the new limit is less than the applicable number of secure addresses, you must reduce the number of secure MAC addresses first.

Security violations and actions

Port security triggers security violations when either of the following events occurs:

- **MAC Count Violation**

  Ingress traffic arrives at an interface from a nonsecure MAC address, and learning the address would exceed the applicable maximum number of secure MAC addresses.

  When an interface has both a VLAN maximum and an interface maximum configured, a violation occurs when either maximum is exceeded. For example, consider the following on a single interface configured with port security:

  - VLAN 1 has a maximum of five addresses
  - The interface has a maximum of ten addresses

  The device detects a violation when any of the following occurs:

  - The device has learned five addresses for VLAN 1, and inbound traffic from a sixth address arrives at the interface in VLAN 1.
  - The device has learned ten addresses on the interface, and inbound traffic from an eleventh address arrives at the interface.

The possible actions that the device can take are as follows:

- **Shutdown**

  Shuts down the interface that received the packet triggering the violation. The interface is error disabled. This action is the default. After you reenable the interface, it retains its port security configuration, including its secure MAC addresses.

  You can use the **errdisable** global configuration command to configure the device to reenable the interface automatically if a shutdown occurs, or you can manually reenable the interface by entering the **shutdown** and **no shutdown** interface configuration commands.

- **Restrict**

  Drops ingress traffic from any nonsecure MAC addresses.

The device keeps a count of the number of dropped MAC addresses, which is called the security violation count. Address learning continues until the maximum security violations have occurred on the interface. Traffic from addresses learned after the first security violation is dropped.

- MAC Move Violation

  Ingress traffic from a secure MAC address arrives at a different interface in the same VLAN as the interface on which the address is secured.

  You see a mac move notification only when the the logging level of Layer2 Forwarding Module (L2FM) is increased to 4 or 5

  When a MAC move violation occurs, the device increments the security violation counter for the interface, and irrespective of the violation mode configured, the interface is error disabled. If the violation mode is configured as Restrict or Protect, the violation is logged in the system log.

  Because a MAC move violation results in the interface being error disabled, irrespective of the violation mode configured, we recommend using the **errdisable** command to enable automatic errdisable recovery.

Port security and port types

You can configure port security only on Layer 2 interfaces. Details about port security and different types of interfaces or ports are as follows:

- **Access Ports**

  You can configure port security on interfaces that you have configured as Layer 2 access ports. On an access port, port security applies only to the access VLAN. VLAN maximums are not useful for access ports.

- **Trunk Ports**

  You can configure port security on interfaces that you have configured as Layer 2 trunk ports. The device allows VLAN maximums only for VLANs associated with the trunk port.

- **SPAN Ports**

  You can configure port security on SPAN source ports but not on SPAN destination ports.

- **Ethernet Port Channels**

  You can configure port security on Layer 2 Ethernet port channels in either access mode or trunk mode.



Note Port security is supported for FEX interfaces only in non-vPC deployments on Cisco Nexus 9300-EX/FX/FX2/FX3 Series switches. Beginning with Cisco NX-OS Release 9.3(5), Nexus 9300-FX3 Series switches are supported.

Port security and Port-Channel interfaces

Port security is supported on Layer 2 port-channel interfaces. Port security operates on port-channel interfaces in the same manner as on physical interfaces, except as described in this section.

- **General Guidelines**

Port security on a port-channel interface operates in either access mode or trunk mode. In trunk mode, the MAC address restrictions enforced by port security apply to all member ports on a per-VLAN basis.

Enabling port security on a port-channel interface does not affect port-channel load balancing.

Port security does not apply to port-channel control traffic passing through the port-channel interface. Port security allows port-channel control packets to pass without causing security violations. Port-channel control traffic includes the following protocols:

- Port Aggregation Protocol (PAgP)
- Link Aggregation Control Protocol (LACP)
- Inter-Switch Link (ISL)
- IEEE 802.1Q

- **Configuring Secure Member Ports**

  The port security configuration of a port-channel interface has no effect on the port security configuration of member ports.

- **Adding a Member Port**

  If you add a secure interface as a member port of a port-channel interface, the device discards all dynamic secure addresses learned on the member port but retains all other port-security configuration of the member port in the running configuration. Static secure MAC addresses learned on the secure member port are also stored in the running configuration rather than NVRAM.

  If port security is enabled on the member port and not enabled on the port-channel interface, the device warns you when you attempt to add the member port to the port-channel interface. You can use the **force** keyword with the **channel-group** command to forcibly add a secure member port to a nonsecure port-channel interface.

  While a port is a member of a port-channel interface, you cannot configure port security on the member port. To do so, you must first remove the member port from the port-channel interface.

- **Removing a Member Port**

  If you remove a member port from a port-channel interface, the device restores the port security configuration of the member port. Static secure MAC addresses that were learned on the port before you added it to the port-channel interface are restored to NVRAM and removed from the running configuration.



Note To ensure that all ports are secure as needed after you remove a port-channel interface, we recommend that you closely inspect the port-security configuration of all member ports.

- **Removing a Port-Channel Interface**

  If you remove a secure port-channel interface, the following occurs:

  - The device discards all secure MAC addresses learned for the port-channel interface, including static secure MAC addresses learned on the port-channel interface.
  - The device restores the port-security configuration of each member port. The static secure MAC addresses that were learned on member ports before you added them to the port-channel interface

are restored to NVRAM and removed from the running configuration. If a member port did not have port security enabled prior to joining the port-channel interface, port security is not enabled on the member port after the port-channel interface is removed.

Note To ensure that all ports are secure as needed after you remove a port-channel interface, we recommend that you closely inspect the port-security configuration of all member ports.

- **Disabling Port Security**

  If port security is enabled on any member port, the device does not allow you to disable port security on the port-channel interface. To do so, remove all secure member ports from the port-channel interface first. After disabling port security on a member port, you can add it to the port-channel interface again, as needed.

Port type changes

When you have configured port security on a Layer 2 interface and you change the port type of the interface, the device behaves as follows:

- **Access Port to Trunk Port**

  When you change a Layer 2 interface from an access port to a trunk port, the device drops all secure addresses learned by the dynamic method. The device moves the addresses learned by the static method to the native trunk VLAN.

- **Switched Port to Routed Port**

  When you change an interface from a Layer 2 interface to a Layer 3 interface, the device disables port security on the interface and discards all port security configuration for the interface. The device also discards all secure MAC addresses for the interface, regardless of the method used to learn the address.

- **Routed Port to Switched Port**

  When you change an interface from a Layer 3 interface to a Layer 2 interface, the device has no port security configuration for the interface.

Prerequisites for port security

Use this reference to review prerequisites for port security.

Port security has the following prerequisites:

- You must globally enable port security for the device that you want to protect with port security.

Default settings for port security

Use this reference to review the default settings for port security.

Table 89: Default settings for port security parameters

Parameters	Default
Port security enablement globally	Disabled
Port security enablement per interface	Disabled
MAC address learning method	Dynamic
Interface maximum number of secure MAC addresses	1
Security violation action	Shutdown

Guidelines and limitations for port security

Use this reference to review the guidelines and limitations for port security.

Port security configuration guidance

When configuring port security, follow these guidelines:

Feature support and interface limitations

- Port security does not support switched port analyzer (SPAN) destination ports.
- Port security does not depend upon other features.
- Port security is supported for FEX interfaces only in non-vPC deployments on Cisco Nexus 9300-EX Series switches.

Release-specific USB and VLAN limitations

- Beginning with Cisco NX-OS Release 10.2(1)F, disabling the USB Port is supported on Cisco NX-OS switches. To disable or enable the USB ports, use the **[no] port usb disable** command.
- After configuring the association between the primary and secondary VLANs and deleting the association, all static MAC addresses that were created on the primary VLANs remain on the primary VLAN only.

Note In some cases, the configuration is accepted with no error messages, but the commands have no effect.

After configuring the association between the primary and secondary VLANs:

- Static MAC addresses for the secondary VLANs cannot be created.
- Dynamic MAC addresses that learned the secondary VLANs are aged out.

Guidelines and limitations for port security on vPCs

Use this reference to review the guidelines and limitations for port security on vPCs.

Port security on vPC support

Apart from the guidelines and limitations for port security, check that you can meet the following guidelines and limitations for port security on vPCs:

vPC peer configuration guidelines

- Port security is not supported on FEX interfaces in vPC deployments.
- You must enable port security globally on both vPC peers in a vPC domain.
- You must enable port security on the vPC interfaces of both vPC peers.
- You must configure a static secure MAC address on the primary vPC peer. The static MAC address is synchronized with the secondary vPC peer. You can also configure a static secure MAC address on the secondary peer. The second static MAC address appears in the secondary vPC configuration but does not take affect.
- You must ensure that the maximum MAC count value remains the same for both primary and secondary vPC ports.
- On a secondary vPC port, there is no limit check for static MACs configured. Cisco recommends that you configure the same number of static MACs on a secondary vPC port as defined in the maximum MAC count.
- All learned MAC addresses are synchronized between vPC peers.
- Both vPC peers can be configured using the dynamic or static MAC address learning method. Cisco recommends that you configure both vPC peers using the same method. This helps prevent port shut down (errDisabled state) in certain cases, such as a vPC role change.
- Dynamic MAC addresses are dropped only after the age limit is reached on both vPC peers.

Primary vPC switch enforcement and ISSU limitations

- You set the maximum number of secure MAC addresses on the primary vPC switch. The primary vPC switch does the count validation and disregards any maximum number settings on the secondary switch.
- You must configure the violation action on the primary vPC. When a security violation is triggered, the security action defined on the primary vPC switch occurs.
- You can use the **show vpc consistency-parameters id** command to verify that the configuration is correct on both vPC peers.
- While a switch undergoes an in-service software upgrade (ISSU), port security operations are stopped on its peer switch. The peer switch does not learn any new MAC addresses, and MAC moves occurring during this operation are ignored. When the ISSU is complete, the peer switch is notified and normal port security functionality resumes.
- ISSU to higher versions is supported; however, ISSU to lower versions is not supported.

Configuring port security

Explains configuring port security.

Enable or disable port security globally

You can enable or disable port security globally on a device. By default, port security is disabled globally.

When you disable port security, all port security configuration on the interface is ineffective. When you disable port security globally, all port security configuration is lost.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **feature port-security**

Example:

```
switch(config)#feature port-security
```

Enables port security globally. The **no** option disables port security globally.

Step 3 (Optional) **show port-security**

Example:

```
switch#show port-security
```

Displays the status of port security.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable port security on a layer 2 interface

You can enable or disable port security on a Layer 2 interface. By default, port security is disabled on all interfaces.

When you disable port security on an interface, all switchport port security configuration for the interface is lost.

Before you begin

You must have enabled port security globally.

If a Layer 2 Ethernet interface is a member of a port-channel interface, you cannot enable or disable port security on the Layer 2 Ethernet interface.

If any member port of a secure Layer 2 port-channel interface has port security enabled, you cannot disable port security for the port-channel interface unless you first remove all secure member ports from the port-channel interface.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot*/*port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode for the Ethernet or port-channel interface that you want to configure with port security.

Step 3 **switchport**

Example:

```
switch(config-if)#switchport
```

Configures the interface as a Layer 2 interface.

Step 4 [**no**] **switchport port-security**

Example:

```
switch(config-if)#switchport port-security
```

Enables port security on the interface. The **no** option disables port security on the interface.

Step 5 (Optional) **show running-config port-security**

Example:

```
switch#show running-config port-security
```

Displays the port security configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable sticky MAC address learning

You can disable or enable sticky MAC address learning on an interface. If you disable sticky learning, the device returns to dynamic MAC address learning on the interface, which is the default learning method.

By default, sticky MAC address learning is disabled.

Before you begin

You must have enabled port security globally.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot/port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode for the interface that you want to configure with sticky MAC address learning.

Step 3 **switchport**

Example:

```
switch(config-if)#switchport
```

Configures the interface as a Layer 2 interface.

Step 4 [**no**] **switchport port-security mac-address sticky**

Example:

```
switch(config-if)#switchport port-security mac-address sticky
```

Enables sticky MAC address learning on the interface. The **no** option disables sticky MAC address learning.

Step 5 (Optional) **show running-config port-security**

Example:

```
switch#show running-config port-security
```

Displays the port security configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Add a static secure MAC address on an interface

You can add a static secure MAC address on a Layer 2 interface.

Note If the MAC address is a secure MAC address on any interface, you cannot add it as a static secure MAC address to another interface until you remove it from the interface on which it is already a secure MAC address.

By default, no static secure MAC addresses are configured on an interface.

Before you begin

- You must have enabled port security globally.
- Verify that the interface maximum has not been reached for secure MAC addresses. If needed, you can remove a secure MAC address, or you can change the maximum number of addresses on the interface.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot/port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode for the interface that you specify.

Step 3 [**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*]

Example:

```
switch(config-if)#switchport port-security mac-address 0019.D2D0.00AE
```

Configures a static MAC address for port security on the current interface. Use the **vlan** keyword if you want to specify the VLAN that traffic from the address is allowed on.

Step 4 (Optional) **show running-config port-security**

Example:

```
switch#show running-config port-security
```

Displays the port security configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Remove a static secure MAC address on an interface

You can remove a static secure MAC address on a Layer 2 interface.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot/port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode for the interface from which you want to remove a static secure MAC address.

Step 3 **no switchport port-security mac-address** *address*

Example:

```
switch(config-if)#no switchport port-security mac-address 0019.D2D0.00AE
```

Removes the static secure MAC address from port security on the current interface.

Step 4 (Optional) **show running-config port-security**

Example:

```
switch#show running-config port-security
```

Displays the port security configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Remove a sticky secure MAC address

You can remove a sticky secure MAC address, which requires that you temporarily disable sticky address learning on the interface that has the address that you want to remove.

Before you begin

You must have enabled port security globally.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot*/*port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode for the interface from which you want to remove a sticky secure MAC address.

Step 3 **no switchport port-security mac-address sticky**

Example:

```
switch(config-if)#no switchport port-security mac-address sticky
```

Disables sticky MAC address learning on the interface, which converts any sticky secure MAC addresses on the interface to dynamic secure MAC addresses.

Step 4 **clear port-security dynamic address** *address*

Example:

```
switch(config-if)#clear port-security dynamic address 0019.D2D0.02GD
```

Removes the dynamic secure MAC address that you specify.

Step 5 (Optional) **show port-security address interface** {**ethernet** *slot / port* | **port-channel** *channel-number*}

Example:

```
switch#show port-security address interface ethernet 2/1
```

Displays secure MAC addresses. The address that you removed should not appear.

Step 6 (Optional) **switchport port-security mac-address sticky**

Example:

```
switch(config-if)#switchport port-security mac-address sticky
```

Enables sticky MAC address learning again on the interface.

Remove a dynamic secure MAC address

You can remove dynamically learned, secure MAC addresses.

Before you begin

You must have enabled port security globally.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **clear port-security dynamic**{**interface ethernet** *slot*/*port* | **address** *address*} [**vlan** *vlan-ID*]

Example:

```
switch(config)#clear port-security dynamic interface ethernet 2/1
```

Removes dynamically learned, secure MAC addresses, as specified.

If you use the **interface** keyword, you remove all dynamically learned addresses on the interface that you specify.

If you use the **address** keyword, you remove the single, dynamically learned address that you specify.

Use the **vlan** keyword if you want to further limit the command to removing an address or addresses on a particular VLAN.

Step 3 (Optional) **show port-security address**

Example:

```
switch#show port-security address
```

Displays secure MAC addresses.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure a maximum number of MAC addresses

You can configure the maximum number of MAC addresses that can be learned or statically configured on a Layer 2 interface. You can also configure a maximum number of MAC addresses per VLAN on a Layer 2 interface. The largest maximum number of addresses that you can configure on an interface is 1025 addresses. The system maximum number of addresses is 8192.

By default, an interface has a maximum of one secure MAC address. VLANs have no default maximum number of secure MAC addresses.

Note When you specify a maximum number of addresses that is less than the number of addresses already learned or statically configured on the interface, the device rejects the command. To remove all addresses learned by the dynamic method, use the **shutdown** and **no shutdown** commands to restart the interface.

Before you begin

You must have enabled port security globally.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot*/*port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode, where *slot* is the interface that you want to configure with the maximum number of MAC addresses.

Step 3 [**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*]

Example:

```
switch(config-if)#switchport port-security maximum 425
```

Configures the maximum number of MAC addresses that can be learned or statically configured for the current interface. The highest valid *number* is 1025. The **no** option resets the maximum number of MAC addresses to the default, which is 1.

If you want to specify the VLAN that the maximum applies to, use the **vlan** keyword.

Step 4 (Optional) **show running-config port-security**

Example:

```
switch#show running-config port-security
```

Displays the port security configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure an address aging type and time

You can configure the MAC address aging type and the length of time that the device uses to determine when MAC addresses learned by the dynamic method have reached their age limit.

Absolute aging is the default aging type.

By default, the aging time is 0 minutes, which disables aging.

Before you begin

You must have enabled port security globally.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot/port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode for the interface that you want to configure with the MAC aging type and time.

Step 3 [**no**] **switchport port-security aging type**{**absolute** | **inactivity**}

Example:

```
switch(config-if)#switchport port-security aging type inactivity
```

Configures the type of aging that the device applies to dynamically learned MAC addresses. The **no** option resets the aging type to the default, which is absolute aging.

Step 4 [**no**] **switchport port-security aging time** *minutes*

Example:

```
switch(config-if)#switchport port-security aging time 120
```

Configures the number of minutes that a dynamically learned MAC address must age before the device drops the address. The maximum valid *minutes* is 1440. The **no** option resets the aging time to the default, which is 0 minutes (no aging).

Note
For Cisco Nexus 9200 and 9300-EX Series switches, up to 2 minutes might be added to the configured aging time. For example, if you set the aging time to 10 minutes, the age out occurs between 10 and 12 minutes after traffic stops.

Step 5 (Optional) **show running-config port-security**

Example:

```
switch#show running-config port-security
```

Displays the port security configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure a security violation action

You can configure the action that the device takes if a security violation occurs. The violation action is configurable on each interface that you enable with port security.

The default security action is to shut down the port on which the security violation occurs.

Before you begin

You must have enabled port security globally.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Enter one of the following commands:

- **interface ethernet** *slot/port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode for the interface that you want to configure with a security violation action.

Step 3 [**no**] **switchport port-security violation** {**protect** | **restrict** | **shutdown**}

Example:

```
switch(config-if)#switchport port-security violation restrict
```

Configures the security violation action for port security on the current interface. The **no** option resets the violation action to the default, which is to shut down the interface.

Step 4 (Optional) **show running-config port-security**

Example:

```
switch#show running-config port-security
```

Displays the port security configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Port security configuration verification commands

Use this reference to verify the Port security configuration.

To display the port security configuration information, perform one of the following tasks

Table 90: Port security configuration verification commands

Command	Purpose
show running-config port-security	Displays the port security configuration.
show port-security	Displays the port security status of the device.
show port-security interface	Displays the port security status of a specific interface.

Command	Purpose
show port-security address	Displays secure MAC addresses.
show vpc consistency-parameters vpc id	Verifies configuration on both vPC peers.

Displaying secure MAC addresses

Use this reference to display secure MAC addresses.

Secure MAC address display command

Use the **show port-security address** command to display secure MAC addresses.

Configuration example for port security

Use this reference to review configuration example for port security.

Port security trunk interface example

The following example shows a port security configuration for the Ethernet 2/1 interface with VLAN and interface maximums for secure addresses. In this example, the interface is a trunk port. Additionally, the violation action is set to Restrict.

```
feature port-security
interface Ethernet 2/1
  switchport
  switchport port-security
  switchport port-security maximum 10
  switchport port-security maximum 7 vlan 10
  switchport port-security maximum 3 vlan 20
  switchport port-security violation restrict
```

Configuration examples for port security in a vPC domain

Use this reference to review configuration examples for port security in a vPC domain.

The following example shows how to enable and configure port security on vPC peers in a vPC domain. The first switch is the primary vPC peer and the second switch is the secondary vPC peer. Before configuring port security on the switches, create the vPC domain and check that the vPC peer-link adjacency is established.

Example: configuring port security on an orphan port

Use this reference to review example: configuring port security on an orphan port.

Configuring port security on an orphan port command examples

```
primary_switch#configure terminal
primary_switch(config)#feature port-security
primary_switch(config)#interface ethernet 1/1
```

```
primary_switch(config-if)#switchport port-security
primary_switch(config-if)#switchport port-security maximum 1025
primary_switch(config-if)#switchport port-security violation restrict
primary_switch(config-if)#switchport port-security aging time 4
primary_switch(config-if)#switchport port-security aging type absolute
primary_switch(config-if)#switchport port-security mac sticky
primary_switch(config-if)#switchport port-security mac-address 0.0.1 vlan 101
primary_switch(config-if)#switchport port-security mac-address 0.0.2 vlan 101
primary_switch(config-if)#exit
primary_switch(config)#exit
primary_switch#copy running-config startup-config
secondary_switch#configure terminal
secondary_switch(config)#feature port-security
secondary_switch(config)#interface ethernet 3/1
secondary_switch(config-if)#switchport port-security
secondary_switch(config-if)#switchport port-security maximum 1025
secondary_switch(config-if)#switchport port-security violation restrict
secondary_switch(config-if)#switchport port-security aging time 4
secondary_switch(config-if)#switchport port-security aging type absolute
secondary_switch(config-if)#switchport port-security mac sticky
secondary_switch(config-if)#switchport port-security mac-address 0.0.1 vlan 101
secondary_switch(config-if)#switchport port-security mac-address 0.0.2 vlan 101
secondary_switch(config-if)#exit
secondary_switch(config)#exit
secondary_switch#copy running-config startup-config
```

Example: configuring port security on the vPC leg

Use this reference to review example: configuring port security on the vPC leg.

Configuring port security on the vPC leg command examples

```
primary_switch#configure terminal
primary_switch(config)#feature port-security
primary_switch(config)#interface port-channel 10
primary_switch(config-if)#switchport port-security
primary_switch(config-if)#switchport port-security maximum 1025
primary_switch(config-if)#switchport port-security violation restrict
primary_switch(config-if)#switchport port-security aging time 4
primary_switch(config-if)#switchport port-security aging type absolute
primary_switch(config-if)#switchport port-security mac sticky
primary_switch(config-if)#switchport port-security mac-address 0.0.1 vlan 101
primary_switch(config-if)#switchport port-security mac-address 0.0.2 vlan 101
primary_switch(config-if)#vpc 10
primary_switch(config-if)#exit
primary_switch(config)#exit
primary_switch#copy running-config startup-config
secondary_switch#configure terminal
secondary_switch(config)#feature port-security
secondary_switch(config)#interface port-channel 10
secondary_switch(config-if)#switchport port-security
secondary_switch(config-if)#switchport port-security maximum 1025
secondary_switch(config-if)#switchport port-security violation restrict
secondary_switch(config-if)#switchport port-security aging time 4
secondary_switch(config-if)#switchport port-security aging type absolute
secondary_switch(config-if)#switchport port-security mac sticky
secondary_switch(config-if)#switchport port-security mac-address 0.0.1 vlan 101
secondary_switch(config-if)#switchport port-security mac-address 0.0.2 vlan 101
secondary_switch(config-if)#vpc 10
secondary_switch(config-if)#exit
```

```
secondary_switch(config)#exit
secondary_switch#copy running-config startup-config
```

Additional references for port security

Use this reference to find additional references for port security.

Related Documents

Table 91: Related documents

Related Topic	Document Title
Layer 2 switching	*Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide*

MIBs

Cisco NX-OS provides read-only SNMP support for port security.

Table 92: MIBs

MIBs	MIBs Link
CISCO-PORT-SECURITY-MIB **Note** Traps are supported for notification of secure MAC address violations.	To locate and download MIBs, go to the following URL: https://snmp.cloudapps.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2

Port security support for VXLAN EVPN

This section describes how to configure Port Security for VXLAN EVPN.

Guidelines and limitations for port security support for VXLAN EVPN

Use this reference to review the guidelines and limitations for port security support for VXLAN EVPN.

- Beginning with Cisco NX-OS Release 10.3(3)F, the L2 port security feature is supported on VXLAN BGP EVPN (single VTEP) for Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, 9408, 9332C, 9364C switches and Cisco Nexus 9500 switches with 9700-EX/ FX/GX line cards with the following limitations:
  - Only a Single VTEP solution is supported. However, secure MAC mobility is not supported on a VXLAN environment.
- Beginning with Cisco NX-OS Release 10.3(3)F, IPv6 underlay is supported on port security (single VTEP) for VXLAN EVPN on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 switches and Cisco Nexus 9500 switches with 9700-EX/ FX/GX line cards.

- Port-security is not supported with Fabric Peering.
- If the L2 Port Security feature is enabled, the following behavior is observed:
  - Secure MACs will be sent as static MACs and will be seen as static MACs on remote VTEPs. Hence, if there is any attempt to learn the secure MAC on a remote VTEP as a dynamic MAC (due to the malicious host with the same MAC), it will be prevented.
  - If **restrict** option is set on the violated MACs, then these violated MACs will be sent using the **static drop** set. On remote VTEPs, the MACs will be configured with the **static drop** so that any attempt to send traffic to these hosts from the remote VTEPs will be dropped at the remote VTEP itself.
  - Both the local static and secure MAC is advertised to fabric with a sticky bit, so for a remote VTEP there is no difference if the remote static MAC is from a VTEP for secure or static MAC.
  - If local static exists already, that will take precedence over the remote static (either it is from secure or static).
  - There might be multiple updates for a MAC learned on a secure port from local VTEP to fabric based on the security decision made locally for the MAC, however, the final security behavior for the MAC will be consistent for the local and remote VTEP.
  - You can specify the **inactivity** value for a secure MAC. If there is no activity, then the secure MAC will be removed, and the secure MAC host can move to another port.

Port security support for VXLAN EVPN verification commands

Use this reference to verify the Port security support for VXLAN EVPN.

To display the Port Security support for VXLAN EVPN configuration information, enter one of the following commands:

Table 93: Port security support for VXLAN EVPN verification commands

Command	Purpose
show running-config interface *interface-name*	Displays running configuration of interface.
show port-security	Displays port security configuration information.

Example of show running-config interface command

```
switch#show running-config inter e1/48
!Command: show running-config interface Ethernet1/48
!Running configuration last done at: Thu Feb 16 08:39:43 2023
!Time: Fri Feb 17 06:07:33 2023

version 10.3(3) Bios:version 01.08

interface Ethernet1/48
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 200
  spanning-tree port type edge trunk
  switchport port-security maximum 1025
```

```
  switchport port-security
  no shutdown

LVTEP:
------
switch#show mac address-table inter e1/48
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan,
(NA)- Not Applicable
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*  101     0012.0100.0001   secure   -         T      F    Eth1/48
*  101     0012.0100.0002   secure   -         T      F    Eth1/48
*  101     0012.0100.0003   secure   -         T      F    Eth1/48
*  101     0012.0100.0004   secure   -         T      F    Eth1/48
```

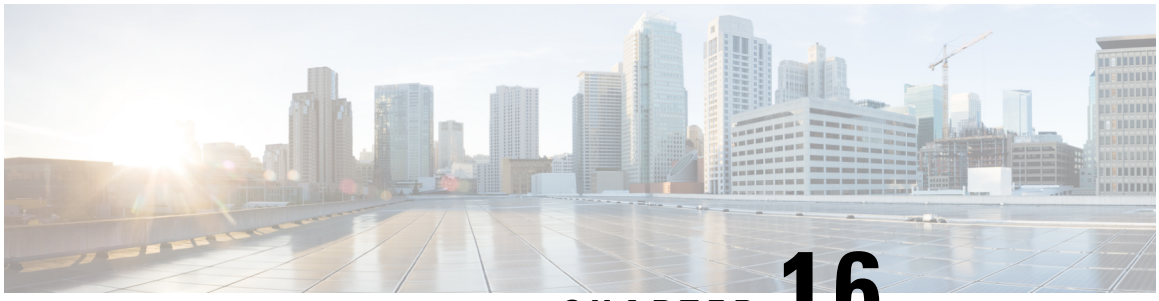
Example of show port-security command

```
switch#show port-security
Total Secured Mac Addresses in System (excluding one mac per port)     : 1024
Max Addresses limit in System (excluding one mac per port) : 7168

-------------------------------------------------------------------------------
Secure Port    MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
(Count)       (Count)          (Count)
-------------------------------------------------------------------------------
Ethernet1/48          1025         1025            0              Shutdown

switch#show port-security address interface e1/48
-------------------------------------------------------------------------
Secure Mac Address Table
-------------------------------------------------------------------------
Vlan    Mac Address    Type      Remaining  Remotely Remotely Ports
age        learnt   aged
(mins)              out
----    -----------   ------     ------     -------  -----    ----
 101   0012.0100.0001   DYNAMIC  0          No       No     Ethernet1/48
 101   0012.0100.0002   DYNAMIC  0          No       No     Ethernet1/48
 101   0012.0100.0003   DYNAMIC  0          No       No     Ethernet1/48
 101   0012.0100.0004   DYNAMIC  0          No       No     Ethernet1/48


RVTEP:
------
switch#show mac address-table
C  101     0012.0100.0001   static   -         F      F    nve1(2001:db8::1)
C  101     0012.0100.0002   static   -         F      F    nve1(2001:db8::1)
C  101     0012.0100.0003   static   -         F      F    nve1(2001:db8::1)
C  101     0012.0100.0004   static   -         F      F    nve1(2001:db8::1)
```



CHAPTER 16

Configuring DHCP

This chapter describes how to configure the Dynamic Host Configuration Protocol (DHCP) on a Cisco NX-OS device.

This chapter includes the following sections:

- About DHCP snooping, on page 441
- About the DHCP Relay Agent, on page 447
- About the DHCPv6 Relay Agent, on page 451
- DHCPv6 Smart Relay Agent, on page 452
- Guidelines and Limitations for DHCPv6 Smart Relay, on page 453
- About DHCP client, on page 453
- Prerequisites for DHCP, on page 453
- Guidelines and limitations for DHCP, on page 453
- Default settings for DHCP, on page 455
- Configuring DHCP, on page 456
- Configuring DHCPv6, on page 478
- Enable the DHCP client, on page 486
- Configuring UDP relay, on page 488
- DHCP configuration verification commands, on page 492
- Displaying IPv6 RA guard statistics, on page 494
- Displaying DHCP snooping bindings, on page 494
- Clearing the DHCP snooping binding database, on page 495
- Monitoring DHCP, on page 495
- Clearing DHCP snooping statistics, on page 496
- Clearing DHCP relay statistics, on page 496
- Clearing DHCPv6 relay statistics, on page 496
- Clearing DHCPv6-PD binding, on page 497
- Configuration examples for DHCP, on page 497
- Configuration examples for DHCP client, on page 498
- Additional references for DHCP, on page 499

About DHCP snooping

DHCP snooping is a security feature that acts as a firewall between untrusted hosts and trusted DHCP servers.

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Uses the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

Enabling and scope of DHCP snooping

DHCP snooping can be enabled globally and on a per-VLAN basis. By default, the feature is disabled globally and on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

Trusted and untrusted sources

You can configure whether DHCP snooping trusts traffic sources. An untrusted source may initiate traffic attacks or other hostile actions. To prevent such attacks, DHCP snooping filters messages from untrusted sources.

In an enterprise network, a trusted source is a device that is under your administrative control. These devices include the switches, routers, and servers in the network. Any device beyond the firewall or outside the network is an untrusted source. Generally, host ports are treated as untrusted sources.

In a service provider environment, any device that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In the Cisco NX-OS device, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to devices (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.



Note For DHCP snooping to function properly, all DHCP servers must be connected to the device through trusted interfaces.

DHCP snooping binding database

Using information extracted from intercepted DHCP messages, DHCP snooping dynamically builds and maintains a database. The database contains an entry for each untrusted host with a leased IP address if the host is associated with a VLAN that has DHCP snooping enabled. The database does not contain entries for hosts connected through trusted interfaces.



Note The DHCP snooping binding database is also referred to as the DHCP snooping binding table.

DHCP snooping updates the database when the device receives specific DHCP messages. For example, the feature adds an entry to the database when the device receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the device receives a DHCPRELEASE message from the host.

Each entry in the DHCP snooping binding database includes the MAC address of the host, the leased IP address, the lease time, the binding type, and the VLAN number and interface information associated with the host.

Dynamic ARP inspection (DAI) and IP Source Guard also use information stored in the DHCP snooping binding database.

You can remove entries from the binding database by using the **clear ip dhcp snooping binding** command.

DHCP snooping in a vPC environment

A virtual port channel (vPC) allows two Cisco NX-OS switches to appear as a single logical port channel to a third device. The third device can be a switch, a server, or any other networking device that supports port channels.

In a typical vPC environment, DHCP requests can reach one vPC peer switch, and the responses can reach the other vPC peer switch, resulting in a partial DHCP (IP-MAC) binding entry in one switch and no binding entry in the other switch. As a result, DHCP snooping and associated features such as dynamic ARP inspection (DAI) and IP Source Guard are disrupted. This issue is addressed by using Cisco Fabric Service over Ethernet (CFSoE) distribution to ensure that all DHCP packets (requests and responses) appear on both switches, which helps in creating and maintaining the same binding entry on both switches for all clients behind the vPC link.

CFSoE distribution also allows only one switch to forward the DHCP requests and responses on the vPC link. In non-vPC environments, both switches forward the DHCP packets.

Synchronizing DHCP snooping binding entries

The dynamic DHCP binding entries should be synchronized in the following scenarios:

- When the remote vPC is online, all the binding entries for that vPC link should be synchronized with the peer.
- When DHCP snooping is enabled on the peer switch, the dynamic binding entries for all vPC links should be synchronized with the peer.

Packet validation

The device validates DHCP packets received on the untrusted interfaces of VLANs that have DHCP snooping enabled. The device forwards the DHCP packet unless any of the following conditions occur (in which case, the packet is dropped):

- The device receives a DHCP response packet (such as a DHCPACK, DHCPNAK, or DHCPOFFER packet) on an untrusted interface.
- The device receives a packet on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match. This check is performed only if the DHCP snooping MAC address verification option is turned on.
- The device receives a DHCPRELEASE or DHCPDECLINE message from an untrusted host with an entry in the DHCP snooping binding table, and the interface information in the binding table does not match the interface on which the message was received.

In addition, you can enable strict validation of DHCP packets, which checks the options field of DHCP packets, including the “magic cookie” value in the first four bytes of the options field. By default, strict validation is

disabled. When you enable it, by using the **ip dhcp packet strict-validation** command, if DHCP snooping processes a packet that has an invalid options field, it drops the packet.

DHCP snooping option 82 data insertion

DHCP can centrally manage the IP address assignments for a large number of subscribers. When you enable Option 82, the device identifies a subscriber device that connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can connect to the same port on the access device and are uniquely identified.

When you enable Option 82 on the Cisco NX-OS device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the Cisco NX-OS device receives the DHCP request, it adds the Option 82 information in the packet. The Option 82 information contains the device MAC address (the remote ID suboption) and the port identifier vlan-ifindex (for non-vPCs) or vlan-vpcid (for vPCs), from which the packet is received (the circuit ID suboption).



Note For vPC peer switches, the remote ID suboption contains the vPC switch MAC address, which is unique in both switches. This MAC address is computed with the vPC domain ID. The Option 82 information is inserted at the switch where the DHCP request is first received before it is forwarded to the other vPC peer switch.

3. The device forwards the DHCP request that includes the Option 82 field to the DHCP server.
4. The DHCP server receives the packet. If the server is Option 82 capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the Option 82 field in the DHCP reply.
5. The DHCP server sends the reply to the Cisco NX-OS device. The Cisco NX-OS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The Cisco NX-OS device removes the Option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

If the previously described sequence of events occurs, the following values do not change:

- Circuit ID suboption fields
  - Suboption type
  - Length of the suboption type
  - Circuit ID type
  - Length of the circuit ID type
- Remote ID suboption fields
  - Suboption type
  - Length of the suboption type
  - Remote ID type

- Length of the circuit ID type

This figure shows the packet formats for the remote ID suboption and the circuit ID suboption. The Cisco NX-OS device uses the packet formats when you globally enable DHCP snooping and when you enable Option 82 data insertion and removal. For the circuit ID suboption, the module field is the slot number of the module.

Figure 10: Suboption packet formats

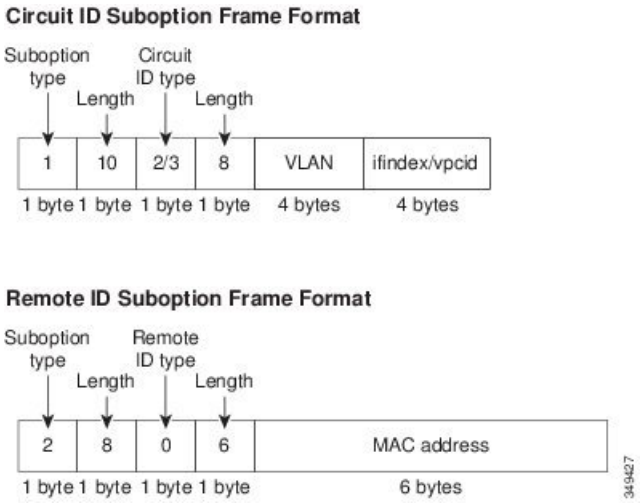


Figure 11: Suboption packet format (non TLV format)

About the DHCP Relay Agent

DHCP relay agent

The DHCP relay agent is a network device feature that forwards DHCP packets between clients and servers located on different subnets.

- Receives DHCP messages from clients and generates new DHCP messages to send on another interface.
- Sets the gateway address (giaddr field) in the DHCP packet and can add Option 82 (relay agent information option) if configured.
- Forwards replies from the DHCP server back to the client, removing Option 82 if present.

DHCP relay agent reference information

The DHCP relay agent is useful when DHCP clients and servers are not on the same physical subnet.



Note When the device relays a DHCP request that already includes Option 82 information, the device forwards the request with the original Option 82 information without altering it.

DHCP relay agent option 82

The DHCP relay agent Option 82 is a feature that enables a device to insert and remove Option 82 information in DHCP packets forwarded by the relay agent.

- Allows insertion of Option 82 information, including device MAC address (remote ID) and port identifier (circuit ID), into DHCP packets.
- Enables DHCP servers to use Option 82 data for IP address assignment and policy enforcement.
- Facilitates communication between DHCP clients and servers on different subnets by relaying and modifying DHCP messages.

How DHCP relay agent option 82 works

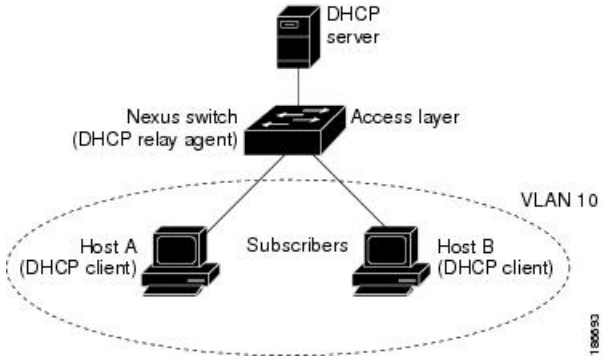
DHCP relay agent Option 82 enhances DHCP message exchange by appending identifying information to requests and managing packet forwarding between clients and servers on separate subnets.

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the Cisco NX-OS device receives the DHCP request, it adds the Option 82 information in the packet. The Option 82 information contains the device MAC address (the remote ID suboption) and the port identifier ifindex (for non-VXLAN VLANs) or vn-segment-id-mod-port (for VXLAN VLANs), from which the packet is received (the circuit ID suboption). In DHCP relay, the circuit ID is filled with the ifindex of the SVI or Layer 3 interface on which DHCP relay is configured.
3. The device adds the IP address of the relay agent to the DHCP packet.

4. The device forwards the DHCP request that includes the Option 82 field to the DHCP server.
5. The DHCP server receives the packet. If the server is Option 82 capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the Option 82 field in the DHCP reply.
6. The DHCP server unicasts the reply to the Cisco NX-OS device if the request was relayed to the server by the device. The Cisco NX-OS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The Cisco NX-OS device removes the Option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

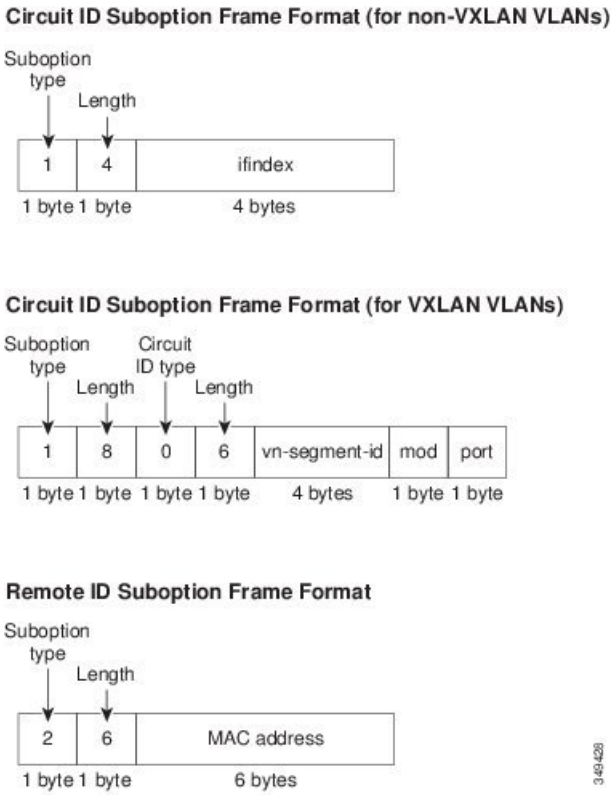
This figure shows an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the device at the access layer. Because the DHCP clients and their associated DHCP server do not reside on the same IP network or subnet, a DHCP relay agent is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.

Figure 12: DHCP relay agent in a metropolitan ethernet network



This figure shows the packet formats for the circuit ID suboption and the remote ID suboption.

Figure 13: Suboption packet formats



Example: DHCP relay agent option 82 in action

For example, in a metropolitan Ethernet network, a DHCP relay agent configured on a Cisco NX-OS device inserts Option 82 information into DHCP requests from clients. The DHCP server uses this information to assign IP addresses and enforce policies, then the relay agent removes the Option 82 data before forwarding the reply to the client.

VRF support for the DHCP relay agent

The VRF support for the DHCP relay agent allows DHCP broadcast messages from clients in one VRF to be forwarded to DHCP servers in another VRF, enabling centralized DHCP support and efficient IP address utilization.

- Enables a single DHCP server to serve clients across multiple VRFs using a shared IP address pool.
- Requires Option 82 to be enabled for the DHCP relay agent.
- Option 82 information includes VPN identifier, link selection, and server identifier override.

How VRF support works with the DHCP relay agent

You can configure the DHCP relay agent to forward DHCP broadcast messages from clients in a virtual routing and forwarding (VRF) instance to DHCP servers in a different VRF. By using a single DHCP server

to provide DHCP support to clients in multiple VRFs, you can conserve IP addresses by using a single IP address pool rather than one for each VRF. For general information about VRFs, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* .

- DHCP relay agent forwards client messages between VRFs.
- Centralized DHCP server supports multiple VRFs.
- Option 82 must be enabled for VRF support.

1. Enable Option 82 for the DHCP relay agent.
2. Configure DHCP relay address and VRF information on the interface.
3. Device inserts Option 82 information and forwards the request to the DHCP server in the server VRF.
4. Device strips Option 82 from the response and forwards it to the client in the client VRF.

The Option 82 information includes the following:

- **VPN identifier**: Name of the VRF that the interface receiving the DHCP request is a member of.
- **Link selection**: Subnet address of the interface receiving the DHCP request. When DHCP smart relay is enabled, this is the subnet of the active giaddr.
- **Server identifier override**: IP address of the interface receiving the DHCP request. When DHCP smart relay is enabled, this is the active giaddr.



Note The DHCP server must support the VPN identifier, link selection, and server identifier override options.

VRF support for DHCP relay agent

For example, if a DHCP request arrives on an interface configured with a DHCP relay address and VRF information, and the DHCP server address belongs to a network in a different VRF, the device inserts Option 82 information (including VPN identifier, link selection, and server identifier override) and forwards the request to the server in the server VRF. The response is then returned to the client in the client VRF after stripping Option 82.

DHCP smart relay agent

The DHCP smart relay agent is a network component that forwards DHCP broadcast requests and manages IP address allocation from available subnet pools.

- Forwards DHCP broadcast request packets from hosts to the DHCP server by setting the giaddr to the primary address of the inbound interface.
- Allocates IP addresses from the primary subnet pool until it is exhausted.
- Can be configured to allocate IP addresses from the secondary subnet pool if the primary pool is exhausted or ignored by the server.

Reference information for DHCP smart relay agent

The DHCP smart relay agent enhances address allocation flexibility, especially when the number of hosts exceeds the available addresses in the primary pool or when multiple subnets are configured on an interface.

Example of DHCP smart relay agent operation

For example, if a network interface has both primary and secondary IP subnets configured, and the primary subnet pool is exhausted, the DHCP smart relay agent can allocate addresses from the secondary subnet pool to accommodate additional hosts.

About the DHCPv6 Relay Agent

Dhcpv6 relay agent

The DHCPv6 relay agent is a network function that forwards DHCPv6 packets between clients and servers when they are not on the same physical subnet.

- Receives DHCPv6 messages from clients.
- Generates new DHCPv6 messages to send out on another interface.
- Sets the gateway address (giaddr field) and forwards messages to the DHCPv6 server.

Supporting reference information for DHCPv6 relay agent

The DHCPv6 relay agent is used when DHCPv6 clients and servers are located on different subnets, allowing communication between them.

VRF support for the DHCPv6 relay agent

The DHCPv6 relay agent with VRF support enables forwarding of DHCPv6 broadcast messages from clients in one VRF instance to DHCPv6 servers in another VRF instance.

- Allows a single DHCPv6 server to provide DHCP support to clients in multiple VRFs.
- Conserves IP addresses by using a single IP address pool for multiple VRFs.
- Eliminates the need for separate IP address pools for each VRF.

Reference information for VRF support in DHCPv6 relay agent

The DHCPv6 relay agent can be configured to forward messages between clients and servers across VRFs, optimizing IP address usage.

VRF support for DHCPv6 relay agent

For example, a network administrator can configure a single DHCPv6 server to assign addresses to clients in multiple VRFs, reducing the number of required IP address pools and simplifying management.

IPv6 availability for delegated prefix through the v6 relay agent

DHCPv6 Prefix Delegation (DHCPv6-PD) is a mechanism that assigns complete subnets and network parameters from a DHCPv6-PD server to a DHCPv6-PD client, extending the functionality of the DHCPv6 relay agent as defined in RFC3633.

- Enables assignment of subnets and network/interface parameters from server to client.
- Extends DHCPv6 relay agent functionality to support prefix delegation.
- Facilitates static route creation and distribution for delegated prefixes on the switch.

Details on IPv6 delegated prefix availability via v6 relay agent

DHCPv6-PD allows a DHCPv6 relay agent to forward network address requests from clients to a DHCPv6 server using an IANA option. If a client requires a prefix address, it includes an IAPD option in the request. The DHCPv6 server delegates the requested prefix if available in its pool.

- Prefix delegation does not interwork with First Hop Security (FHS).
- DHCPv6 does not support prefix delegation with relay chaining.

When the CLI is enabled, DHCPv6-PD adds a static route on the switch for the delegated prefix, making the prefix routable from the switch. A DHCPv6-PD binding is created for each client, along with an IPv6 route for the delegated prefix. The added static routes are distributed to neighbors through the OSPFv3 routing protocol.



Note Currently, DHCPv6-PD route distribution is not supported for other routing protocols like IS-IS or BGP. For more information, refer to the "Configuring Redistribution" section of *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* .

DHCPv6 Smart Relay Agent

The DHCPv6 Smart Relay Agent is a network component that forwards solicit packets from hosts to the DHCPv6 server, setting the link address to the inbound interface address.

- Forwards solicit packets from hosts to the DHCPv6 server.
- Sets the link address to the address of the inbound interface.
- Allocates IP addresses from the primary or secondary subnet pool based on availability.

How the DHCPv6 Smart Relay Agent Allocates IP Addresses

When the DHCPv6 smart relay agent receives solicit packets from a host, it sets the link address to the address of the inbound interface and forwards the packets to the server. The server allocates IP addresses from the link address subnet pool until the pool is exhausted and ignores further requests. If the first subnet pool is exhausted or the server ignores further requests, the relay agent can allocate IP addresses from the secondary subnet pool. This is useful when the number of hosts exceeds the available IP addresses or when multiple subnets are configured on an interface using secondary addresses.

Example: DHCPv6 Smart Relay Agent in Action

For example, if a network interface has both a primary and a secondary subnet configured, and the primary subnet pool is exhausted, the DHCPv6 smart relay agent can allocate IP addresses from the secondary subnet pool to ensure continued address assignment for new hosts.

Guidelines and Limitations for DHCPv6 Smart Relay

- In a vPC environment, it is recommended that the subnet of the IPv6 address(es) of an interface should be the same on both switches.
- The number of hosts that use DHCPv6 smart relay at an instance is restricted to 10,000.
- DHCPv6 Smart Relay is supported on cloud-based platforms.

About DHCP client

The DHCP client feature is a network capability that allows automatic assignment of IPv4 or IPv6 addresses to interfaces.

- Supports both IPv4 and IPv6 address assignment.
- Can be configured on routed ports, the management port, and switch virtual interfaces (SVIs).

Prerequisites for DHCP

Use this reference to review prerequisites for DHCP.

- You should be familiar with DHCP before you configure DHCP snooping or the DHCP relay agent.

Guidelines and limitations for DHCP

DHCP configuration on Cisco Nexus 9000 Series switches is subject to specific guidelines and limitations, including feature support and platform restrictions.

DHCP guidelines overview

- Key limitations include lack of support for relaying bootp packets, DHCP subnet broadcast, and certain features on specific line cards and platforms.
- Platform-specific support for DHCP snooping, relay, and client features varies by NX-OS release and hardware model.

DHCP snooping and relay guidelines

- For secure POAP, enable DHCP snooping and set firewall rules to block unintended or malicious DHCP servers.
- Cisco Nexus 9000 Series switches do not support relaying of bootp packets, but do support Layer 2 switched bootp packets.
- DHCP subnet broadcast is not supported.
- Enable Option 82 insertion for DHCP packets to support the highest DHCP snooping scale.
- Before globally enabling DHCP snooping, ensure that the DHCP server and relay agent are configured and enabled.
- DHCP snooping does not work when DHCP relay is configured on the same Cisco Nexus device.
- DHCP snooping, DAI, and FHS are not supported on Cisco Nexus 9500 platform switches with -R/-RX/-R2 line cards.
- DHCP snooping is not supported on VXLAN VLANs.

Interface, VRF, and DHCP smart relay guidelines

- DHCP snooping supports multiple IP addresses with the same MAC address and VLAN in static binding entries.
- VXLAN supports DHCP relay when the DHCP server is reachable through a default VRF.
- If a VLAN ACL (VACL) is configured on a VLAN with DHCP snooping, ensure the VACL permits DHCP traffic between servers and hosts. When both DHCP snooping and relay are enabled, relay takes precedence.
- If an ingress router ACL is configured on a Layer 3 interface with a DHCP server address, ensure the ACL permits DHCP traffic between servers and hosts.
- If using DHCP relay with clients and servers in different VRFs, use only one DHCP server within a VRF.
- ACL statistics are not supported if DHCP snooping is enabled.
- Synchronize DHCP configuration across switches in a vPC link to avoid run-time errors and dropped packets.
- DHCP Smart Relay is limited to the first 100 IP addresses of the interface on which it is enabled.
- Configure a helper address on the interface to use DHCP smart relay.

DHCPv6 and DHCP client guidelines

- If DHCP Smart Relay is enabled in a vPC environment, primary and secondary interface IP addresses should share a subnet between peers.
- When configuring DHCPv6 server addresses on an interface, a destination interface cannot be used with global IPv6 addresses.
- DHCPv6-PD Routes are not generated when a DHCPv6 client initiates a Rebind. Existing IAPD entries are refreshed but not created. For IAPD route creation, a full Solicit, Advertise, Request, Reply sequence must be seen by the DHCPv6 Relay agent.
- If using DHCP relay on an unnumbered interface, configure the switch to insert option 82.
- DHCPv6 Prefix Delegation Routes are not generated when Option 14 **Rapid Commit** is present. A full Solicit, Advertise, Request, Reply sequence is needed to generate an IAPD route.
- The following guidelines and limitations apply to the DHCP client feature:
  - You can configure multiple SVIs, but each interface VLAN should be in a different subnet. The DHCP client feature cannot configure different IP addresses with the same subnet on different interface VLANs on the same device.
  - DHCP client and DHCP relay are not supported on the same switch.
  - DHCP client is not supported for Layer 3 subinterfaces.
  - DHCP client is supported on Cisco Nexus 9300 and 9500 Series switches.
  - DHCP client is not supported on Cisco Nexus 9500 platform switches with N9K-X9636C-R, N9K-X9636C-RX, N9K-X9636Q-R, and N9K-X96136YC-R line cards.
- For DHCP configuration limits, see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* .

Supported platforms and releases

- Beginning with Cisco NX-OS Release 9.3(3), DHCP snooping and relay are supported on Cisco Nexus 9364C-GX, 9316D-GX, and 93600CD-GX switches.
- Beginning with Cisco NX-OS Release 10.3(1)F , DHCP relay is supported on Cisco Nexus X9836DM-A line card of the Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, DHCP relay is supported on Cisco Nexus X98900CD-A line card of Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, DHCP relay is supported on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

Default settings for DHCP

Use this reference to review the default settings for DHCP.

Default settings for DHCP information

This table lists the default settings for DHCP parameters.

Table 94: Default DHCP parameters

Parameters	Default
DHCP feature	Disabled
DHCP snooping	Disabled
DHCP snooping on VLANs	Disabled
DHCP snooping MAC address verification	Enabled
DHCP snooping Option 82 support	Disabled
DHCP snooping trust	Untrusted
DHCP relay agent	Enabled
DHCPv6 relay agent	Enabled
VRF support for the DHCP relay agent	Disabled
VRF support for the DHCPv6 relay agent	Disabled
DHCP Option 82 for relay agent	Disabled
DHCP smart relay agent	Disabled
DHCP server IP address	None

Configuring DHCP

Configure the minimum DHCP settings

Procedure

Step 1 Enable the DHCP feature.

When the DHCP feature is disabled, you cannot configure DHCP snooping.

Step 2 Enable DHCP snooping globally.

Step 3 Enable DHCP snooping on at least one VLAN.

By default, DHCP snooping is disabled on all VLANs.

Step 4 Make sure that the DHCP server is connected to the device using a trusted interface.

Step 5 (Optional) Enable the DHCP relay agent.

Step 6 (Optional) If DHCP servers and clients are in different VRFs, do the following:

a) Enable Option 82 for the DHCP relay agent.

b) Enable VRF support for the DHCP relay agent.

Step 7 (Optional) Configure an interface with the IP address of the DHCP server.

Enable or disable the DHCP feature

You can enable or disable the DHCP feature on the device. By default, DHCP is disabled.

When the DHCP feature is disabled, you cannot configure the DHCP relay agent, DHCP snooping, or any of the features that depend on DHCP. In addition, all DHCP configuration is removed from the device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **feature dhcp**

Example:

```
switch(config)#feature dhcp
```

Enables the DHCP feature. The **no** option disables the DHCP feature and erases all DHCP configuration.

Step 3 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configuring DHCP snooping

Enable or disable DHCP snooping globally

You can enable or disable DHCP snooping globally on the device.

Before you begin

Make sure that you have enabled the DHCP feature.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp snooping**

Example:

```
switch(config)#ip dhcp snooping
```

Enables DHCP snooping globally. The **no** form of this command disables DHCP snooping.

Step 3 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable DHCP snooping on a VLAN

You can enable or disable DHCP snooping on one or more VLANs. By default, DHCP snooping is disabled on all VLANs.

Before you begin

Make sure that the DHCP feature is enabled.

Note If a VACL is configured on a VLAN that you are configuring with DHCP snooping, make sure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp snooping vlan** *vlan-list*

Example:

```
switch(config)#ip dhcp snooping vlan 100,200,250-252
```

Enables DHCP snooping on the VLANs specified by *vlan-list* . The **no** form of this command disables DHCP snooping on the VLANs specified.

Step 3 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable DHCP snooping MAC address verification

You can enable or disable DHCP snooping MAC address verification. If the device receives a packet on an untrusted interface and the source MAC address and the DHCP client hardware address do not match, address verification causes the device to drop the packet. MAC address verification is enabled by default.

Before you begin

Make sure that the DHCP feature is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp snooping verify mac-address**

Example:

```
switch(config)#ip dhcp snooping verify mac-address
```

Enables DHCP snooping MAC address verification. The **no** form of this command disables MAC address verification.

Step 3 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable option 82 data insertion and removal

Use this procedure to enable or disable Option 82 data insertion and removal for DHCP packets on your device.

You can enable or disable the insertion and removal of Option 82 information for DHCP packets forwarded without the use of the DHCP relay agent. By default, the device does not include Option 82 information in DHCP packets.

Note DHCP relay agent support for Option 82 is configured separately.

Note To support a higher DHCP pps scale, you must enable the insertion of Option 82 information for DHCP packets.

Note You must add Option82 as specified in the format string in the command configuration.

- The length of the Option82 string increases based on the length of the format string.
- The circuit-id must include the ascii value of the format string.

Before you begin

Make sure that the DHCP feature is enabled.

Procedure

Step 1 [**no**] **ip dhcp snooping information option**

Example:

```
switch#configure terminal
switch(config)#ip dhcp snooping information option
```

Enables the insertion and removal of Option 82 information for DHCP packets. The **no** form of this command disables the insertion and removal of Option 82 information.

Step 2 (Optional) [**no**] **ip dhcp option82 sub-option circuit-id** *format_type string format*

Example:

```
switch(config)#ip dhcp snooping sub-option circuit-id format-type string format
```

Example:

```
switch(config)#ip dhcp snooping sub-option circuit-id format-type string format?
WORD Format string (Max Size 64)
```

Configures Option 82 as follows:

- If you do not specify *format-type* , the *circuit-id* displays the incoming port, for example, *ethernet1/1* .
- If you specify format <*word*> , the *circuit-id* displays the specified word
- If you specify *%h* instead of <*word*> , the *circuit-id* displays the host name.
- If you specify *%p* instead of <*word*> , the *circuit-id* displays the port name.
- If you specify *%h:%p* instead of <*word*> , the *circuit-id* displays both host and port name.

Note
The *no* option disables this behavior.

Step 3 (Optional) [**no**] **ip dhcp snooping sub-option format non-tlv**

Example:

```
switch(config)#ip dhcp snooping sub-option format non-tlv
```

Removes subtype, and its length, from Circuit ID and Remote ID suboptions of Option 82 information.

Step 4 **interface** *interface slot/port*

Example:

```
switch(config)#interface ethernet 2/2
switch(config-if)#
```

Enters the interface configuration mode, where slot/port is the interface where you want to enable or disable snooping.

Step 5 (Optional) **ip dhcp option82 sub-option circuit-id**

Example:

```
switch(config-if)#ip dhcp option82 sub-option circuit-id?WORD Format string (Max Size 64)
```

Example:

```
switch(config-if)#ip dhcp option82 sub-option circuit-id test
```

Configures Option 82 at the interface.

Note
This command is not supported at SVI and Sub-Interface.

Note
The *no* option disables this behavior

Step 6 **exit**

Example:

```
switch(config-if)#exit
switch(config)#exit
switch#
```

Exits interface configuration mode.

Step 7 (Optional) **show ip dhcp option82 info interface** *intf_name*

Displays the DHCP configuration. It shows whether option82 is enabled or disabled on that interface and the transmitted packets for an interface that is option82 enabled.

Step 8 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 9 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable strict DHCP packet validation

You can enable or disable the strict validation of DHCP packets. By default, strict validation of DHCP packets is disabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp packet strict-validation**

Example:

```
switch(config)#ip dhcp packet strict-validation
```

Enables the strict validation of DHCP packets. The **no** form of this command disables strict DHCP packet validation.

Step 3 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure an interface as trusted or untrusted

Configure an interface as trusted or untrusted to control DHCP message sources for DHCP snooping.

You can configure whether an interface is a trusted or untrusted source of DHCP messages. By default, all interfaces are untrusted. You can configure DHCP trust on the following types of interfaces:

- Layer 2 Ethernet interfaces
- Layer 2 port-channel interfaces

Before you begin

Make sure that the DHCP feature is enabled.

Make sure that the interface is configured as a Layer 2 interface.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Do one of the following options:

- **interface ethernet** *slot* / *port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

- Enters interface configuration mode, where *slot* / *port* is the Layer 2 Ethernet interface that you want to configure as trusted or untrusted for DHCP snooping.
- Enters interface configuration mode, where *slot* / *port* is the Layer 2 port-channel interface that you want to configure as trusted or untrusted for DHCP snooping.

Step 3 [**no**] **ip dhcp snooping trust**

Example:

```
switch(config-if)#ip dhcp snooping trust
```

Configures the interface as a trusted interface for DHCP snooping. The **no** form of this command configures the port as an untrusted interface.

Step 4 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable the DHCP relay trusted port functionality

You can enable or disable the DHCP relay trusted port functionality. By default, if the gateway address is set to all zeros in the DHCP packet and the relay information option is already present in the packet, the DHCP relay agent will not discard the packet. If the **ip dhcp relay information option trust** command is configured globally, the DHCP relay agent will discard the packet if the gateway address is set to all zeros.

Before you begin

Make sure that the DHCP feature is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp relay information option trust**

Example:

```
switch(config)#ip dhcp relay information option trust
```

Enables the DHCP relay trusted port functionality. The **no** form of this command disables this functionality.

Step 3 (Optional) **show ip dhcp relay**

Example:

```
switch#show ip dhcp relay
```

Displays the DHCP relay configuration.

Step 4 (Optional) **show ip dhcp relay information trusted-sources**

Example:

```
switch#show ip dhcp relay information trusted-sources
```

Displays the DHCP relay trusted ports configuration.

Step 5 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure an interface as a DHCP relay trusted or untrusted port

Configure a Layer 3 interface as a DHCP relay trusted or untrusted port to control DHCP relay agent information processing.

You can configure whether a Layer 3 interface is a DHCP relay trusted or untrusted interface. By default, all interfaces are untrusted. You can configure DHCP relay trust on the following types of interfaces:

- Layer 3 Ethernet interfaces and subinterfaces
- Layer 3 port-channel interfaces

Before you begin

Make sure that the DHCP feature is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface** [**ethernet** *slot/port* [. *number*] | **port-channel** *channel-number*]

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Enters interface configuration mode, where *slot/port* is the Layer 3 Ethernet interface that you want to configure as trusted or untrusted or *channel-number* is the Layer 3 port-channel interface that you want to configure as trusted or untrusted.

Step 3 [**no**] **ip dhcp relay information trusted**

Example:

```
switch(config-if)#ip dhcp relay information trusted
```

Configures the interface as a trusted interface for DHCP relay agent information. The **no** form of this command configures the port as an untrusted interface.

Note
For any Layer 3 interface, if the interface is configured as trusted either through a global command or an interface-level command, the interface is considered as a trusted interface. Hence, when the trusted-port command is enabled at the global level, any Layer 3 interface cannot be considered as untrusted irrespective of the interface-level configuration.

Step 4 (Optional) **show ip dhcp relay information trusted-sources**

Example:

```
switch#show ip dhcp relay information trusted-sources
```

Displays the DHCP relay trusted ports configuration.

Step 5 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure all interfaces as trusted or untrusted

You can configure all Layer 3 interfaces as DHCP relay trusted or untrusted interfaces. By default, all interfaces are untrusted. You can configure DHCP relay trust on the following types of interfaces:

- Layer 3 Ethernet interfaces and subinterfaces
- Layer 3 port-channel interfaces

When you enable the **ip dhcp relay information trust-all** command, any Layer 3 interface cannot be considered as untrusted irrespective of the interface-level configuration.

Before you begin

Make sure that the DHCP feature is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp relay information trust-all**

Example:

```
switch(config)#ip dhcp relay information trust-all
```

Configures the interfaces as trusted sources of DHCP messages. The **no** form of this command configures the ports as untrusted interfaces.

Step 3 (Optional) **show ip dhcp relay information trusted-sources**

Example:

```
switch#show ip dhcp relay information trusted-sources
```

Displays the DHCP relay trusted ports configuration.

Step 4 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable the DHCP relay agent

You can enable or disable the DHCP relay agent. By default, the DHCP relay agent is enabled.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp relay**

Example:

```
switch(config)#ip dhcp relay
```

Enables the DHCP relay agent. The **no** option disables the DHCP relay agent.

Step 3 (Optional) **show ip dhcp relay**

Example:

```
switch#show ip dhcp relay
```

Displays the DHCP relay configuration.

Step 4 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable option 82 for the DHCP relay agent

You can enable or disable the device to insert and remove Option 82 information on DHCP packets forwarded by the relay agent.

By default, the DHCP relay agent does not include Option 82 information in DHCP packets.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

Step 1 Enters the global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enable or Disable DHCP Relay Agent Information Option using the [**no**] **ip dhcp relay information option** command.

Example:

```
switch(config)#ip dhcp relay information option
```

Enables the DHCP relay agent to insert and remove Option 82 information on the packets that it forwards. The Option 82 information is in binary ifindex format by default. The **no** option disables this behavior.

Step 3 (Optional) Customize circuit ID format for SVIs using the [**no**] **ip dhcp relay sub-option circuit-id customized** command.

Example:

```
switch(config)#ip dhcp relay sub-option circuit-id customized
```

Programs Option 82 with the VLAN + slot + port format. This command is applicable only for SVIs. The **no** option disables this behavior.

Note
On Port-Channel interfaces, the custom circuit-ID format is not used. Instead, Option 82 uses the default binary ifindex format.

Step 4 (Optional) Configure Non-TLV format for suboptions using the [**no**] **ip dhcp relay sub-option format non-tlv** command.

Example:

```
switch(config)#ip dhcp relay sub-option format non-tlv
```

Removes subtype, and its length, from Circuit ID and Remote ID suboptions of Option 82 information.

Step 5 (Optional) Configure circuit ID format type as string using the [**no**] **ip dhcp relay sub-option circuit-id format-type string** command.

Example:

```
switch(config)#ip dhcp relay sub-option circuit-id format-type string
```

Configures Option 82 to use encoded string format instead of the default binary ifindex format. The **no** option disables this behavior.

For VLANs and SVIs:

- When this command and the **ip dhcp relay sub-option circuit-id customized** command are both configured, the **ip dhcp relay sub-option circuit-id format-type string** command is programmed.
- When the **ip dhcp relay sub-option circuit-id format-type string** command is removed, the **ip dhcp relay sub-option circuit-id customized** command is programmed.

- When both commands are removed, the ifindex is programmed.

For other interfaces, if the **ip dhcp relay sub-option circuit-id format-type string** command is configured, it is used. Otherwise, the default ifindex is programmed.

Step 6 (Optional) Display DHCP Relay configuration using the **show ip dhcp relay** command.

Example:

```
switch#show ip dhcp relay
```

Step 7 (Optional) Display DHCP configuration using the **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Note
In the output of this command, the **circuit-id** parameter value will be displayed in double quotes, even if configured with or without double quotes.

Enable or disable VRF support for the DHCP relay agent

You can configure the device to support the relaying of DHCP requests that arrive on an interface in one VRF to a DHCP server in a different VRF.

Before you begin

You must enable Option 82 for the DHCP relay agent.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp relay information option vpn**

Example:

```
switch(config)#ip dhcp relay information option vpn
```

Enables VRF support for the DHCP relay agent. The **no** option disables this behavior.

Step 3 [**no**] **ip dhcp relay sub-option type cisco**

Example:

```
switch(config)#ip dhcp relay sub-option type cisco
```

Enables DHCP to use Cisco proprietary numbers 150, 152, and 151 when filling the link selection, server ID override, and VRF name/VPN ID relay agent Option 82 suboptions. The **no** option causes DHCP to use RFC numbers 5, 11, and 151 for the link selection, server ID override, and VRF name/VPN ID suboptions.

Step 4 (Optional) **show ip dhcp relay**

Example:

```
switch#show ip dhcp relay
```

Displays the DHCP relay configuration.

Step 5 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Disable the server identifier override option

Beginning with Cisco NX-OS Release 9.3(3), you can disable the server identifier override option. This option is added by default in DHCP Option 82 packets for a DHCP relay VPN configuration or source interface configuration.

Before you begin

You must enable Option 82 for the DHCP relay agent.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp relay information option server-id-override-disable**

Example:

```
switch(config)#ip dhcp relay information option server-id-override-disable
```

Disables the server identifier override option in DHCP Option 82 packets.

Note

You can use the **no** form of this command to re-enable the server identifier override option.

Configure DHCP server addresses on an interface

Configure DHCP server IP addresses on an interface to enable the relay agent to forward DHCP requests and replies between hosts and servers.

You can configure DHCP server IP addresses on an interface. When an inbound DHCP BOOTREQUEST packet arrives on the interface, the relay agent forwards the packet to all DHCP server IP addresses specified. The relay agent forwards replies from all DHCP servers to the host that sent the request.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCP server is correctly configured.

Determine the IP address for each DHCP server that you want to configure on the interface.

If the DHCP server is in a different VRF than the interface, ensure that you have enabled VRF support.

Note If an ingress router ACL is configured on an interface that you are configuring with a DHCP server address, ensure that the router ACL permits DHCP traffic between DHCP servers and DHCP hosts.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 Do one of the following options:

- **interface ethernet** *slot* / *port* [**.** *number*]
- **interface vlan** *vlan-id*
- **interface port-channel** *channel-id* [. *subchannel-id*]

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

- Enters interface configuration mode, where *slot* / *port* is the physical Ethernet interface that you want to configure with a DHCP server IP address. If you want to configure a subinterface, include the *number* argument to specify the subinterface number.

  Note
  Port-channel subinterfaces are supported only in Cisco NX-OS Releases 6.1(2)I3(3) and 6.1(2)I3(3a). They are not supported in Cisco NX-OS Release 9.2(1).

- Enters interface configuration mode, where *vlan-id* is the ID of the VLAN that you want to configure with a DHCP server IP address.
- Enters interface configuration mode, where *channel-id* is the ID of the port channel that you want to configure with a DHCP server IP address. If you want to configure a subchannel, include the *subchannel-id* argument to specify the subchannel ID.

Step 3 **ip dhcp relay address** *IP-address* [**use-vrf** *vrf-name*]

Example:

```
switch(config-if)#ip dhcp relay address 10.132.7.120 use-vrf red
```

Configures an IP address for a DHCP server to which the relay agent forwards BOOTREQUEST packets received on this interface.

To configure more than one IP address, use the **ip dhcp relay address** command once per address.

Step 4 (Optional) **show ip dhcp relay address**

Example:

```
switch#show ip dhcp relay address
```

Displays all the configured DHCP server addresses.

Step 5 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure the DHCP relay source interface

Configure the source interface for the DHCP relay agent to control which address is used as the source address in relayed DHCP packets.

You can configure the source interface for the DHCP relay agent. By default, the DHCP relay agent uses the relay agent address as the source address of the outgoing packet. Configuring the source interface enables you to use a more stable address (such as the loopback interface address) as the source address of relayed messages. When DHCP relay source interface is configured, the device adds the configured source interface IP address as giaddr to the DHCP packet if source interface VRF is same as that of DHCP server VRF. Otherwise, IP address of the interface through which the server is reachable, will be used as giaddr.

Before you begin

- Ensure that the DHCP feature is enabled.
- Ensure that the DHCP relay agent is enabled.

- Ensure CLI dhcp relay information option and ip dhcp relay information option vpn are enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp relay source-interface** *interface*

Example:

```
switch(config)#ip dhcp relay source-interface loopback 2
```

Configures the source interface for the DHCP relay agent.

Note

- The DHCP relay source interface can be configured globally, per interface, or both. When both the global and interface levels are configured, the interface-level configuration overrides the global configuration.
- When configuring DHCP relay on an interface with a source interface configuration, ensure that Option 82 or VPN configuration is enabled, regardless of whether the server and client are in the same or different VRFs.

Step 3 (Optional) **show ip dhcp relay** [**interface** *interface*]

Example:

```
switch#show ip dhcp relay
```

Displays the DHCP relay configuration.

Step 4 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable DHCP smart relay globally

You can enable or disable DHCP smart relay globally on the device.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCP relay agent is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ip dhcp smart-relay global**

Example:

```
switch(config)#ip dhcp smart-relay global
```

Enables DHCP smart relay globally. The **no** form of this command disables DHCP smart relay.

Step 3 (Optional) **show ip dhcp relay**

Example:

```
switch#show ip dhcp relay
```

Displays the DHCP smart relay configuration.

Step 4 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable DHCP smart relay on a layer 3 interface

You can enable or disable DHCP smart relay on Layer 3 interfaces.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCP relay agent is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface** *interface slot* / *port*

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

Enters interface configuration mode, where *slot* / *port* is the interface for which you want to enable or disable DHCP smart relay.

Step 3 [**no**] **ip dhcp smart-relay**

Example:

```
switch(config-if)#ip dhcp smart-relay
```

Enables DHCP smart relay on the interface. The **no** form of this command disables DHCP smart relay on the interface.

Step 4 **exit**

Example:

```
switch(config-if)#exit
switch(config)#
```

Exits interface configuration mode.

Step 5 **exit**

Example:

```
switch(config)#exit
switch#
```

Exits global configuration mode.

Step 6 (Optional) **show ip dhcp relay**

Example:

```
switch#show ip dhcp relay
```

Displays the DHCP smart relay configuration.

Step 7 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure the DHCP relay Subnet-Selection

If an interface includes both a primary and a secondary IP address, then by default the DHCP relay uses the primary subnet to request the IP address allocation from the server. You must enable DHCP smart relay if you want the DHCP relay to use the secondary IP address. With smart relay enabled, DHCP relay first requests the IP address in the primary subnet. If it fails to get the IP address in the primary subnet, it requests the IP address of the secondary subnet. The IP address of the secondary subnet is not chosen by default.

With the introduction of the DHCP relay subnet selection feature, you have an option to choose the IP address of either the primary or the secondary subnet based on your requirements. When you configure the DHCP relay subnet selection, the DHCP relayed packet includes the subnet that is used in subnet-selection for a source and relay agent. If there is a VPN or a source interface option, the option 82 link selection is updated with the configured subnet.

The DHCP Smart relay and the subnet-selection configuration are mutually exclusive at the interface level. If DHCP Smart relay is enabled globally and the subnet-selection is configured on the interface level, then the interface configuration takes precedence.

With the DHCP VPN or the source interface option, the DHCP server must use the option 82 link-selection to assign the IP address.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface vlan** *vlan-id*

Example:

```
switch(config)#interface vlan 3
switch(config-if)#
```

Enters interface configuration mode.

Step 3 **ip dhcp relay subnet-selection** *ip address*

Example:

```
switch(config-if)#ip dhcp relay subnet-selection 192.0.2.1
```

Configures the DHCP relay subnet-selection for the specified IP address.

Configuring DHCPv6

Enable or disable the DHCPv6 relay agent

Use this procedure to enable or disable the DHCPv6 relay agent on your device.

You can enable or disable the DHCPv6 relay agent. By default, the DHCPv6 relay agent is enabled.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ipv6 dhcp relay**

Example:

```
switch(config)#ipv6 dhcp relay
```

Enables the DHCPv6 relay agent. The **no** option disables the relay agent.

Step 3 (Optional) **show ipv6 dhcp relay** [**interface** *interface*]

Example:

```
switch#show ipv6 dhcp relay
```

Displays the DHCPv6 relay configuration.

Step 4 [**no**] **ipv6 dhcp relay prefix-delegation route**

Example:

```
switch(config)#ipv6 dhcp relay prefix-delegation route
```

Enables the IPv6 prefix route addition. The **no** option disables the prefix route addition.

Step 5 (Optional) **show ipv6 dhcp relay prefix-delegation**

Example:

```
switch#show ipv6 dhcp relay prefix-delegation
```

Displays the DHCPv6 IAPD entries (Client IP, interface, lease timer and delegated prefix).

Step 6 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or disable VRF support for the DHCPv6 relay agent

You can configure the device to support the relaying of DHCPv6 requests that arrive on an interface in one VRF to a DHCPv6 server in a different VRF.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ipv6 dhcp relay option vpn**

Example:

```
switch(config)#ipv6 dhcp relay option vpn
```

Enables VRF support for the DHCPv6 relay agent. The **no** option disables this behavior.

Step 3 [**no**] **ipv6 dhcp relay option type cisco**

Example:

```
switch(config)#ipv6 dhcp relay option type cisco
```

Causes the DHCPv6 relay agent to insert virtual subnet selection (VSS) details as part of the vendor-specific option. The **no** option causes the DHCPv6 relay agent to insert VSS details as part of the VSS option (68), which is defined in RFC-6607. This command is useful when you want to use DHCPv6 servers that do not support RFC-6607 but allocate IPv6 addresses based on the client VRF name.

Step 4 (Optional) **show ipv6 dhcp relay** [**interface** *interface*]

Example:

```
switch#show ipv6 dhcp relay
```

Displays the DHCPv6 relay configuration.

Step 5 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or Disable DHCPv6 Smart Relay Globally

You can enable or disable DHCPv6 smart relay globally on the device.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **ipv6 dhcp smart-relay global**

Example:

```
switch(config)# ipv6 dhcp smart-relay global
```

Enables DHCPv6 smart relay globally. The **no** form of this command disables DHCPv6 smart relay.

Step 3 (Optional) **show ipv6 dhcp relay**

Example:

```
switch# show ipv6 dhcp relay
```

Displays the DHCPv6 smart relay configuration.

Step 4 (Optional) **show running-config dhcp**

Example:

```
switch# show running-config dhcp
```

Displays the DHCPv6 configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable or Disable DHCPv6 Smart Relay on a Layer 3 Interface

You can enable or disable DHCP smart relay on Layer 3 interfaces.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface** *interface slot* / *port*

Example:

```
switch(config)# interface ethernet 2/3
switch(config-if)#
```

Enters interface configuration mode, where *slot* / *port* is the interface for which you want to enable or disable DHCPv6 smart relay.

Step 3 [**no**] **ipv6 dhcp smart-relay**

Example:

```
switch(config-if)# ipv6 dhcp smart-relay
```

Enables DHCPv6 smart relay on the interface. The **no** form of this command disables DHCPv6 smart relay on the interface.

Step 4 **exit**

Example:

```
switch(config-if)# exit
switch(config)#
```

Exits interface configuration mode.

Step 5 **exit**

Example:

```
switch(config)# exit
switch#
```

Exits global configuration mode.

Step 6 (Optional) **show ipv6 dhcp relay**

Example:

```
switch# show ipv6 dhcp relay
```

Displays the DHCPv6 smart relay configuration.

Step 7 (Optional) **show running-config dhcp**

Example:

```
switch# show running-config dhcp
```

Displays the DHCPv6 configuration.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure DHCPv6 server addresses on an interface

You can configure DHCPv6 server IP addresses on an interface. When an inbound DHCP BOOTREQUEST packet arrives on the interface, the relay agent forwards the packet to all DHCPv6 server IP addresses specified. The relay agent forwards replies from all DHCPv6 servers to the host that sent the request.

Before you begin

- Ensure that the DHCP feature is enabled.
- Ensure that the DHCPv6 server is correctly configured.
- Determine the IP address for each DHCPv6 server that you want to configure on the interface.
- If the DHCPv6 server is in a different VRF than the interface, ensure that you have enabled VRF support.



Note If an ingress router ACL is configured on an interface that you are configuring with a DHCPv6 server address, ensure that the router ACL permits DHCP traffic between DHCPv6 servers and DHCP hosts.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Do one of the following options:

- **interface ethernet** *slot* / *port*
- **interface port-channel** *channel-id*

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

- Enters interface configuration mode, where *slot* / *port* is the physical Ethernet interface that you want to configure with a DHCPv6 server IP address.
- Enters interface configuration mode, where *channel-id* is the ID of the port channel that you want to configure with a DHCPv6 server IP address.

Step 3 [**no**] **ipv6 dhcp relay address** *IPv6-address* [**use-vrf** *vrf-name*] [**interface** *interface*]

Example:

```
switch(config-if)#ipv6 dhcp relay address FF02:1::FF0E:8C6C use-vrf red
```

Configures an IP address for a DHCPv6 server to which the relay agent forwards BOOTREQUEST packets received on this interface.

Use the **use-vrf** option to specify the VRF name of the server if it is in a different VRF and the other argument interface is used to specify the output interface for the destination.

The server address can either be a link-scoped unicast or multicast address or a global or site-local unicast or multicast address. The **interface** option is mandatory for a link-scoped server address and multicast address. It is not allowed for a global or site-scoped server address.

To configure more than one IP address, use the **ipv6 dhcp relay address** command once per address.

Step 4 (Optional) **show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCPv6 configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable DHCPv6 option 79

Beginning with Cisco NX-OS Release 9.3(3), you can enable the use of the DHCPv6 client's link-layer address through Option 79. When you enable this feature, the switch adds Option 79 with relay forward packets, and the IPv6 client's link-layer address is inserted into the Options field of the DHCPv6 packet.

This feature is supported for both regular DHCPv6 and DHCPv6 with VXLAN.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 **ipv6 dhcp relay option79**

Example:

```
switch(config)#ipv6 dhcp relay option79
```

Enables the DHCP relay forward packets that are transmitted from the relay server to the DHCP server to carry the DHCPv6 host's link-layer address.

This command affects the transmitted relay forward packets only.

Configure the DHCPv6 relay source interface

You can configure the source interface for the DHCPv6 relay agent. By default, the DHCPv6 relay agent uses the relay agent address as the source address of the outgoing packet. Configuring the source interface enables you to use a more stable address (such as the loopback interface address) as the source address of relayed messages.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCPv6 relay agent is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 [**no**] **ipv6 dhcp relay source-interface** *interface*

Example:

```
switch(config)#ipv6 dhcp relay source-interface loopback 2
```

Configures the source interface for the DHCPv6 relay agent.

Note
The DHCPv6 relay source interface can be configured globally, per interface, or both. When both the global and interface levels are configured, the interface-level configuration overrides the global configuration.

Step 3 (Optional) **show ipv6 dhcp relay** [**interface** *interface*]

Example:

```
switch#show ipv6 dhcp relay
```

Displays the DHCPv6 relay configuration.

Step 4 (Optional) **show running-config dhcp show running-config dhcp**

Example:

```
switch#show running-config dhcp
```

Displays the DHCP configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure the IPv6 RA guard feature

Use this procedure to enable the IPv6 RA guard feature on a Layer 2 interface, which helps prevent unwanted IPv6 router advertisements from being processed.

You can configure the IPv6 router advertisement (RA) guard feature for Cisco Nexus 9200, 9300 , and 9300-EX Series switches and the N9K-X9732C-EX line card . This feature is used to drop all incoming IPv6 RA packets on a Layer 2 interface.

Before you begin

- You must enable DHCP (using the **feature dhcp** command).
- To enable DHCP relay on any interface, you must disable DHCP on interfaces that have an IPv4 or IPv6 address assigned using DHCP (dynamic IP addressing).
- Make sure that both PTP (**feature ptp**) and NV overlay (**feature nv overlay**) are not already configured. A dynamic ifacl label is reserved when these features are configured. However, only two dynamic ifacl label bits are available. If both of these features are already configured, a dynamic ifacl label will not be available for IPv6 RA guard, and the feature cannot be enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface** *interface slot/port*

Example:

```
switch(config)#interface ethernet 2/2
switch(config-if)#
```

Enters interface configuration mode.

Step 3 [**no**] **ipv6 nd raguard**

Example:

```
switch(config-if)#ipv6 nd raguard
```

Enables the IPv6 RA guard feature on the specified interface.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Enable the DHCP client

You can use the DHCP client feature to enable the configuration of an IPv4 or IPv6 address on an interface. Interfaces can include routed ports, the management port, and switch virtual interfaces (SVIs). Layer 3 subinterfaces are not supported.

Note DHCP client is independent of the DHCP relay and DHCP snooping processes, so it does not require that the **feature dhcp** command be enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 Do one of the following options:

- **interface ethernet** *slot* / *port*
- **interface mgmt 0**
- **interface vlan** *vlan-id*

Example:

```
switch(config)#interface vlan 3
switch(config-if)#
```

- Enters interface configuration mode, where *slot* / *port* is the physical Ethernet interface for which you want to enable the DHCP client feature.
- Enters interface configuration mode and specifies the management interface as the interface for which you want to enable the DHCP client feature.
- Enters interface configuration mode, where *vlan-id* is the ID of the VLAN for which you want to enable the DHCP client feature.

Step 3 **ipv6 address use-link-local-only**

Example:

```
switch(config-if)#ipv6 address use-link-local-only
```

You must enter this command before assigning an IPv6 address to the interface in the next step. This command is not required if you will assign an IPv4 address to the interface.

Step 4 [**no**] {**ip** | **ipv6**} **address dhcp**

Example:

```
switch(config-if)#ip address dhcp
```

Assigns an IPv4 or IPv6 address to the interface.

The **no** form of this command releases the IP address.

Step 5 (Optional) Do one of the following options:

- **show running-config interface ethernet** *slot* / *port*
- **show running-config interface mgmt 0**
- **show running-config interface vlan** *vlan-id*

Example:

```
switch#show running-config interface vlan 3
```

Displays the IPv4 or IPv6 address assigned to the interface in the running configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Only the {**ip** | **ipv6**} **address dhcp** command is saved. The assigned IP address is not saved even though it shows in the running configuration.

Configuring UDP relay

Explains configuring UDP relay.

About UDP relay

The UDP relay feature, also known as the IP Helper feature, enables routers to forward broadcast packets destined for UDP ports except DHCPv4 port numbers 67 and 68.

- Routers do not forward broadcast packets by default; UDP relay must be configured to enable this functionality.
- The **ip forward-protocol udp** command enables the UDP relay feature, which is disabled by default.
- UDP relay is supported on physical ports, interface VLANs (SVIs), L3 port channels, L3 subinterfaces, and M1/M2 FEX interfaces.

UDP relay configuration and supported ports

The UDP relay feature can be enabled using the **ip forward-protocol udp** command, and is supported on various Layer 3 interfaces.

The following UDP ports are enabled by default when you run the **ip forward-protocol udp** command:

- Trivial File Transfer Protocol (TFTP) (port 69)
- Domain Naming System (port 53)
- Time service (port 37)
- NetBIOS Name Server (port 137)
- NetBIOS Datagram Server (port 138)
- TACACS service (port 49)
- IEN-116 Name Service (port 42)

To forward a packet, configure IP address object groups with the forwarding destination IP addresses or network addresses and associate them with the L3 interfaces. Subnet broadcast can also be configured for each Layer 3 interface.

The UDP relay feature is supported on the following types of Layer 3 interfaces:

- Physical port
- Interface VLAN (SVI)
- L3 port channel
- L3 subinterfaces

- M1 and M2 FEX interfaces

By default, UDP forwarding for directed broadcast packets is not enabled on an interface. You can enable the UDP relay feature on a L3 or switch virtual interface (SVI) by using the **ip udp relay subnet-broadcast** command. When you enable subnet broadcast, all the UDP packets that meet the following criteria are forwarded:

1. The packet must be an IP level-directed broadcast, that is, the primary subnet broadcast or any of the secondary subnet broadcasts for the interface.
2. The destination UDP port number of the packet must be any of the default UDP ports or any other UDP port that is specified by using the **ip forward-protocol udp** *udp-port-num* configuration command.
3. When you enable the subnet broadcast, policies should be updated for the respective L3 or SVI interface.

Enabling UDP relay on a layer 3 interface

For example, to enable UDP relay on a switch virtual interface (SVI), use the **ip forward-protocol udp** command to specify the UDP ports, and configure the forwarding destination addresses using IP address object groups. This setup allows the router to relay broadcast packets for services such as TFTP, DNS, and NetBIOS.

Guidelines and limitations for UDP relay

Use this reference to review the guidelines and limitations for UDP relay.

UDP relay support and configuration limits

- UDP relay is supported on Cisco Nexus 9200, 9332C, 9364C, 9300-EX, 9300-FX/FX2/FXP platform switches, and Cisco Nexus 9500 platform switches with -EX/FX line cards, beginning with Cisco NX-OS Release 9.3(5).
- UDP relay is not supported on Cisco Nexus 9300-FX3/GX/GX2/H2R/H1 Series switches and Cisco Nexus 9500 platform switches with -R/-RX line cards.
- The UDP port must be in the range of 1 to 65565.
- Any L3 or SVI interface can be associated with a maximum of one object group, and any interface can be associated with up to 300 UDP relay IP addresses.
- The UDP relay feature supports seven UDP ports.
- The object-group name can be a maximum of 64 alphanumeric characters.
- DHCP and UDP relay cannot co-exist.
- Subnet broadcast is not supported.
- Beginning with Cisco NX-OS Release 10.6(1)F, you must enable `ip forward-protocol udp` globally before configuring `ip forward-protocol udp <port>`. If you do not enable the global configuration, the system displays a warning message.

Configure the UDP relay feature

Before you begin

Ensure that you have enabled the DHCP feature.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
```

Enters global configuration mode.

Step 2 **[no] ip forward-protocol udp**

Example:

```
switch(config)#ip forward-protocol udp
```

Enables the UDP relay feature. By default, the UDP relay feature is disabled. However, it is enabled on the predefined set of UDP ports.

Step 3 (Optional) **[no] ip forward-protocol udp** *udp-port-number*

Example:

```
switch(config)#ip forward-protocol udp 1
```

Enable the UDP relay feature on the non-default UDP ports.

Note
You can enable or disable UDP forwarding for any UDP port in the range 1 to 65565 except the DHCP ports.

Step 4 **[no] object-group udp relay ip address** *object-group-name*

Example:

```
switch(config)#object-group udp relay ip address relay1
switch(config-udp-ogroup)#
```

Configure an object group that consists of destination IP addresses to which the packets are forwarded

Step 5 **[no] {host** *host-addr* | *network-addr network-mask* | *network-addr/mask-length*}

Example:

```
switch(config-udp-ogroup)#host 192.0.2.1
192.0.2.254 192.0.2.1
10.1.1.1/24
```

Configures the destination IP addresses to which the packets are forwarded.

Note
For each entry that you want to create, use the **host** command to specify a single host, or omit the **host** command to specify a network of hosts.

Step 6 **exit**

Example:

```
switch(config-udp-ogroup)#exit
```

Exists the interface configuration mode.

Step 7 **interface ethernet** *slot/port*

Example:

```
switch(config)#interface ethernet 1/1
switch(config-if)#
```

Associates the object froup with a Layer 3 interface.

Note
The L3 interface can be a physical port, interface VLAN (SVI), L3 port channel, or L3 subinterfaces.

Step 8 **ip udp relay addrgroup** *object-group-name*

Example:

```
switch(config-if)#ip udp relay addrgroup group1
```

Associates an object group to the interface.

Configuration example for UDP relay

The configuration example for UDP relay demonstrates the required commands to enable and set up UDP relay functionality on a network device.

- Shows the use of **feature dhcp** and **ip forward-protocol udp** commands.
- Illustrates how to define an object group for UDP relay IP addresses.
- Demonstrates applying the UDP relay address group to an interface.

Supporting reference information for UDP relay configuration

The following information provides context and details for configuring UDP relay using a sample running configuration.

Configuring UDP relay

This example shows a running configuration to configure the UDP relay feature.

```
configure terminal
 feature dhcp
 ip forward-protocol udp
 object-group udp relay ip address udprelay1
  host 192.0.2.1
  192.0.2.2 192.0.2.3
  10.1.1.1/24
 exit
 interface ethernet e1/1
  ip udp relay addrgroup udprelay1
 exit
```

Verifying the UDP Relay Configuration

The UDP relay configuration can be verified using specific show commands that display configuration details, interface attributes, and object-group associations.

- Use **show ip udp relay** to display the UDP relay configuration.
- Use **show ip udp relay interface** with optional interface parameters to display interface-level attributes.
- Use **show ip udp relay object-group** to display all configured UDP relay object-groups and their associated IP addresses.
- Use **show ip udp relay object-group** *object-group-name* to display a specific object-group and its associated IP addresses.

Show Commands for UDP Relay Verification

To display UDP relay configuration information, perform one of the following tasks:

Table 95: UDP Relay Verification Commands

Command	Purpose
show ip udp relay	Displays the UDP relay configuration.
show ip udp relay interface [{ *interface-type* | *interface-range* }]	Displays the interface level attributes.
show ip udp relay object-group	Displays all configured UDP relay object-groups and the associated IP addresses.
show ip udp relay object-group *object-group-name*	Displays the object-group and the associated IP addresses.

DHCP configuration verification commands

Use this reference to verify the DHCP configuration.

- DHCP configuration verification involves using show commands to display relay, snooping, and client information.
- Commands are available for both IPv4 and IPv6 DHCP features, including relay, prefix delegation, and snooping.
- Some commands provide detailed options for filtering and displaying specific DHCP-related data.

DHCP configuration verification commands and descriptions

Table 96: DHCP verification commands

Command	Purpose	Options/Notes
show ip dhcp relay	Displays the DHCP relay configuration.	-
show ipv6 dhcp relay [**interface** *interface*]	Displays the DHCPv6 relay global or interface-level configuration.	-
show ipv6 dhcp relay prefix-delegation	Displays the DHCPv6 IAPD entries on Relay agent.	**client** : Displays the prefix bindings for a client. **detail** : Displays the detailed information. **interface** : Displays the prefix bindings for an interface. **prefix** : Displays a specific prefix binding.
show ipv6 route dhcpv6	Displays connected routes owned by dhcpv6.	**all** : Displays the routes for protocol for backup next-hops too. **bind-label** : Displays the routes with this bind-label only. **detail** : Displays the routes in full detail. **interface** : Displays the routes with this output interface only. **next-hop** : Displays the routes with this next-hop only. **summary** : Displays the route counts. **updated** : Displays the routes filtered by last updated time. **vrf** : Displays per-VRF information. **Note** Ensure that the DHCPv6-PD feature is enabled.
show ip dhcp relay address	Displays all the DHCP server addresses configured on the device.	-
show ip dhcp snooping	Displays general information about DHCP snooping.	-
show running-config dhcp [**all**]	Displays the DHCP configuration in the running configuration. **Note** The **show running-config dhcp** command displays the **ip dhcp relay** and the **ipv6 dhcp relay** commands, although these are configured by default.	-

Command	Purpose	Options/Notes
show running-config interface {**ethernet** *slot/port* | **mgmt 0** | **vlan** *vlan-id*}	Displays the IPv4 or IPv6 address assigned to the interface when DHCP client is enabled.	-
show startup-config dhcp [**all**]	Displays the DHCP configuration in the startup configuration.	-

Displaying IPv6 RA guard statistics

- The **show ipv6 raguard statistics** command displays IPv6-related RA guard statistics.

How to display IPv6 RA guard statistics

To display IPv6 RA guard statistics, perform the following task:

- Use the **show ipv6 raguard statistics** command to view statistics related to IPv6 RA guard.

Table 97: Command Reference

Command	Purpose
show ipv6 raguard statistics	Displays IPv6-related RA guard statistics.

Example: displaying IPv6 RA guard statistics

The following example shows sample statistics output for the **show ipv6 raguard statistics** command:

```
switch#show ipv6 raguard statistics
-------------------------------------
Interface     Rx         Drops
------------- ---------- ------------
Ethernet1/53  4561102    4561102
```

Displaying DHCP snooping bindings

The **show ip dhcp snooping binding** command displays all entries from the DHCP snooping binding database.

- Allows filtering by IP address, MAC address, VLAN, or interface.
- Shows binding type (dynamic or static) and lease information.
- Displays associated VLAN and interface details for each binding entry.

Reference information for displaying DHCP snooping bindings

The **show ip dhcp snooping binding** command provides a detailed view of DHCP snooping bindings on the device.

- MacAddress: The MAC address of the client.
- IpAddress: The assigned IP address.
- LeaseSec: Lease duration in seconds or 'infinite' for static entries.
- Type: Indicates if the entry is static or dynamic.
- VLAN: VLAN ID associated with the binding.
- Interface: The interface where the binding was learned.

Table 98: DHCP snooping binding table example

MacAddress	IpAddress	LeaseSec	Type	VLAN	Interface
00:00:5E:00:53:01	10.3.2.2	infinite	static	13	Ethernet2/46
00:00:5E:00:53:02	10.2.2.2	infinite	static	100	Ethernet2/10

Example: displaying DHCP snooping bindings

The following example shows the output of the **show ip dhcp snooping binding** command, listing static bindings for two clients:

Clearing the DHCP snooping binding database

The DHCP snooping binding database can be cleared using specific CLI commands to remove all or selected entries.

- Use **clear ip dhcp snooping binding** to clear all entries from the database.
- Use **clear ip dhcp snooping binding interface ethernet** *slot/port* to clear entries for a specific Ethernet interface.
- Use **clear ip dhcp snooping binding interface port-channel** *channel-number* to clear entries for a specific port-channel interface.
- Use **clear ip dhcp snooping binding vlan** *vlan-id* [**mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot /port* | **port-channel** *channel-number*}] to clear a specific VLAN entry.

Monitoring DHCP

- DHCP monitoring involves using show commands to view DHCP snooping and relay statistics.
- Statistics can be monitored at both the global and interface levels.
- Separate commands are available for DHCP snooping, DHCP relay, and DHCPv6 relay.

To monitor DHCP snooping and relay statistics, use the following commands as appropriate for your environment.

- Use the **show ip dhcp snooping statistics** command to monitor DHCP snooping.
- Use the **show ip dhcp relay statistics** [**interface** *interface*] command to monitor DHCP relay statistics at the global or interface level.
- Use the **show ipv6 dhcp relay statistics** [**interface** *interface*] command to monitor DHCPv6 relay statistics at the global or interface level.

Clearing DHCP snooping statistics

The **clear ip dhcp snooping statistics** command is used to remove accumulated DHCP snooping statistics from the device.

- Clears all DHCP snooping statistics collected on the device.
- Can be used with the **vlan** option to target a specific VLAN.
- Helps in resetting counters for troubleshooting or monitoring purposes.

Command usage for clearing DHCP snooping statistics

To clear DHCP snooping statistics, use the following command in privileged EXEC mode:

- **clear ip dhcp snooping statistics**
- **clear ip dhcp snooping statistics vlan** *vlan-id*

Clearing DHCP relay statistics

- The **clear ip dhcp relay statistics** command clears global DHCP relay statistics.
- The **clear ip dhcp relay statistics interface** *interface* command clears DHCP relay statistics for a specific interface.
- The **clear ip dhcp global statistics** command clears DHCP statistics globally.

Clearing DHCPv6 relay statistics

- The **clear ipv6 dhcp relay statistics** command clears global DHCPv6 relay statistics.
- The **clear ipv6 dhcp relay statistics interface** *interface* command clears DHCPv6 relay statistics for a specific interface.

Clearing DHCPv6 relay statistics reference

Use the following commands to clear DHCPv6 relay statistics either globally or for a specific interface.

Example: clearing DHCPv6 relay statistics

For example, to clear all DHCPv6 relay statistics globally, enter **clear ipv6 dhcp relay statistics** . To clear statistics for a specific interface, use **clear ipv6 dhcp relay statistics interface** *Ethernet1/1* .

Clearing DHCPv6-PD binding

Clearing DHCPv6-PD binding removes the binding and the associated IPv6 route from the device.

- Clearing the binding also clears the corresponding IPv6 route.
- In vPC environments, clearing the binding on one peer also clears it on the other peer.
- Specific clear commands are available for all entries, client entries, or interface-specific entries.

Reference information for clearing DHCPv6-PD binding

The following clear commands are used to remove DHCPv6-PD bindings and their associated IPv6 routes.

- **clear ipv6 dhcp relay prefix-delegation all** : Clears all entries in the DHCPv6-PD binding.
- **clear ipv6 dhcp relay prefix-delegation client** : Clears a client's IPv6 address entries in the DHCPv6-PD binding.
- **clear ipv6 dhcp relay prefix-delegation interface** : Clears entries associated with a specific interface in the DHCPv6-PD binding.

Note After a valid PD entry is cleared through the CLI, no further PD entry and route learning occurs with Renew/Rebind packets.

Clearing DHCPv6-PD binding

For example, in a vPC environment, clearing the DHCPv6-PD binding at one vPC peer switch will also clear the binding and the corresponding IPv6 route at the other vPC peer switch.

Configuration examples for DHCP

- DHCP snooping can be enabled on specific VLANs with Option 82 support.
- Ethernet interfaces can be configured as trusted for DHCP server connectivity.
- DHCP relay and smart relay agents allow forwarding DHCP requests to servers in specific VRFs or with custom giaddr fields.

Reference information for DHCP configuration examples

These examples demonstrate how to configure DHCP snooping, relay, and smart relay features on Ethernet interfaces for different network scenarios.

DHCP configuration examples

This section provides configuration examples for enabling DHCP snooping, DHCP relay agent, and DHCP smart relay agent on Ethernet interfaces.

Example 1: Enable DHCP snooping on VLANs with Option 82 and trusted interface

This example shows how to enable DHCP snooping on VLANs 1 and 50, enable Option 82 support, and configure Ethernet interface 2/5 as trusted because the DHCP server is connected to that interface:

```
feature dhcp
ip dhcp snooping
ip dhcp snooping information option
interface ethernet 2/5
ip dhcp snooping trust
ip dhcp snooping vlan 1
ip dhcp snooping vlan 50
```

Example 2: Enable DHCP relay agent and configure server IP for a VRF

This example shows how to enable the DHCP relay agent and configure the DHCP server IP address for Ethernet interface 2/3, where the DHCP server IP address is 10.132.7.120 and the DHCP server is in the VRF instance named red:

```
feature dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
interface ethernet 2/3
ip dhcp relay address 10.132.7.120 use-vrf red
```

Example 3: Enable and use the DHCP smart relay agent

This example shows how to enable and use the DHCP smart relay agent. The device forwards DHCP broadcast packets received on Ethernet interface 2/2 to the DHCP server (10.55.11.3), inserting 192.168.100.1 in the giaddr field. If the DHCP server has a pool configured for the 192.168.100.0/24 network, it responds. If the server does not respond, the device sends two more requests using 192.168.100.1 in the giaddr field. If the device still does not receive a response, it starts using 172.16.31.254 in the giaddr field instead.

```
feature dhcp
ip dhcp relay
ip dhcp smart-relay global
interface ethernet 2/2
ip address 192.168.100.1/24
ip address 172.16.31.254/24 secondary
ip dhcp relay address 10.55.11.3
```

Configuration examples for DHCP client

The DHCP client feature enables automatic assignment of IPv4 addresses to VLAN interfaces using configuration commands.

- Allows dynamic IP address assignment to VLAN interfaces.
- Reduces manual configuration effort for network administrators.

- Ensures VLAN interfaces receive valid IP addresses from a DHCP server.

The following information supports the configuration of a DHCP client on a VLAN interface.

Assigning an IPv4 address to a VLAN interface using DHCP

The following example shows how the DHCP client feature can be used to assign an IPv4 address to a VLAN interface:

```
switch#configure terminal
switch(config)#interface vlan 7
switch(config-if)#no shutdown
switch(config-if)#ip address dhcp
switch#show running-config interface vlan 7
interface Vlan7
no shutdown
ip address dhcp
```

Additional references for DHCP

This topic provides related documents and standards for DHCP, including references to Dynamic ARP inspection, IP Source Guard, vPCs, VRFs, and relevant RFCs.

Related Documents

Table 99: Related documents

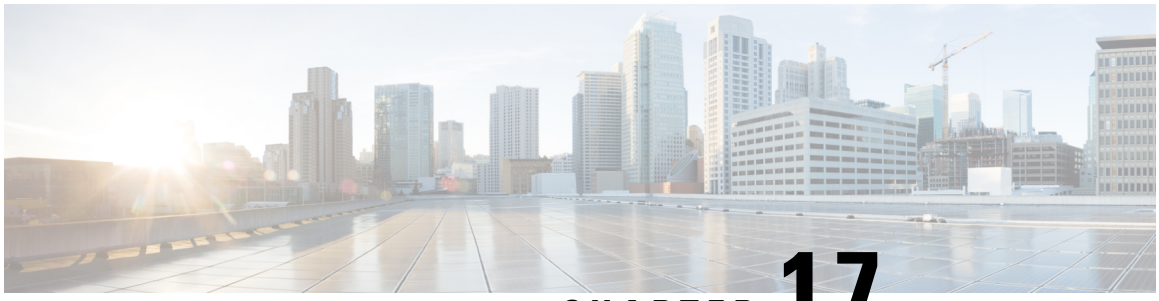
Related Topic	Document Title
Dynamic ARP inspection (DAI)	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
IP Source Guard	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
vPCs	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
VRFs and Layer 3 virtualization	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards

Table 100: Standards

Standards	Title
RFC 2131	Dynamic Host Configuration Protocol (https://example.com/doc/html/rfc2131)
RFC 3046	DHCP Relay Agent Information Option (https://example.com/doc/html/rfc3046)

Standards	Title
RFC3633	IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6 (RFC) (https://example.com/html/rfc3633)
RFC 6607	Virtual Subnet Selection Options for DHCPv4 and DHCPv6 (https://example.com/doc/html/rfc6607)
RFC 6939	Client Link-Layer Address Option in DHCPv6 (https://example.com/doc/html/rfc6939)



CHAPTER 17

Configuring IPv6 First Hop Security

This chapter describes how to configure First Hop Security (FHS) features on Cisco NX-OS devices.

This chapter includes the following sections:

- About First-Hop security, on page 501
- Guidelines and limitations of First-Hop security, on page 502
- About vPC First-Hop security configuration, on page 503
- RA guard, on page 508
- Dhcpv6 guard, on page 510
- IPv6 snooping, on page 511
- How to configure IPv6 FHS, on page 512
- Configuration examples, on page 519
- Additional references for IPv6 First-Hop security, on page 520

About First-Hop security

First-Hop Security (FHS) is a set of features that optimize IPv6 link operation and provide protection from rogue or misconfigured users in large Layer 2 domains.

- FHS operates on Layer 2 and Layer 3 switches facing end nodes, often referred to as "first hops".
- FHS features help secure networks by preventing attacks from unauthorized or misconfigured devices.
- Extended FHS features can be used for different deployment scenarios or attack vectors.

Supported First-Hop security features

The following FHS features are supported:

- IPv6 RA Guard
- DHCPv6 Guard
- IPv6 Snooping

Note See Guidelines and limitations of First-Hop security, on page 502 for information about enabling this feature.

Note Use the **feature dhcp** command to enable the FHS features on a switch.

Enabling First-Hop security features

For example, to enable FHS features on a switch, use the **feature dhcp** command. This ensures that the supported features such as IPv6 RA Guard, DHCPv6 Guard, and IPv6 Snooping are active for end node protection.

IPv6 global policies

IPv6 global policies are mechanisms that provide storage and access policy database services for IPv6 features, including IPv6 Snooping, DHCPv6 Guard, and IPv6 RA Guard.

- Policy attributes are stored in the software policy database each time an IPv6 global policy is configured.
- When a policy is applied to an interface, the database entry is updated to include the interface.
- IPv6 global policies support features such as IPv6 Snooping, DHCPv6 Guard, and RA Guard.

IPv6 global policies are implemented and managed through specific commands and hardware configurations, and their behavior may vary depending on the platform.

- All port level FHS policies are programmed in the ifacl region, while VLAN level policies are programmed in the FHS region.
- Use the **hardware profile tcam regionfhs tcam_size** command to configure the FHS. The range for the TCAM size is 0-4096.
- Use the **hardware access-list tcam region ing-redirect tcam_size** command to configure FHS. You can resize the **ing-racl** region to allocate space to the **ing-redirect** region.
- Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches: FHS packets take the **copp-s-dhcpreq** queue for software processing.
- Cisco Nexus 9300, 9500 platform switches , N9K-X9432C-S line card use the class default.

IPv6 First-Hop security binding table

A database table of IPv6 neighbors connected to the device is created from information sources such as IPv6 snooping. This database, or binding table is used by various IPv6 guard features to validate the link-layer address (LLA), the IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

Guidelines and limitations of First-Hop security

Use this reference to review the guidelines and limitations of First-Hop security.

- First-Hop Security (FHS) requires **ing-redirect** TCAM carving before enabling FHS on an interface or VLAN.

- Use the **hardware access-list tcam region ing-redirect** *tcam_size* command to perform TCAM carving.
- You can resize the **ing-racl** region to allocate space to the **ing-redirect** region.

Supported platform and release for First-Hop security

The following table lists the supported software releases and corresponding Cisco Nexus switch platforms for First-Hop Security.

Table 101: Supported releases and platforms for First-Hop security

Supported Release	Supported Platform
7.0(3)I7(1) and later	Cisco Nexus 9300- EX/ FX/FX2 Series switches
9.3(5) and later	Cisco Nexus 9300-GX Series switches and N9K-C93180YC-FX3S switches
10.1(1) and later	CiscoNexus 9300-FX3 Series switches
10.2(3)Fand later	CiscoNexus 9300-GX2 Series switches
10.4(1)Fand later	CiscoNexus 9332D-H2R Series switches
10.4(2)Fand later	CiscoNexus 93400LD-H1 Series switches
10.4(3)Fand later	CiscoNexus 9364C-H1 Series switches

About vPC First-Hop security configuration

vPC First-Hop Security Configuration refers to the recommended methods for deploying IPv6 First-Hop Security in a vPC environment.

- Supports deployment of DHCP relay on-stack.
- Allows DHCP relay on vPC leg.
- Enables DHCP client and relay on orphan ports.

Best practice deployment scenarios for vPC First-Hop security

You can deploy IPv6 First-Hop Security vPC in several ways, with the following scenarios recommended as best practices.

- DHCP relay on-stack
- DHCP relay on vPC leg
- DHCP client and relay on orphan ports

vPC First-Hop security deployment

For example, in a typical deployment, you might configure DHCP relay on the vPC stack to ensure redundancy, or use orphan ports for DHCP client and relay roles when devices are not part of the vPC domain.

DHCP relay on-stack

DHCP Relay On-stack is a deployment scenario in which DHCP relay is configured on a Nexus switch, allowing clients to connect either directly behind the vPC link or through an intermediary switch. This setup enables efficient configuration of IPv6 Snooping at the interface level and optimizes control traffic processing.

- Clients can be connected directly behind the vPC link or through an intermediary switch with DHCP relay on the Nexus switch.
- IPv6 Snooping can be configured on vPC interface links directly, rather than at the VLAN level.
- Control traffic (DHCP/ND) is not redirected to the CPU for processing on both vPC peers if it traverses the peer link, and packets are not processed a second time.

Efficient DHCP relay deployment with vPC and IPv6 snooping

In this scenario, connecting clients behind an intermediary switch with DHCP relay running on the Nexus switch is ideal for efficient configuration and traffic processing.

- Direct client connection behind vPC link
- Client connection through intermediary switch with DHCP relay on Nexus switch

Configuration at the interface level is efficient for the following reasons:

1. Control traffic (DHCP/ND) is not redirected to CPU for processing on both vPC peers if it goes over the peer link.
2. Packets switched over the peer link are not processed a second time.

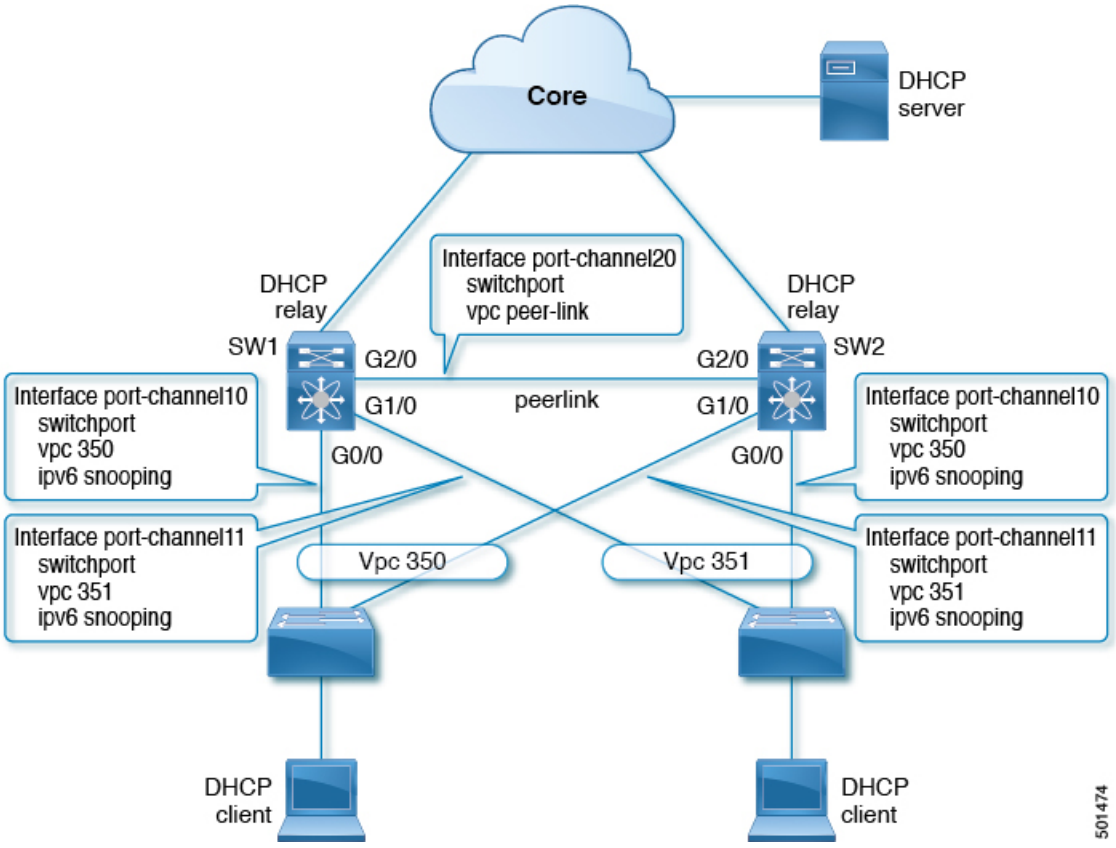
In this scenario, the two vPC peers learn all the host IP/MAC bindings behind the vPC links and synchronize these bindings between themselves using both IPv6 ND and IPv6 DHCP control protocols.



Note TIP: Configuring snooping policy at the interface level improves efficiency and reduces unnecessary CPU processing on vPC peers.

The following figure illustrates FHS configuration with DHCP relay on-stack:

Figure 14: FHS Configuration with DHCP relay on-stack



DHCP relay on-stack deployment

For example, when clients are connected through an intermediary switch with DHCP relay enabled on the Nexus switch, IPv6 Snooping can be applied directly to the vPC interfaces, allowing both vPC peers to learn and synchronize host IP/MAC bindings efficiently.

DHCP relay on vPC leg

DHCP relay on a vPC leg refers to a deployment where the DHCP relay agent or server is positioned behind a vPC link, rather than running directly on the vPC peers.

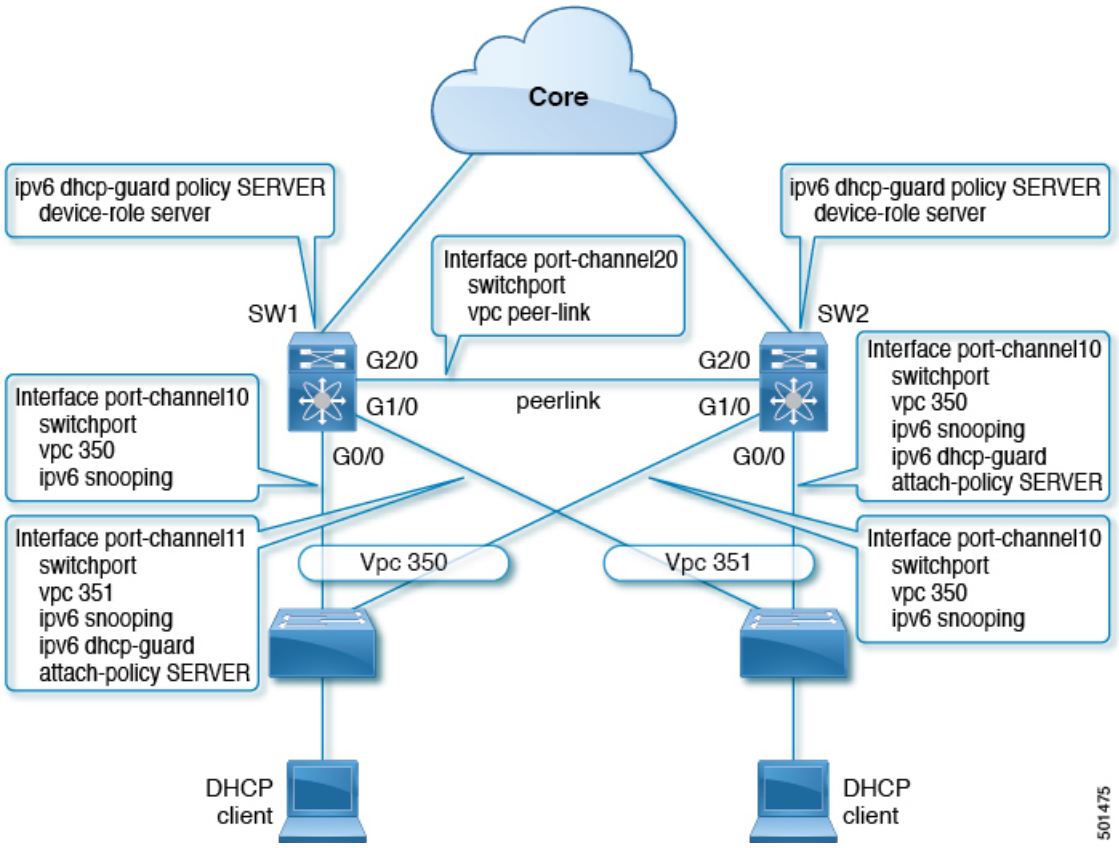
- The relay agent does not run on the vPC peers; it operates behind the vPC link.
- IPv6 Snooping does not implicitly trust DHCP server messages in this scenario and drops them by default.
- Custom IPv6 policies are required to allow DHCP server messages and ensure correct client binding creation.

In this configuration, the relay agent does not run on the vPC peers. Instead, the DHCP relay agent (or a DHCP server) runs behind a vPC link, which can be towards the access or even somewhere in the core. In such a deployment scenario, the IPv6 Snooping feature does not implicitly trust DHCP Server messages and drops DHCP Server messages by default.

- Security-level glean
- IPv6 DHCP Guard policy with device-role server. In this configuration, IPv6 Snooping trusts DHCP server messages attached to the vPC link.

Both server or relay facing and client facing IPv6 snooping policies are needed to create client binding entries via DHCP control traffic. IPv6 Snooping must see both client and server packets to create the binding. The IPv6 DHCP Guard policy must be configured to allow DHCP server traffic by the IPv6 Snooping policy. Both vPC peers require the same configuration because they synchronize all newly learned client entries on the vPC port.

Figure 15: FHS Configuration with external DHCP relay



Example: DHCP relay on vPC leg

For example, clients are located behind the vPC links with the default IPv6 Snooping policy. By attaching both IPv6 Snooping and ipv6 dhcp-guard attach-policy SERVER policies to the links where DHCP server traffic arrives, you ensure that both client and server packets are visible to IPv6 Snooping, allowing for proper client binding creation.

DHCP client relay on orphan ports

DHCP client relay on orphan ports refers to the process of handling DHCP client traffic when the client is connected via an orphan port, which is not directly connected to both vPC peers.

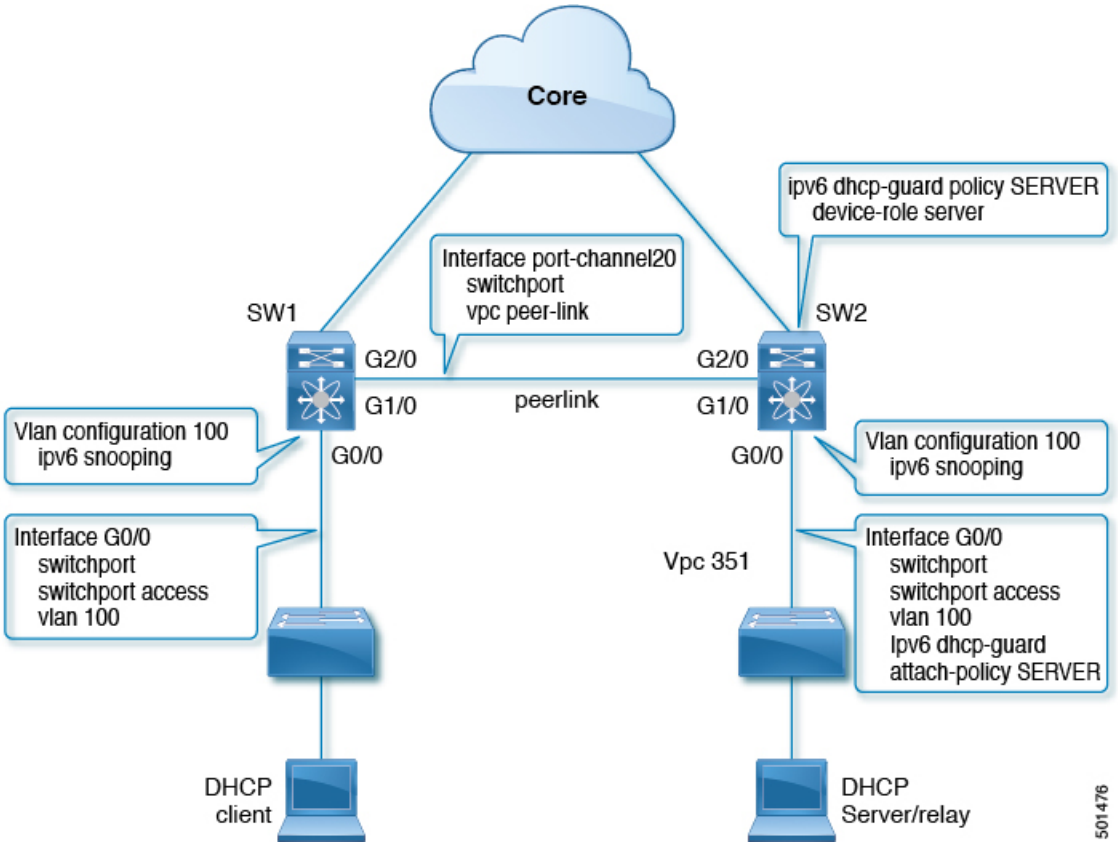
- IPv6 Snooping only syncs client bindings on vPC ports, not on orphan ports.
- On orphan ports, IPv6 Snooping runs independently on each switch.
- Special configuration is required to ensure DHCP server packets are properly handled on orphan ports.

Configuration and behavior of DHCP client relay on orphan ports

When a client is connected via an orphan port, IPv6 Snooping does not synchronize client bindings between vPC peers. Each switch manages its own bindings independently, and specific configuration steps are required to ensure correct DHCP traffic handling.

- Attach the IPv6 Snooping policy on the client-facing interface of the first switch.
- To accommodate DHCP server packets from an orphan port behind the vPC peer, apply the policy at the VLAN level. This allows inspection of both client and server traffic.
- DHCP traffic arriving via the vPC peer is implicitly trusted; if policing is required, the vPC peer automatically drops such traffic.
- On the second switch, configure IPv6 at the VLAN level and apply the IPv6 DHCP Guard policy with a “device-role server” on the server-facing orphan port to prevent dropping DHCP server packets.
- Both switches learn client binding entries individually and do not synchronize them, as the client is not on a vPC link.

Figure 16: FHS configuration with client and DHCP relay on orphan port



DHCP client relay on orphan ports

For example, if a client is connected to an orphan port, apply the IPv6 Snooping policy at the VLAN level on both switches and configure the DHCP Guard policy with the appropriate device role to ensure correct DHCP packet handling.

RA guard

Explains RA guard.

Overview of IPv6 RA guard

The IPv6 RA Guard feature is a security mechanism that allows network administrators to control and filter Router Advertisement (RA) messages on network devices.

- Blocks or rejects unwanted or rogue RA messages arriving at the device.
- Analyzes and filters RAs sent by unauthorized devices.
- Validates RA and router redirect messages against device configuration before forwarding or dropping them.

Supporting reference information for IPv6 RA guard

The IPv6 RA Guard feature operates by comparing configuration information on the Layer 2 (L2) device with the information found in received RA frames. In host mode, all RA and router redirect messages are disallowed on the port. If the L2 device validates the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If not validated, the RA is dropped.

IPv6 RA guard in operation

For example, when an unauthorized device attempts to send a rogue RA message, the IPv6 RA Guard feature detects and blocks the message, preventing it from reaching other devices on the network.

IPv6 RA router advertisement and the flags

IPv6 Router Advertisement (RA) is a mechanism that informs devices on a network how to configure their IPv6 addresses and obtain other network information using specific flags.

- The A (Address Autoconfiguration) flag enables hosts to use the specified prefix for IPv6 autoconfiguration.
- The O (Other Configuration) flag instructs hosts to obtain additional information, such as DNS server addresses, from a stateless DHCPv6 server.
- The M (Managed Address Configuration) flag directs hosts to use a stateful DHCPv6 server for their global unicast address and other configuration details.

- The L (On-Link) flag identifies that a specific prefix is on the local link or subnet, affecting how packets are routed.

IPv6 RA flags and their functions

IPv6 Router Advertisement messages use four primary flags to guide hosts in address configuration and network communication.

- **A flag (Address Autoconfiguration)** : Enabled by default; allows hosts to use the advertised prefix for IPv6 autoconfiguration.
- **O flag (Other Configuration)** : Disabled by default; instructs hosts to obtain additional configuration (such as DNS) from a stateless DHCPv6 server.
- **M flag (Managed Address Configuration)** : Disabled by default; tells hosts to use a stateful DHCPv6 server for address assignment and other information.
- **L flag (On-Link)** : Enabled by default; indicates that the prefix is on the local link or subnet.

1. Address Autoconfiguration (A flag) is enabled by default and allows hosts to use the prefix for autoconfiguration.
2. Other Configuration (O flag) is disabled by default and is used for obtaining additional information from a stateless DHCPv6 server.
3. Managed Address Configuration (M flag) is disabled by default and is used for stateful DHCPv6 address assignment.
4. On-Link (L flag) is enabled by default and identifies the prefix as being on the local link.



Note When the M flag is enabled, the A flag should usually be disabled. Manually enabling the M flag does not automatically disable the A flag. To disable the A flag, use the **ipv6 nd prefix** *ipv6-prefix/prefix-length* **no-autoconfig** command.

Using IPv6 RA flags

For example, enabling the **ipv6 managed-config-flag** command activates the M flag, instructing hosts to use a stateful DHCPv6 server for address assignment. By default, the A and L flags are advertised via ICMPv6 Router Advertisement (RA).

Guidelines and limitations of IPv6 RA guard

Use this reference to review the guidelines and limitations of IPv6 RA guard.

- The IPv6 RA Guard feature does not offer protection in environments where IPv6 traffic is tunneled.
- Beginning with Cisco NX-OS Release 10.1(1), IPv6 RA Guard is supported on Cisco Nexus 9300-GX platform switches.

- This feature is supported only in hardware when the ternary content addressable memory (TCAM) is programmed.
- This feature can be configured on a switch port interface in the ingress direction.
- This feature supports host mode and router mode.
- This feature is supported only in the ingress direction; it is not supported in the egress direction.
- This feature is supported on auxiliary VLANs and private VLANs (PVLANs). In the case of PVLANs, primary VLAN features are inherited and merged with port features.
- Packets dropped by the IPv6 RA Guard feature can be spanned.

Dhcpv6 guard

Explains dhcpv6 guard.

Overview of DHCP—DHCPv6 guard

The DHCPv6 Guard feature is a security mechanism that blocks DHCP reply and advertisement messages from unauthorized DHCP servers and relay agents.

- Blocks DHCP reply and advertisement messages from unauthorized sources.
- Allows client messages and relay agent messages from clients to servers without blocking.
- Filtering decisions are based on the device role assigned to the receiving switch port, trunk, or VLAN.

How DHCPv6 guard works

The DHCPv6 Guard feature determines whether to block or allow DHCP messages based on the type of message and the device role configuration.

- Packets are classified into three DHCP message types: client messages, server messages, and relay agent messages.
- All client messages are always switched, regardless of device role.
- DHCP server messages are only processed further if the device role is set to server.
- Further processing of DHCP server advertisements occurs for server preference checking.
- If the device is configured as a DHCP server, all messages are switched, regardless of device role configuration.

DHCPv6 guard in action

For example, if a switch port is configured with the DHCPv6 Guard feature and receives a DHCP advertisement from an unauthorized server, the message is blocked, preventing potential DoS attacks or traffic redirection.

Limitation of DHCPv6 guard

Use this reference to review the limitations of DHCPv6 guard.

DHCPv6 guard limitations

- If a packet arriving from a DHCP server is a Relay Forward or a Relay Reply, only the device role is checked.
- IPv6 DHCP Guard does not apply the policy for a packet sent out by the local relay agent running on the switch.

IPv6 snooping

Explains IPv6 snooping.

Overview of IPv6 snooping

IPv6 snooping is a Layer 2 security feature that learns and secures IPv6 address bindings, analyzes snooping messages, and builds a trusted binding table to mitigate vulnerabilities in neighbor discovery mechanisms.

- Operates at Layer 2 or between Layer 2 and Layer 3 to provide IPv6 security and scalability.
- Mitigates vulnerabilities such as attacks on duplicate address detection (DAD), address resolution, device discovery, and the neighbor cache.
- Builds and maintains a trusted binding table by analyzing snooping and DHCPv6 messages and enforcing address ownership.

Supporting reference information for IPv6 snooping

IPv6 snooping provides several mechanisms to secure IPv6 address assignments and neighbor discovery at Layer 2, supporting accurate binding tables and integration with other security features.

- Secures stateless autoconfiguration addresses in Layer 2 neighbor tables.
- Analyzes snooping messages to build a trusted binding table and drops messages without valid bindings.
- Redirects snooping protocol and DHCPv6 traffic to the switch integrated security features (SISF) infrastructure for processing.
- Registers capture rules to the classifier, which aggregates and installs rules as ACLs in platform-dependent modules.
- Provides host liveness tracking to update neighbor tables when IPv6 hosts disappear.
- Enforces address ownership and limits the number of addresses a node can claim.

IPv6 snooping in operation

For example, when IPv6 snooping is enabled on a switch port, the device inspects Neighbor Discovery Protocol (NDP) and DHCPv6 messages, verifies IPv6-to-MAC mappings, and updates the binding

table. If a message does not have a valid binding, it is dropped, preventing unauthorized address usage.

Guidelines and limitations for IPv6 snooping

Use this reference to review the guidelines and limitations for IPv6 snooping.

The guidelines and limitations of IPv6 Snooping are as follows:

- You must perform the same configurations on both the vPC peers. Automatic consistency checker for IPv6 snooping is not supported.
- The IPv6 Snooping feature is supported only in hardware when the ternary content addressable memory (TCAM) is programed.
- This feature can be configured on a switch port interface or VLAN only on the ingress port.
- For IPv6 Snooping to learn DHCP bindings, it must see both server and client replies. A IPv6 snooping policy must be attached to both the client facing the interface (or VLAN) as well as the DHCP server facing interface (or VLAN). In the case of DHCP Relay, an IPv6 Snooping policy must be attached at the VLAN level to see the server replies.

How to configure IPv6 FHS

Explains how to configure IPv6 FHS.

Configure the IPv6 RA guard policy on the device



Note When the **ipv6 nd raguard** command is configured on ports, router solicitation messages are not replicated to these ports. To replicate router solicitation messages, all ports that face routers must be set to the router role.

Procedure

Step 1 Use the **ipv6 nd raguard policy** *policy-name* command to define the RA guard policy name and enters RA guard policy configuration mode.

Example:

```
switch#configure terminal
switch(config)#ipv6 nd raguard policy policy1
```

Step 2 Use the **device-role** {**host** | **router** | **monitor** | **switch**} command to specify the role of the device attached to the port.

Example:

```
switch(config-raguard-policy)#device-role router
```

- device-role host —Interface or VLAN where you connect a regular node or host. This where you apply the IPV6 RA Guard policy. The device-role host allows incoming RS packets, and blocks incoming RA or RR packets. RS packets that are received on another interface, are not redirected to the device-role host. Only RA and RR packets (that are allowed) are redirected to the device-role host.
- device-role switch —The device-role switch behaves similar to the device-role host. For example, you can use it as a label for a trunk port.
- device-role monitor —This device monitors network traffic. It behaves similar to the device-role host, except that RS packets are also sent to this interface. This helps capture traffic.
- device-role router —Interface that connects to the router. This interface allows incoming RS, RA, or RR packets.

Step 3 (Optional) Use the **hop-limit** {**maximum** | **minimum** *limit*} command to enable verification of the advertised hop count limit.

Example:

```
switch(config-raguard-policy)#hop-limit minimum 3
```

If not configured, this check will be bypassed.

Step 4 (Optional) Use the **managed-config-flag** {**on** | **off**} command to enable verification that the advertised managed address configuration flag is on.

Example:

```
switch(config-raguard-policy)#managed-config-flag on
```

Note
When enabling the M flag, it is recommended to disable the A flag.

If not configured, this check will be bypassed.

Step 5 (Optional) Use the **other-config-flag** {**on** | **off**} command to enable verification of the advertised *other* configuration parameter.

Example:

```
switch(config-raguard-policy)#other-config-flag on
```

Step 6 (Optional) Use the **router-preference maximum** {**high** | **low** | **medium**} command to enable verification that the advertised default router preference parameter value is lower than or equal to a specified limit.

Example:

```
switch(config-raguard-policy)#router-preference maximum high
```

Step 7 (Optional) Use the **trusted-port** command to specify that this policy is being applied to trusted ports.

Example:

```
switch(config-raguard-policy)#trusted-port
switch(config-raguard-policy)#exit
```

All RA guard policing will be disabled.

Configure IPv6 RA guard on an interface

Procedure

Step 1 **interface** *type number*

Example:

```
switch#configure terminal
switch(config)#interface ethernet 1/1
switch(config-if)#
```

Example:

```
switch(config)#vlan configuration 10
```

Specifies an interface type and number, and places the device in interface or VLAN configuration mode.

Step 2 **ipv6 nd raguard attach-policy** [*policy-name*]

Example:

```
switch(config-if)#ipv6 nd raguard attach-policy policy1
switch(config-if)#exit
switch(config)#exit
switch#exit
```

Applies the IPv6 RA Guard feature to a specified interface.

Step 3 **show ipv6 nd raguard policy** [*policy-name*]

Example:

```
switch#show ipv6 nd raguard policy host
Policy host configuration:
  device-role host
Policy applied on the following interfaces:
  Et0/0        vlan all
  Et1/0        vlan all
```

Displays the RA guard policy on all interfaces configured with the RA guard.

Step 4 **debug ipv6 snooping raguard** [*filter* | *interface* | *vlanid*]

Example:

```
switch#debug ipv6 snooping raguard
```

Enables debugging for IPv6 RA guard snooping information.

Configure DHCPv6 guard

Procedure

Step 1 **ipv6 dhcp guard policy** *policy-name*

Example:

```
Device#configure terminal
Device(config)#ipv6 dhcp guard policy pol1
Device(config-dhcpg-policy)#device-role server
```

Defines the DHCPv6 guard policy name and enters DHCP guard configuration mode and Specifies the device role of the device attached to the target (interface or VLAN).

- **device-role client** —Interface where a normal DHCPv6 client is connected. It blocks any incoming server packets.
- **device-role server** —Interface where a normal DHCPv6 server is connected. It allows all DHCPv6 packets originating on this interface.

Step 2 **preference min** *limit*

Example:

```
Device(config-dhcpg-policy)#preference min 0
Device(config-dhcpg-policy)#preference max 255
```

(Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. If not specified, this check will be bypassed.

(Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. If not specified, this check will be bypassed.

Step 3 **trusted-port**

Example:

```
Device(config-dhcpg-policy)#trusted-port
Device(config-dhcpg-policy)#exit
```

(Optional) Specifies that this policy is being applied to trusted ports. All DHCP guard policing will be disabled.

Step 4 **interface** *type number*

Example:

```
Device(config)#interface Ethernet 1/1
```

Specifies an interface and enters interface configuration mode.

Step 5 **switchport**

Example:

```
Device(config-if)#switchport
```

Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.

Step 6 **ipv6 dhcp guard** [**attach-policy** *policy-name*]

Example:

```
Device(config-if)#ipv6 dhcp guard attach-policy pol1
Device(config-if)#exit
```

Attaches a DHCPv6 guard policy to an interface.

Step 7 **vlan configuration** *vlan-id*

Example:

```
Device(config)#vlan configuration 1
```

Specifies a VLAN and enters VLAN configuration mode.

Step 8 **ipv6 dhcp guard** [**attach-policy** *policy-name*]

Example:

```
Device(config-vlan-config)#ipv6 dhcp guard attach-policy pol1
Device(config-vlan-config)#exit
```

Attaches a DHCPv6 guard policy to a VLAN.

Step 9 (Optional) **show ipv6 dhcp guard policy** [*policy-name*]

Example:

```
Device#show ipv6 dhcp policy guard pol1
```

Displays the policy configuration as well as all the interfaces where the policy is applied.

Configure an IPv6 snooping policy

Create and apply IPv6 snooping policies to control IPv6 address bindings and improve security.

IPv6 snooping policies help monitor, restrict, and secure IPv6 address assignment on a network by tracking bindings and setting appropriate controls.

Before you begin

- Ensure you have access privileges for configuration.
- Identify the interfaces or VLANs requiring IPv6 snooping policies.
- Determine policy settings: address limits, roles, protocols, security level.

Procedure

Step 1 Enter configuration mode and create a new IPv6 snooping policy.

Example:

```
Device#configure terminal
Device(config)#ipv6 snooping policy policy1
```

Enters IPv6 snooping configuration mode.

Step 2 Assign the device role (node or switch) for the policy.

Example:

```
Device(config-snoop-policy)#device-role node
```

```
Device(config-snoop-policy)#device-role switch
```

- Node: Default—creates bindings and probes entries.
- Switch: Does not probe entries or create bindings if trusted port is enabled.

Step 3 Set a limit on address binding entries, or remove the limit if desired.

Example:

```
Device(config-snoop-policy)#limit address-count 500
Device(config-snoop-policy)#no limit address-count
```

Limits or removes binding entry restriction.

Step 4 Enable protocol monitoring for DHCP or NDP as needed.

Example:

```
Device(config-snoop-policy)#protocol dhcp
Device(config-snoop-policy)#protocol ndp
```

Step 5 Apply the policy to trusted ports if required.

Example:

```
Device(config-snoop-policy)#trusted-port
```

Traffic is not blocked or dropped on trusted ports.

Step 6 Set the desired security level for the policy.

Example:

```
Device(config-snoop-policy)#security-level glean
Device(config-snoop-policy)#security-level inspect
Device(config-snoop-policy)#security-level guard
```

- Glean: Learns bindings, doesn't drop packets.
- Inspect: Learns bindings, drops packets if issues detected.
- Guard: Drops packets on threat (e.g., IPv6, ND, RA, DHCP Server).

Step 7 Enable tracking and exit policy configuration mode.

Example:

```
Device(config-snoop-policy)#tracking enable
Device(config-snoop-policy)#exit
```

Step 8 Attach the policy to an interface or VLAN.

Example:

Interface:

```
Device(config)#interface ethernet 1/25
Device(config-if)#ipv6 snooping attach-policy policy1
Device(config-if)#exit
```

VLAN:

```
Device(config)#vlan configuration 333
Device(config-vlan-config)#ipv6 snooping attach-policy policy1
Device(config-vlan-config)#exit
```

Step 9 Verify the configuration and attached interfaces/VLANs.

Example:

```
Device#show ipv6 snooping policy policy1
```

Displays policy configuration and usage.

IPv6 snooping policy is configured and applied, securely managing address bindings and network traffic.

What to do next

Monitor logs and snooping reports to ensure the policy is effective.

Verify and troubleshoot IPv6 snooping

Procedure

Step 1 **show ipv6 snooping capture-policy** [**interface** *type number*]

Example:

```
Device#show ipv6 snooping capture-policy interface ethernet 0/0
```

Displays snooping message capture policies.

Step 2 **show ipv6 snooping counter** [**interface** *type number*]

Example:

```
Device#show ipv6 snooping counter interface Ethernet 1/1
```

Displays information about the packets counted by the interface counter.

Step 3 **show ipv6 snooping features**

Example:

```
Device#show ipv6 snooping features
```

Displays information about snooping features configured on the device.

Step 4 **show ipv6 snooping policies** [**interface** *type number*]

Example:

```
Device#show ipv6 snooping policies
```

Displays information about the configured policies and the interfaces to which they are attached.

Step 5 **debug ipv6 snooping**

Example:

```
Device#debug ipv6 snooping
```

Enables debugging for snooping information in IPv6.

Configuration examples

Explains configuration examples.

Example: IPv6 RA guard configuration

This example demonstrates the configuration of IPv6 RA Guard on an interface and how to display the running configuration.

Configutation

```
Device(config)#interface ethernet 1/1
Device(config-if)#ipv6 nd raguard attach-policy policy1
```

Configutation output

```
Device#show running-config interface ethernet 1/1
Building configuration...
Current configuration : 129 bytes
!
interface ethernet1/1
 switchport
 switchport access vlan 222
 switchport mode access
 access-group mode prefer port
 ipv6 nd raguard
end
```

Example: configuring DHCPv6 guard

The following example displays a sample configuration for DHCPv6 guard.

DHCPv6 guard configuration example

```
configure terminal
ipv6 dhcp guard policy pol1
device-role server
preference min 0
preference max 255
trusted-port
interface Ethernet 1/1
switchport
ipv6 dhcp guard attach-policy pol1
vlan configuration 1
ipv6 dhcp guard attach-policy pol1
show ipv6 dhcp guard policy pol1
```

Example: configuring IPv6 First-Hop security binding table

This example demonstrates the configuration of the IPv6 First-Hop Security Binding table.

```
configure terminal
 ipv6 neighbor binding vlan 100 2001:db8::1 interface ethernet3/0
 ipv6 neighbor binding max-entries 100
 ipv6 neighbor binding logging
 ipv6 neighbor binding retry-interval 8
```

```
 exit
show ipv6 neighbor binding
```

Example: configuring IPv6 snooping

This example demonstrates the configuration steps for IPv6 snooping and how to verify the applied policy.

Configuration

```
Device#configure terminal
Device(config)#ipv6 snooping policy policy1
Device(config-ipv6-snooping)#exit
Device(config)#vlan configuration 333
Device(config-vlan-config)#switchport
Device(config-vlan-config)#ipv6 snooping attach-policy policy1
```

Configuration output

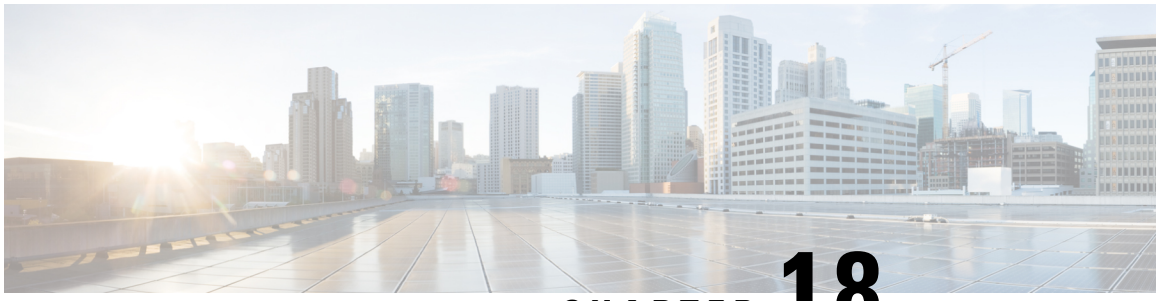
```
Device#show ipv6 snooping policies policy1
Policy policy1 configuration:
  trusted-port
  device-role node
Policy applied on the following interfaces:
   Et0/0                vlan all
   Et1/0       vlan all
Policy applied on the following vlans:
   vlan 1-100,200,300-400
```

Additional references for IPv6 First-Hop security

This section includes additional information related to configuring IPv6 First-Hop Security.

Table 102: Related documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Security Command Reference*



CHAPTER 18

Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) on a Cisco NX-OS device.

This chapter includes the following sections:

- About DAI, on page 521
- Prerequisites for DAI, on page 525
- Guidelines and limitations for DAI, on page 525
- Guidelines and limitations for DHCP relay with DAI, on page 526
- Default settings for DAI, on page 526
- Configuring DAI, on page 527
- DAI configuration verification commands, on page 532
- Monitoring and clearing DAI statistics, on page 532
- Configuration examples for DAI, on page 532
- Examples for DHCP relay with DAI, on page 537
- Additional references for DAI, on page 537

About DAI

Explains DAI.

ARP

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, host B wants to send information to host A but does not have the MAC address of host A in its ARP cache. In ARP terms, host B is the sender and host A is the target.

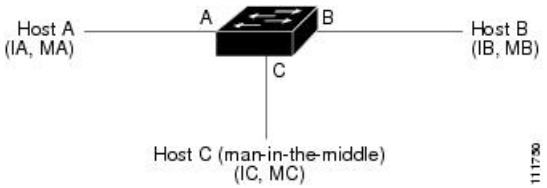
To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

ARP spoofing attacks

ARP spoofing attacks and ARP cache poisoning can occur because ARP allows a reply from a host even if an ARP request was not received. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

An ARP spoofing attack can affect hosts, switches, and routers connected to your Layer 2 network by sending false information to the ARP caches of the devices connected to the subnet. Sending false information to an ARP cache is known as ARP cache poisoning. Spoof attacks can also intercept traffic that is intended for other hosts on the subnet.

Figure 17: ARP cache poisoning



Hosts A, B, and C are connected to the device on interfaces A, B, and C, which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, host A uses IP address IA and MAC address MA. When host A needs to send IP data to host B, it broadcasts an ARP request for the MAC address that is associated with IP address of IB. When host B receives the ARP request, the ARP cache on host B is populated with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When host B responds and the response reaches host A, the ARP cache on host A is populated with an ARP binding for a host with the IP address IB and MAC address MB. The device in between does not populate the ARP cache as both the request and the response are not destined to its local IP address.

Host C can poison the ARP caches of host A and host B by broadcasting two forged ARP responses with bindings: one for a host with the IP address of IA, a MAC address of MC, and another for a host with an IP address of IB and a MAC address of MC. Host B then uses the MAC address MC as the destination MAC address for traffic intended for IA, which means that host C intercepts that traffic. Similarly, host A uses MAC address MC as the destination MAC address for traffic intended for IB.

Because host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. This topology, in which host C has inserted itself into the traffic stream from host A to host B, is an example of a *man-in-the middle* attack.

DAI and ARP spoofing attacks

DAI ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, a Cisco Nexus device performs these activities:

- Intercepts all ARP requests and responses on untrusted ports
- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

DAI can determine the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a Dynamic Host Configuration Protocol (DHCP) snooping binding database. This database can also contain

static entries that you create. If the ARP packet is received on a trusted interface, the device forwards the packet without any checks. On untrusted interfaces, the device forwards the packet only if it is valid.

You can configure DAI to drop ARP packets when the IP addresses in the packets are invalid or when the MAC addresses in the body of the ARP packets do not match the addresses specified in the Ethernet header.

Interface trust states and network security

DAI associates a trust state with each interface on the device. Packets that arrive on trusted interfaces bypass all DAI validation checks, and packets that arrive on untrusted interfaces go through the DAI validation process.

In a typical network configuration, the guidelines for configuring the trust state of interfaces are as follows:

- Untrusted

  Interfaces that are connected to hosts

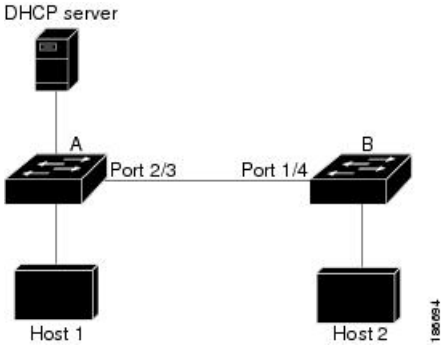
- Trusted

  Interfaces that are connected to devices

With this configuration, all ARP packets that enter the network from a device bypass the security check. No other validation is needed at any other place in the VLAN or in the network.

Caution Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

Figure 18: ARP packet validation on a VLAN enabled for DAI



If you configure interfaces as trusted when they should be untrusted, you may open a security hole in a network. If device A is not running DAI, host 1 can easily poison the ARP cache of device B (and host 2, if you configured the link between the devices as trusted). This condition can occur even though device B is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a device that runs DAI do not poison the ARP caches of other hosts in the network; however, DAI does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a device that runs DAI.

If some devices in a VLAN run DAI and other devices do not, the guidelines for configuring the trust state of interfaces on a device that runs DAI become the following:

- **Untrusted**

  Interfaces that are connected to hosts or to devices that are not running DAI

- **Trusted**

  Interfaces that are connected to devices that are running DAI

When you cannot determine the bindings of packets from devices that do not run DAI, isolate at Layer 3 the devices that run DAI from devices that do not run DAI.



Note Depending on your network setup, you may not be able to validate a given ARP packet on all devices in the VLAN.

Logging DAI packets

Cisco NX-OS maintains a buffer of log entries about DAI packets processed. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You can also specify the type of packets that are logged. By default, a Cisco Nexus device logs only packets that DAI drops.

If the log buffer overflows, the device overwrites the oldest DAI log entries with newer entries. You can configure the maximum number of entries in the buffer.



Note Cisco NX-OS does not generate system messages about DAI packets that are logged.

DHCP relay with dynamic ARP inspection

DAI uses DHCP snooping client binding database to validate the ARP packets. In releases earlier than Cisco NX-OS Release 10.1(1), this database was built by the DHCP Snooping process, which runs on the switch. The binding database isn't built when the switch acts as a DHCP relay. When snooping, DHCP relay and DAI are enabled together, the relay process takes precedence over snooping for processing incoming DHCP packets. Hence, snooping doesn't build the binding database. Since DAI depends on the binding database, it can't operate with DHCP relay. However, from Cisco NX-OS Release 10.1(1), you can build the binding database using DHCP relay DAI.

When a switch receives a DHCP request, a temporary binding entry is created consisting of the client's MAC address, VLAN, and the incoming interface. After receiving DHCPACK from the server, the binding entry is qualified. The offered IP address is added to the qualified temporary entry and the binding entry type is updated as dhcp-relay.

When you upgrade to Cisco NX-OS Release 10.1(1) or a later release and if you enable this feature, the ISSU proceeds without any error. Disable this feature before you downgrade from Cisco NX-OS Release 10.1(1) to an earlier release.

Prerequisites for DAI

Use this reference to review the prerequisites for DAI.

The prerequisites for DAI are as follows:

- You must enable the DHCP feature before you can configure DAI. See Configuring DHCP, on page 441.
- You must configure the VLANs on which you want to enable DAI. See the *Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide*.
- You must configure the ACL TCAM region size for DAI using the **hardware access-list tcam region arp-ether** command. The DAI configuration will not be accepted unless the arp-ether region is effective. See Configure ACL TCAM region sizes, on page 344.

Guidelines and limitations for DAI

Use this reference to review the guidelines and limitations for DAI.

DAI behavior

DAI has the following configuration guidelines and limitations:

- DAI is an ingress security feature; it does not perform any egress checking.
- DAI is not effective for hosts connected to devices that do not support DAI or that do not have this feature enabled. Because man-in-the-middle attacks are limited to a single Layer 2 broadcast domain, you should separate the domain with DAI from domains without DAI. This separation secures the ARP caches of hosts in the domain with DAI.

DHCP configuration guidelines

- When you use the **feature dhcp** command to enable the DHCP feature, there is a delay of approximately 30 seconds before the I/O modules receive the DHCP or DAI configuration. This delay occurs regardless of the method that you use to change from a configuration with the DHCP feature disabled to a configuration with the DHCP feature enabled. For example, if you use the rollback feature to revert to a configuration that enables the DHCP feature, the I/O modules receive the DHCP and DAI configuration approximately 30 seconds after you complete the rollback.

Interface and port-channel guidelines

- DAI is supported on access ports, trunk ports, and port-channel ports.
- The DAI trust configuration of a port channel determines the trust state of all physical ports that you assign to the port channel. For example, if you have configured a physical port as a trusted interface and then you add that physical port to a port channel that is an untrusted interface, the physical port becomes untrusted.
- When you remove a physical port from a port channel, the physical port does not retain the DAI trust state configuration of the port channel.

- When you change the trust state on the port channel, the device configures a new trust state on all the physical ports that comprise the channel.

Binding guidelines

- If you want DAI to use static IP-MAC address bindings to determine if ARP packets are valid, make sure that you have configured the static IP-MAC address bindings.
- ARP ACLs are not supported.
- If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, make sure that DHCP snooping is enabled.

Platform support

Beginning with Cisco NX-OS Release 9.3(3), DAI is supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

Guidelines and limitations for DHCP relay with DAI

Use this reference to review the guidelines and limitations for DHCP relay with DAI.

Supported platforms

The following Cisco Nexus platform switches support this feature:

- Cisco Nexus 9200 platform switches
- Cisco Nexus 9300-EX platform switches
- Cisco Nexus 9300-FX platform switches

Binding database limitations

- The binding database entries are not stored in the hardware.
- The binding database is common for all VRFs. If there are multiple VRFs, map each VRF to a unique VLAN.
- Only IPv4 entries are stored in the binding database. IPv6 is not supported.

Unsupported features

- IP Source Guard (IPSG) does not support this feature.
- This feature does not support vPC.

Default settings for DAI

Use this reference to review the default settings for DAI.

This table lists the default settings for DAI parameters.

Table 103: Default DAI parameters

Parameters	Default
DAI	Disabled on all VLANs.
Interface trust state	All interfaces are untrusted.
Validation checks	No checks are performed.
Log buffer	When DAI is enabled, all denied or dropped ARP packets are logged. The number of entries in the log is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second.
Per-VLAN logging	All denied or dropped ARP packets are logged.

Configuring DAI

Explains configuring DAI.

Enable or disable DAI on VLANs

You can enable or disable DAI on VLANs. By default, DAI is disabled on all VLANs.

Before you begin

- Make sure that the DHCP feature is enabled.
- Make sure that the VLANs on which you want to enable DAI are configured.
- Make sure that the ACL TCAM region size for DAI (arp-ether) is configured.

Procedure

Step 1 Use the [**no**] **ip arp inspection vlan** *vlan-list* command to enable DAI for the specified list of VLANs. The **no** option disables DAI for the specified VLANs.

Example:

```
switch#configure terminal
switch(config)#ip arp inspection vlan 13
```

Step 2 (Optional) Use the **show ip arp inspection vlan** *vlan-id* command to display the DAI configuration for a specific VLAN.

Example:

```
switch#show ip arp inspection vlan 13
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure the DAI trust state of a layer 2 interface

You can configure the DAI interface trust state of a Layer 2 interface. By default, all interfaces are untrusted.

A device forwards ARP packets that it receives on a trusted Layer 2 interface but does not check them.

On untrusted interfaces, the device intercepts all ARP requests and responses and verifies that the intercepted packets have valid IP-MAC address bindings before updating the local cache and forwarding the packet to the appropriate destination. If the device determines that packets have invalid bindings, it drops the packets and logs them according to the logging configuration.

Before you begin

If you are enabling DAI, make sure that the DHCP feature is enabled.

Procedure

Step 1 Use the **interface** *type port/slot* command to enter interface configuration mode.

Example:

```
switch#configure terminal
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Step 2 Use the [**no**] **ip arp inspection trust** command to configure the interface as a trusted ARP interface. The **no** option configures the interface as an untrusted ARP interface.

Example:

```
switch(config-if)#ip arp inspection trust
```

Step 3 (Optional) Use the **show ip arp inspection interface** *type port/slot* command to display the trust state and the ARP packet rate for the specified interface.

Example:

```
switch#show ip arp inspection interface ethernet 2/1
```

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Enable or disable additional validation

You can enable or disable additional validation of ARP packets. By default, no additional validation of ARP packets is enabled. When no additional validation is configured, the source MAC address and the source IP address check against the IP-to-MAC binding entry for ARP packets is performed by using the ARP sender MAC address and the ARP sender IP address.

DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can enable additional validation on the destination MAC address, the sender and target IP addresses, and the source MAC address.

You can use the following keywords with the **ip arp inspection validate** command to implement additional validations:

- **dst-mac**

  Checks the destination MAC address in the Ethernet header against the target MAC address in the ARP body for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

- **ip**

  Checks the ARP body for invalid and unexpected IP addresses. Addresses include 192.0.2.1, 192.0.2.2, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses.

- **src-mac**

  Checks the source MAC address in the Ethernet header against the sender MAC address in the ARP body for ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

When enabling additional validation, follow these guidelines:

- You must specify at least one of the keywords. You can specify one, two, or all three keywords.
- Each **ip arp inspection validate** command that you enter replaces the configuration from any previous commands. If you enter an **ip arp inspection validate** command to enable src-mac and dst-mac validations, and a second **ip arp inspection validate** command to enable ip validation, the src-mac and dst-mac validations are disabled when you enter the second command.

Procedure

Step 1 Use the [**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]} command to enable additional DAI validation. The **no** form of this command disables additional DAI validation.

Example:

```
switch#configure terminal
switch(config)#ip arp inspection validate src-mac dst-mac ip
```

Step 2 (Optional) Use the **show running-config dhcp** command to display the DHCP snooping configuration, including the DAI configuration.

Example:

```
switch#show running-config dhcp
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure the DAI logging buffer size

You can configure the DAI logging buffer size. The default buffer size is 32 messages.

Procedure

Step 1 Use the [**no**] **ip arp inspection log-buffer entries** *number* command to configure the DAI logging buffer size. The **no** option reverts to the default buffer size, which is 32 messages. The buffer size can be between 1 and 1024 messages.

Example:

```
switch#configure terminal
switch(config)#ip arp inspection log-buffer entries 64
```

Step 2 (Optional) Use the **show running-config dhcp** command to display the DHCP snooping configuration, including the DAI configuration.

Example:

```
switch#show running-config dhcp
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure DAI log filtering

You can configure how the device determines whether to log a DAI packet. By default, the device logs DAI packets that are dropped.

Procedure

Step 1 Use the [**no**] **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings** {**all** | **none** | **permit**} command to configure DAI log filtering, as follows. The **no** form of this command removes DAI log filtering.

Example:

```
switch#configure terminal
switch(config)#ip arp inspection vlan 100 dhcp-bindings permit
```

- **all** —Logs all packets that match DHCP bindings.
- **none** —Does not log packets that match DHCP bindings.
- **permit** —Logs packets permitted by DHCP bindings.

Step 2 (Optional) Use the **show running-config dhcp** command to display the DHCP snooping configuration, including the DAI configuration.

Example:

```
switch#show running-config dhcp
```

Step 3 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Enable DHCP relay with DAI

You can create the binding database when DHCP relay and DAI are enabled. This feature is disabled by default.

Before you begin

Enable DAI and DHCP relay. Enable DHCP snooping globally and on VLAN. See the *Configuring DHCP* chapter for more information.

Procedure

Step 1 Use the **ip dhcp relay dai** command to enable creation of binding database in the relay.

Example:

```
switch#configure terminal
switch(config)#ip dhcp relay dai
```

Step 2 (Optional) Use the **show ip dhcp snooping binding relay** command to display the binding entries of the dhcp-relay type.

Example:

```
switch#show ip dhcp snooping binding relay
```

Step 3 (Optional) Use the **show system internal dhcp database global config** command to display if the relay DAI feature is enabled or not.

Example:

```
switch#show system internal dhcp database global config
```

DAI configuration verification commands

Use this reference to verify the DAI configuration.

To display the DAI configuration information, perform one of the following tasks.

Table 104: DAI configuration verification commands

Command	Purpose
show ip arp inspection	Displays the status of DAI.
show ip arp inspection interfaces [**ethernet** *slot/port* | **port-channel** *number*]	Displays the trust state and ARP packet rate for a specific interface or port channel.
show ip arp inspection log	Displays the DAI log configuration.
show ip arp inspection vlan *vlan-id*	Displays the DAI configuration for a specific VLAN.
show running-config dhcp [**all**]	Displays the DAI configuration.

Monitoring and clearing DAI statistics

To monitor and clear DAI statistics, use the commands in this table:

Table 105: Monitoring and clearing DAI statistics

Command	Purpose
show ip arp inspection statistics [**vlan** *vlan-id*]	Displays DAI statistics.
clear ip arp inspection statistics **vlan** *vlan-id*	Clears DAI statistics.
clear ip arp inspection log	Clears DAI logs.

Configuration examples for DAI

Explains configuration examples for DAI.

Two devices support DAI

These procedures show how to configure DAI when two devices support DAI.

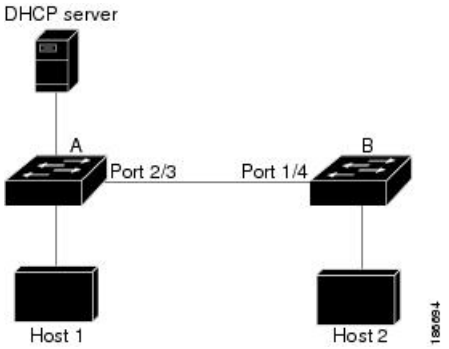
DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically-assigned IP addresses.

- This configuration does not work if the DHCP server is moved from device A to a different location.

- To ensure that this configuration does not compromise security, configure Ethernet interface 2/3 on device A and Ethernet interface 1/4 on device B as trusted.

The following figure shows the network configuration for this example. Host 1 is connected to device A, and Host 2 is connected to device B. Both devices are running DAI on VLAN 1 where the hosts are located. A DHCP server is connected to device A. Both hosts acquire their IP addresses from the same DHCP server. Device A has the bindings for Host 1 and Host 2, and device B has the binding for Host 2. Device A Ethernet interface 2/3 is connected to device B Ethernet interface 1/4.

Figure 19: Two devices supporting DAI



Configure device a

To enable DAI and configure Ethernet interface 2/3 on device A as trusted, follow these steps:

Procedure

Step 1 While logged into device A, verify the connection between device A and device B.

```
switchA#show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID              Local Intrfce   Hldtme  Capability  Platform      Port ID
switchB                Ethernet2/3     177     R S I       WS-C2960-24TC Ethernet1/4
switchA#
```

Step 2 Enable DAI on VLAN 1 and verify the configuration.

```
switchA#configure terminal
switchA(config)#ip arp inspection vlan 1
switchA#show ip arp inspection vlan 1
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan : 1
-----------
Configuration   : Enabled
Operation State : Active
switchA(config)#
```

Step 3 Configure Ethernet interface 2/3 as trusted.

```
switchA(config)#interface ethernet 2/3
switchA(config-if)#ip arp inspection trust
switchA(config-if)#exit
switchA(config)#exit
switchA#show ip arp inspection interface ethernet 2/3
 Interface        Trust State    Rate (pps)    Burst Interval
 -------------    -----------    ----------    --------------
 Ethernet2/3      Trusted           15             5
```

Step 4 Verify the bindings.

```
switchA#show ip dhcp snooping binding
MacAddress          IpAddress        LeaseSec  Type           VLAN  Interface
------------------  ---------------  --------  -------------  ----  -------------
00:00:5E:00:53:01  10.0.0.1         0         dhcp-snooping  1     Ethernet2/3
switchA#
```

Step 5 Check the statistics before and after DAI processes any packets.

```
switchA#show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 0
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchA#
```

If host 1 sends out two ARP requests with an IP address of 10.0.0.1 and a MAC address of 0002.0002.0002, both requests are permitted and are shown as follows:

```
switchA#show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 2
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
```

If host 1 tries to send an ARP request with an IP address of 10.0.0.3, the packet is dropped, and an error message is logged.

```
00:12:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 2 Invalid ARPs (Req) on Ethernet2/3, vlan
1.([0002.0002.0002/10.0.0.3/0000.0000.0000/192.0.2.1/02:42:35 UTC Fri Jan 23 2015])
```

The statistics display as follows:

```
switchA#show ip arp inspection statistics vlan 1
switchA#Vlan : 1
-----------
ARP Req Forwarded  = 2
```

```
ARP Res Forwarded  = 0
ARP Req Dropped    = 2
ARP Res Dropped    = 0
DHCP Drops         = 2
DHCP Permits       = 2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchA#
```

Configure device b

To enable DAI and configure Ethernet interface 1/4 on device B as trusted, follow these steps:

Procedure

Step 1 While logged into device B, verify the connection between device B and device A.

```
switchB#show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID              Local Intrfce   Hldtme  Capability  Platform      Port ID
switchA                Ethernet1/4     120     R S I       WS-C2960-24TC Ethernet2/3
switchB#
```

Step 2 Enable DAI on VLAN 1 and verify the configuration.

```
switchB#configure terminal
switchB(config)#ip arp inspection vlan 1
switchB#show ip arp inspection vlan 1
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan : 1
-----------
Configuration   : Enabled
Operation State : Active
switchB(config)#
```

Step 3 Configure Ethernet interface 1/4 as trusted.

```
switchB(config)#interface ethernet 1/4
switchB(config-if)#ip arp inspection trust
switchB(config-if)#exit
switchB(config)#exit
switchB#show ip arp inspection interface ethernet 1/4
 Interface        Trust State    Rate (pps)    Burst Interval
 -------------    -----------    ----------    --------------
 Ethernet1/4      Trusted           15             5
switchB#
```

Step 4 Verify the list of DHCP snooping bindings.

```
switchB#show ip dhcp snooping binding
MacAddress          IpAddress        LeaseSec  Type           VLAN  Interface
------------------  ---------------  --------  -------------  ----  -------------
00:00:5E:00:53:01  10.0.0.2         4995      dhcp-snooping  1     Ethernet1/4
switchB#
```

Step 5 Check the statistics before and after DAI processes any packets.

```
switchB#show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 0
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchB#
```

If Host 2 sends out an ARP request with the IP address 10.0.0.2 and the MAC address 0001.0001.0001, the packet is forwarded, and the statistics are updated.

```
switchB#show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 1
ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 1
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchB#
```

If Host 2 attempts to send an ARP request with the IP address 10.0.0.1, DAI drops the request and logs the following system message:

```
00:18:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Ethernet1/4, vlan
1.([0001.0001.0001/10.0.0.1/0000.0000.0000/192.0.2.1/01:53:21 UTC Fri Jan 23 2015])
```

The statistics display as follows:

```
switchB#show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 1
ARP Res Forwarded  = 0
ARP Req Dropped    = 1
ARP Res Dropped    = 0
DHCP Drops         = 1
DHCP Permits       = 1
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
```

```
IP Fails-ARP Res   = 0
switchB#
```

Examples for DHCP relay with DAI

Use this reference to review examples for DHCP relay with DAI.

DHCP relay DAI status example

The following example displays whether the DHCP relay DAI feature is enabled. If the feature is not enabled, the value of the *DHCP Relay DAI enabled* entry in the database is *No*.

```
switch# show system internal dhcp database global config
Snooping enabled: Yes
Snoop option-82 enabled: No
Relay enabled: Yes
.
.
DHCP Relay DAI enabled : No
Validate source mac: No
Validate destination mac: No
```

Additional references for DAI

Use this reference to find additional references for DAI.

Related documents

Use this reference to review related documents.

Related documents

Table 106: Related documents

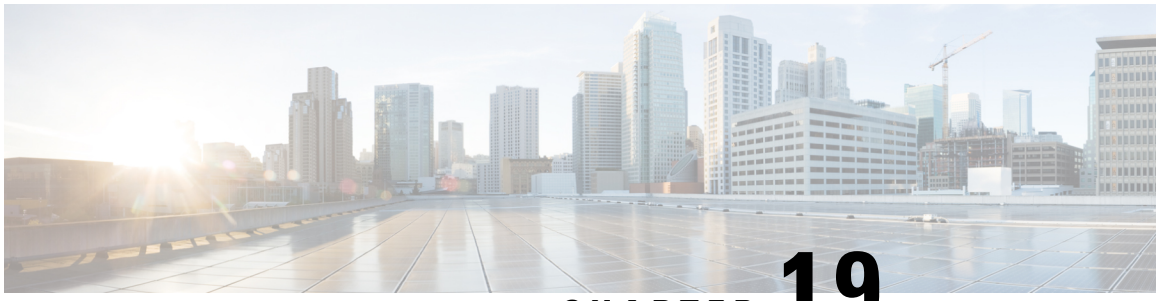
Related Topic	Document Title
ACL TCAM regions	Configure IP ACLs, on page 307
DHCP and DHCP snooping	Configuring DHCP, on page 441

Standards

Use this reference to review standards.

Table 107: Standards:

Standard	Title	
RFC-826	An Ethernet Address Resolution Protocol (https://example.com/doc/html/rfc826)	



CHAPTER 19

Configuring IP Source Guard

This chapter describes how to configure IP Source Guard on Cisco NX-OS devices.

This chapter includes the following sections:

- IP source guard overview, on page 539
- Prerequisites for IP source guard, on page 540
- Recommendation for IP source guard configuration, on page 540
- Default settings for IP source guard, on page 541
- Configuring IP source guard, on page 541
- Displaying IP source guard bindings, on page 544
- Clearing IP source guard statistics, on page 544
- IP source guard configuration example, on page 544
- Additional references, on page 545

IP source guard overview

IP Source Guard is a filter that operates on a per interface basis.

IP traffic is permitted only if the IP address and MAC address of each packet match an entry from either the DHCP snooping binding table or a static IP source entry.

- Entries in the Dynamic Host Configuration Protocol (DHCP) snooping binding table
- Static IP source entries that you configure

Filtering on trusted IP and MAC address bindings helps prevent spoofing attacks. In a spoofing attack, an attacker uses the IP address of a valid host to gain unauthorized network access. To circumvent IP Source Guard, an attacker would have to spoof both the IP address and the MAC address of a valid host.

IP source guard architecture

You can enable IP Source Guard on Layer 2 interfaces that are not trusted by DHCP snooping. It supports interfaces operating in access mode and trunk mode. When you first enable IP Source Guard, the interface blocks all inbound IP traffic except:

- DHCP packets that DHCP snooping inspects and then forwards or drops, depending on the results of the inspection
- IP traffic from static IP source entries you have configured on the NX-OS device.

The device permits IP traffic when DHCP snooping adds a binding table entry for the IP address and MAC address of a packet, or when a static IP source entry has been configured.

The device drops IP packets when neither the IP address nor the MAC address have a binding table entry or a static IP source entry.

For example, assume that the **show ip dhcp snooping binding** command displays this binding table entry:

```
MacAddress          IpAddress     LeaseSec    Type            VLAN    Interface
------------------  ----------    ---------   -------------   ----    ---------
00:00:5E:00:53:01   10.5.5.2      6943        dhcp-snooping   10      Ethernet2/3
```

If the device receives an IP packet with an IP address of 10.5.5.2, IP Source Guard forwards the packet only if the MAC address of the packet is 00:00:5E:00:53:01.

Prerequisites for IP source guard

IP Source Guard requires these prerequisites

- You must enable the DHCP feature and DHCP snooping before you can configure IP Source Guard. See Configuring DHCP, on page 441 .
- You must configure the ACL TCAM region size for IP Source Guard using the **hardware access-list tcam region ipsg** command. See Configure ACL TCAM region sizes, on page 344 .

Note By default, the ipsg region size is zero. Allocate enough entries to this region to store and enforce the SMAC–IP bindings.

Recommendation for IP source guard configuration

Use this reference to review recommendation for IP source guard configuration.

Recommendation for IP source guard configuration information

IP Source Guard has the following configuration guidelines and limitations:

- When you enable IP Source Guard on an interface, only sources with an IP-MAC address binding table entry or a static IP source entry can send IP traffic. You may notice a disruption in IP traffic until hosts on the interface receive a new IP address from a DHCP server.
- You must enable DHCP snooping to build and maintain the IP-MAC address binding table with IP Source Guard, or manually maintain static IP source entries.
- Do not use IP Source Guard on fabric extender (FEX) ports or generic expansion module (GEM) ports because it is not supported.
- Do not configure IP Source Guard on EoR, as it is not supported.
- The following guidelines and limitations apply to the Cisco Nexus 9200 Series switches:
  - IPv6 adjacency is not formed with IPSG enabled on the incoming interface.

- IPSG drops ARP packets at HSRP standby.
- With DHCP snooping and IPSG enabled, if a binding entry exists for the host, traffic is forwarded to the host even without ARP.

- Beginning with Cisco NX-OS Release 9.3(5), IP Source Guard is supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.
- You do not need to configure TCAM carving when you use IP Source Guard on Cisco Nexus 9300-X Cloud Scale Switches.
- If IPSG is enabled, port security can't be enabled on the interface.

Default settings for IP source guard

Use the table to see the default states and values for key IP Source Guard configuration parameters.

Table 108: Default IP source guard parameters

Parameters	Default
IP Source Guard	IP Source Guard is disabled on all interfaces.
IP source entries	There are no static or default IP source entries.

Configuring IP source guard

Use this reference to review configuring IP source guard.

Enable or disable IP source guard on a layer 2 interface

Configure IP Source Guard to control access based on valid IP address assignments.

By default, IP Source Guard is disabled. Enabling it prevents devices with spoofed IP addresses from communicating on the network.

Before you begin

- Ensure that Dynamic Host Configuration Protocol (DHCP) and DHCP snooping are enabled for your device.
- Confirm that the Access Control List (ACL) TCAM region size for IP Source Guard (IPSG) is configured.

Procedure

Step 1 Enter the global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter the interface configuration mode for the specified interface using **interface ethernet** *slot* / *port* command.

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

Step 3 Enable IP Source Guard on the interface using the [**no**] **ip verify source dhcp-snooping-vlan** command.

Example:

```
switch(config-if)#ip verify source dhcp-snooping-vlan
```

The **no** form of this command disables IP Source Guard on the interface.

Step 4 (Optional) Use the show running configuration command for DHCP snooping to display the IP Source Guard configuration using the **show running-config dhcp** command.

Example:

```
switch#show running-config dhcp
```

Step 5 (Optional) Use the copy running-config startup-config command to save the running configuration to the startup configuration using **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

IP Source Guard is enabled or disabled on the interface.

What to do next

Verify your configuration and monitor interface behavior to confirm that the interface operates as intended.

Add or remove a static IP source entry

Ensure your device uses accurate static IP-MAC bindings for DHCP snooping to improve device integrity and network security.

By default, no static IP source entries exist. Add entries to enhance security and prevent unauthorized DHCP assignments. Remove stale entries regularly to keep the bindings accurate.

Follow these steps to add or remove a static IP source entry:

Procedure

Step 1 Enter the global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Create or delete the static IP binding for the current interface using the [**no**] **ip source binding** *ip-address mac-address* **vlan** *vlan-id* **interface** *interface-type slot* / *port* command.

Example:

```
switch(config)#ip source binding 192.0.2.10 001f.28bd.0013 vlan 100 interface ethernet 2/3
```

Use the **no** form of this command to remove the static IP source entry.

Step 3 (Optional) Display current IP-MAC bindings, including static IP source entries, using the **show ip dhcp snooping binding** [**interface** *interface-type slot* / *port*] command.

Example:

```
switch#show ip dhcp snooping binding interface ethernet 2/3
```

All static entries are displayed with the term "static" in the Type column.

Step 4 (Optional) Save the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

After you make changes, the device updates its static IP source binding entries and retains them after a reboot.

What to do next

- Verify correct operation and security posture by reviewing the binding table.
- Remove obsolete entries as network topology changes.

Configure IP source guard on trunk ports

When IP Source Guard is configured on a port, the port drops traffic unless the TCAM has a DHCP snooping entry to allow it. If you enable IP Source Guard on trunk ports and want certain VLANs to bypass this check, specify the VLANs you want to exclude, even if those VLANs do not have DHCP snooping enabled.

Before you begin

Make sure that the DHCP feature and DHCP snooping are enabled.

Procedure

Step 1 Enter the global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify the list of VLANs to exclude from the DHCP snooping check for IP Source Guard on trunk ports using the [**no**] **ip dhcp snooping ipsg-excluded vlan** *vlan-list* command.

Example:

```
switch(config)#ip dhcp snooping ipsg-excluded vlan 1001-1256,3097
```

Step 3 (Optional) Display the excluded VLANs using the **show ip ver source** [**ethernet** *slot/port* | **port-channel** *channel-number*] command.

Example:

```
switch#show ip ver source
```

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

The specified VLANs are excluded from IP Source Guard checks on trunk ports.

What to do next

Verify that the configuration is correct and monitor the trunk ports for expected traffic flow.

Displaying IP source guard bindings

Use the **show ip ver source** [**ethernet** *slot/port* | **port-channel** *channel-number*] command to display the IP-MAC address bindings.

Clearing IP source guard statistics

To clear IP Source Guard statistics, use the commands in this table.

Table 109: Clear IP source guard statistics commands

Command	Purpose
clear access-list ipsg stats [**instance** *number* | **module** *number*]	Clears IP Source Guard statistics.

IP source guard configuration example

This example demonstrates how to create a static IP source entry and enable IP Source Guard on an interface.

```
ip source binding 192.0.2.17 001f.28bd.0013 vlan 100 interface ethernet 2/3
 interface ethernet 2/3
no shutdown
ip verify source dhcp-snooping-vlan
show ip ver source

IP source guard excluded vlans:
-------------------------------------------------------
None
```

```
-----------------------------------
IP source guard is enabled on the following interfaces:
-------------------------------------------------------
ethernet2/3
```

Additional references

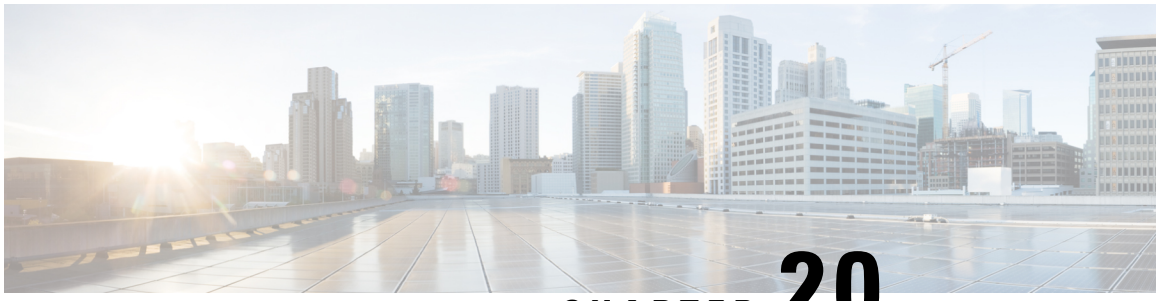
Use this reference to find additional references.

Reference documents for ACL TCAM regions and DHCP snooping

Use this reference to review reference documents for ACL TCAM regions and DHCP snooping.

Table 110: Reference documents for ACL TCAM regions and DHCP snooping

Related Topic	Links to Documentation
ACL TCAM regions	
DHCP and DHCP snooping	Configuring DHCP, on page 441



CHAPTER 20

Configuring Password Encryption

This chapter describes how to configure password encryption on Cisco NX-OS devices.

This chapter includes the following sections:

- AES password encryption and primary encryption keys, on page 547
- Guidelines and limitations for password encryption, on page 548
- Password encryption default settings, on page 550
- Configuring password encryption, on page 550
- Password encryption configuration verification, on page 554
- Configuration examples for password encryption, on page 554

AES password encryption and primary encryption keys

AES password encryption also known as Type-6 encryption is a password security feature that

- uses 128-bit Advanced Encryption Standard to encrypt passwords,
- allows for reversible encryption via a primary encryption key, and
- supports storing passwords for applications (such as RADIUS and TACACS+) in encrypted format.

You must enable the AES password encryption feature and configure a primary encryption key, which is used to encrypt and decrypt passwords.

After you enable AES password encryption and configure a primary key, all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) are stored in Type-6 encrypted format, unless you disable Type-6 password encryption. You can also configure Cisco NX-OS to convert all existing weakly encrypted passwords to Type-6 encrypted passwords.

Related Topics

Configure a primary key and enable the AES password encryption feature, on page 550
Configure global RADIUS keys, on page 67
Configure a key for a specific RADIUS server, on page 68
Configure global TACACS+ keys, on page 99
Configure a key for a specific TACACS+ server, on page 100
Configure a primary key and enable the AES password encryption feature, on page 550

Guidelines and limitations for password encryption

Use this reference to review the guidelines and limitations for password encryption.

Password encryption has the following configuration guidelines and limitations:

Configuration and application support guidelines

- Only users with administrator privilege (network-admin) can configure the AES password encryption feature, associated encryption and decryption commands, and primary keys.
- Beginning with Cisco NX-OS Release 10.3(3)F, RPM keychain infra supports AES password encryption for RPM legacy keychains on Cisco Nexus 9000 Series platform switches.
- Configurations containing Type-6 encrypted passwords are not rollback-compliant.
- You can enable the AES password encryption feature without a primary key, but encryption starts only when a primary key is present in the system.
- For TACACS+ and RPM legacy keychain, after you enable the AES password encryption feature and configure a primary key, you must run the **encryption re-encrypt obfuscated** command to convert the passwords to Type-6 encrypted passwords.
- Deleting the primary key stops Type-6 encryption and causes all existing Type-6 encrypted passwords to become unusable, unless the same primary key is reconfigured.
- To move the device configuration to another device, either decrypt the configuration before porting it to the other device or configure the same primary key on the device to which the configuration will be applied.

Type-6 keychain limitations

- Type-6 encryption is supported only for MACsec and RPM legacy keychain. It is not supported for cloudsec keys.
- Starting from Cisco NX-OS Release 9.3(6), converting Type-6 encrypted passwords back to original state is not supported on MACsec keychain.
- Starting from Cisco NX-OS Release 10.3(3)F, converting Type-6 encrypted passwords back to original state is not supported for RPM legacy keychain.
- Type-6 encryption can be configured only when the AES password encryption feature is enabled and the primary key is configured.
- When the primary key is configured and the AES password encryption feature is enabled on a switch, each MACsec key string configuration under the keychain infra is automatically encrypted with Type-6 encryption.
- Primary key configuration is local to the switch. If you take the Type-6 configured running data from one switch and apply it on another switch where a different primary key is configured, then decryption on the new switch fails.
- If you erase the startup configuration and use the configuration replace feature after a Type-6 encryption, the configuration replace fails because the primary key is not stored in PSS. Therefore, there is configuration loss for MACsec Type-6 encrypted key string.

- When you configure the Type-6 keys, you cannot modify the existing Type-6 encrypted key strings to Type-7 encrypted key string without applying the decrypt command provided by SKSD.
- For RPM legacy keychains, Type-6 key-strings can be configured without AES password encryption feature enabled and primary key configured, but these Type-6 key-strings are unusable until AES password encryption feature is enabled and the primary key with which the Type-6 key-strings were generated is configured.

Upgrade, downgrade, and primary key limitations

- If you downgrade the system by cold reboot with an old image where the Type-6 encryption is not supported, you must take out the configuration before you proceed with the cold reboot. Failing to do so leads to loss in configuration.
- After you downgrade the system, the Type-6 configuration is lost.
- If you downgrade the system by ISSD, capability conf check is invoked and it notifies you to remove the configuration before proceeding with the downgrade. You can use the **encryption decrypt** command to convert the Type-6 encrypted keys to Type-7 encryption keys, and then proceed with the downgrade.
- During an ISSU upgrade, if you migrate from an older image which includes the Type-7 encrypted keys to a new image that supports Type-6 encryption, the rpm does not convert the existing keys to Type-6 encrypted keys until re-encryption is enforced. To enforce a re-encryption, use the **encryption re-encrypt obfuscated** command.
- After ISSU upgrade from an older image which includes the Type-7 encrypted keys to a new image that supports Type-6 encryption, if configuration replace is done using the configuration file saved in older image or configuration file saved after upgrade without re-encrypting the password to Type-6 by using the **encryption re-encrypt obfuscated** command, the configuration replace will fail.
- If you change the primary key after a Type-6 encryption, the decrypt command fails on the existing Type-6 encrypted key-string. You must delete the existing Type-6 key string and configure a new key string.
- Starting from Cisco NX-OS Release 10.3(2)F, you can configure primary key using DME payload and non-interactive mode.
- During upgrade, while performing device reload, if ASCII replay is triggered without binary restore, primary key gets lost. The primary key must be reconfigured after device reload. Use the **key config-key ascii** command to reconfigure the primary key and avoid encryption issues. However, upgrade with binary restore retains the primary key after the reboot.
- During downgrade, where both source and target images support Type-6 encryption, while performing device reload, if ASCII replay is triggered without binary restore, primary key gets lost. The primary key must be reconfigured after device reload. Use the **key config-key ascii** command to reconfigure the primary key and avoid encryption issues. However, downgrade with binary restore retains the primary key after the reboot, provided both source and target images support Type-6 encryption.

  If you downgrade the system from an image that supports Type-6 encryption to an image that does not support Type-6 encryption, compatibility check fails.

Password encryption default settings

This table lists the default settings for password encryption parameters.

Table 111: Password encryption parameters default settings

Parameters	Default
Advanced Encryption Standard (AES) password encryption feature	Disabled
Primary key	Not configured

Configuring password encryption

This section describes the tasks for configuring password encryption on Cisco NX-OS devices.

Configure a primary key and enable the AES password encryption feature

Secure device authentication by using AES password encryption (Type-6).

You can configure a primary key for Type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Beginning with Cisco NX-OS Release 10.3(3)F, Type-6 encryption is supported for RPM legacy keychain.

Follow these steps to configure a primary key and enable the AES password encryption feature:

Procedure

Step 1 Configures a primary key (`Master Key`) to be used with the AES password encryption feature using the [**no**] **key config-key ascii** [<new_key> old <old_master_key>] command.

Example:

```
switch#key config-key ascii
New Master Key:
Retype Master Key:

{"actionLSubj": {"attributes": {"dn": "sys/action/lsubj-[sys/passwdenc]"}
"children": [{"smartcardPasswdEncryptMasterKeyConfigLTask": {"attributes": {"adminSt":
"start",
"dn": "sys/action/lsubj-[sys/passwdenc]/smartcardPasswdEncryptMasterKeyConfigLTask",
"key": "ciscociscociscocisco",
"oldKey": "test1test1test1test1",
"delete": "no",
"freq": "one-shot"}}}]}}
```

The primary key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the primary key at any time.

If you enable the AES password encryption feature before configuring a primary key, a message appears stating that password encryption will not take place unless a primary key is configured. If a primary key is already configured, you are prompted to enter the current primary key before entering a new primary key.

Note
Starting with Cisco NX-OS Release 10.3(2)F, you can configure primary key using DME payload and non-interactive mode.

Step 2 Enter the global configuration mode. **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Step 3 Enable or disable the AES password encryption feature using the [**no**] **feature password encryption aes** command.

Example:

```
switch(config)#feature password encryption aes
```

.

Step 4 Convert existing plain or weakly encrypted passwords to Type-6 encrypted passwords using the **encryption re-encrypt obfuscated** command.

Example:

```
switch(config)#encryption re-encrypt obfuscated
```

Step 5 (Optional) Display the configuration status of the AES password encryption feature and the primary key using the **show encryption service stat** command.

Example:

```
switch#show encryption service stat
```

Step 6 Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Note
This command is necessary to synchronize the primary key in the running configuration and the startup configuration.

Related Topics

AES password encryption and primary encryption keys, on page 547
AES password encryption and primary encryption keys, on page 547
Configure text for a key, on page 562
Configure accept and send lifetimes for a key, on page 564

Convert existing passwords to Type-6 encrypted passwords

You can convert existing plain or weakly encrypted passwords to Type-6 encrypted passwords.

Before you begin

Ensure that you have enabled the AES password encryption feature and configured a primary key.

Procedure

Convert existing plain or weakly encrypted passwords to Type-6 encrypted passwords using the **encryption re-encrypt obfuscated** command.

Example:

```
switch#encryption re-encrypt obfuscated
```

Recover original passwords from Type-6 encrypted values

You can convert encrypted passwords to cleartext to carry out migration or troubleshooting.

You can convert Type-6 encrypted passwords back to their original states. This functionality is not supported for macsec keychain.

Before you begin

Ensure that you have configured a primary key.

Procedure

Use the **encryption decrypt type6** command to recover original passwords..

Example:

```
switch#encryption decrypt type6
Please enter current Master Key:
```

The console displays the original (cleartext) passwords after the system decrypts them.

What to do next

Verify that all required passwords have been decrypted. If you do not need cleartext outputs in your logs, remove them.

Enable Type-6 encryption on MACsec keys

The type-6 encryption feature, also known as the Advanced Encryption Standard (AES) password encryption feature allows you to securely store MACsec keys in a Type-6 encrypted format.

Beginning with Cisco NX-OS Release 9.3(5), you can store MACsec keys in a Type-6 encrypted format on all Cisco Nexus 9000 Series switches which support the MACsec feature.

Procedure

Step 1 [**no**] **key config-key ascii**

Example:

```
switch#key config-key ascii
New Master Key:
Retype Master Key:
```

Configures the primary key (`Master Key`).

Step 2 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 3 [**no**] **feature password encryption aes tam**

Example:

```
switch(config)#feature password encryption aes tam
```

Enables or disables the AES password encryption feature.

Step 4 **key chain** *name* **macsec**

Example:

```
switch(config)#key chain 1 macsec
switch(config-macseckeychain)#
```

Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.

Step 5 **key** *key-id*

Example:

```
switch(config-macseckeychain)#key 1000
switch(config-macseckeychain-macseckey)#
```

Creates a MACsec key and enters MACsec key configuration mode. The range is 1–32 octets, and the maximum size is 32 or 64 bits. AES_128 is used for 32 bit, while AES_256 is used for 64 bits.

Step 6 **key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** | **AES_256_CMAC**}

Example:

```
switch(config-macseckeychain-macseckey)#key-octet-string
abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789
cryptographic-algorithm AES_256_CMAC
```

Configures the octet string for the key. The *octet-string* argument can contain up to 64 hexadecimal characters. The octet key is encoded internally, so the key in clear text does not appear in the output of the **show running-config macsec** command.

The key octet string includes the following:

- 0 Encryption Type - No encryption (default)

- 6 Encryption Type - Proprietary (Type-6 encrypted)
- 7 Encryption Type - Proprietary WORD key octet string with maximum 64 characters

Delete Type-6 encrypted passwords

You can delete all Type-6 encrypted passwords from the Cisco NX-OS device.

Procedure

Delete all Type-6 encrypted passwords using the **encryption delete type6** command.

Example:

```
switch#encryption delete type6
```

Password encryption configuration verification

Use these commands to display password encryption configuration information.

Table 112: Password encryption configuration verification commands

Command	Purpose
show encryption service status	Displays the configuration status of the AES password encryption feature and the primary key.

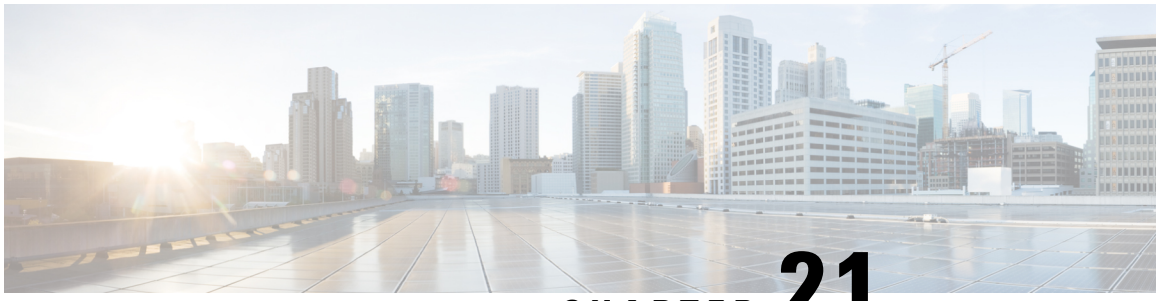
Configuration examples for password encryption

Examples for password encryption

This example demonstrates how to create a primary key, enable the AES password encryption feature, and configure a Type-6 encrypted password for an application that uses TACACS+.

```
key config-key ascii
New Master Key:
Retype Master Key:
configure terminal
feature password encryption aes tam
show encryption service status
Encryption service is enabled.
Master Encryption Key is configured.
Type-6 encryption is being used.
feature tacacs+
tacacs-server key Cisco123
show running-config tacacs+
feature tacacs+
```

```
logging level tacacs 5
tacacs-server key 6
"JDYkqyIFWeBvzpljSfWmRZrmRSRE8syxKlOSjP9RCCkFinZbJI3GD5c6rckJR/Qju2PKLmOewbheAA=="
```



CHAPTER 21

Configuring Keychain Management

This chapter describes how to configure keychain management on a Cisco NX-OS device.

This chapter includes the following sections:

- About keychain management, on page 557
- Prerequisites for keychain management, on page 558
- Guidelines and limitations for keychain management, on page 558
- Default settings for keychain management, on page 559
- Configuring keychain management, on page 559
- Determining active key lifetimes, on page 567
- Keychain management configuration verification commands, on page 567
- Configuration example for keychain management, on page 567
- Where to go next, on page 568
- Additional references for keychain management, on page 568

About keychain management

Keychain management allows you to create and maintain keychains, which are sequences of keys (sometimes called shared secrets). You can use keychains with features that secure communications with other devices by using key-based authentication. The device allows you to configure multiple keychains.

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication. For more information, see the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

Lifetime of a Key

To maintain stable communications, each device that uses a protocol that is secured by key-based authentication must be able to store and use more than one key for a feature at the same time. Based on the send and accept lifetimes of a key, keychain management provides a secure mechanism to handle key rollover. The device uses the lifetimes of keys to determine which keys in a keychain are active.

Each key in a keychain has two lifetimes, as follows:

- Accept lifetime

  The time interval within which the device accepts the key during a key exchange with another device.

- Send lifetime

The time interval within which the device sends the key during a key exchange with another device.

You define the send and accept lifetimes of a key using the following parameters:

- Start-time

  The absolute time that the lifetime begins.

- End-time

  The end time can be defined in one of the following ways:

  - The absolute time that the lifetime ends
  - The number of seconds after the start time that the lifetime ends
  - Infinite lifetime (no end-time)

During a key send lifetime, the device sends routing update packets with the key. The device does not accept communication from other devices when the key sent is not within the accept lifetime of the key on the device.

We recommend that you configure key lifetimes that overlap within every keychain. This practice avoids failure of neighbor authentication due to the absence of active keys.

Prerequisites for keychain management

Keychain management has no prerequisites.

Guidelines and limitations for keychain management

Use this reference to review the guidelines and limitations for keychain management.

Keychain management has the following configuration guidelines and limitations:

- Changing the system clock impacts when the keys are active.
- A keychain's configuration type must match the type it has been linked to within the client protocol. If an attempt is made to mismatch these types, a syslog message is generated to notify the user.

  For example: It is not supported if a keychain named **keychain_abc** is configured as a Macsec keychain but is associated as a Classic keychain with OSPF. Similarly, the case where the keychain is first associated with the client (a process known as forward-referencing) and then configured as a different keychain type, is also not supported.

- It is highly recommended for user to specify the passwordtype and password when programmatically(restconf/Netconf and so on) configuring a neighbor/template’s password. When either one of the property is missing in the programmatic call, BGP will use already available(or default) value of the missing property to configure the neighbor/template’s password.

  If the user has to configure with a property missing then the user has to follow the same sequence of steps in both peer routers.

Default settings for keychain management

Use this reference to review the default settings for keychain management.

Table 113: Default keychain management parameters

Parameters	Default
Key chains	No keychain exists by default.
Keys	No keys are created by default when you create a new keychain.
Accept lifetime	Always valid.
Send lifetime	Always valid.
Key-string entry encryption	Unencrypted.

Configuring keychain management

Explains configuring keychain management.

Create a keychain

You can create a keychain on the device. A new keychain contains no keys.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **key chain** *name*

Example:

```
switch(config)#key chain bgp-keys
switch(config-keychain)#
```

Creates the keychain and enters keychain configuration mode.

Step 3 (Optional) **show key chain** *name*

Example:

```
switch#show key chain bgp-keys
```

Displays the keychain configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Remove a keychain

You can remove a keychain on the device.



Note Removing a keychain removes any keys within the keychain.

Before you begin

If you are removing a keychain, ensure that no feature uses it. If a feature is configured to use a keychain that you remove, that feature is likely to fail to communicate with other devices.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **no key chain** *name*

Example:

```
switch(config)#no key chain bgp-keys
```

Removes the keychain and any keys that the keychain contains.

Step 3 (Optional) **show key chain** *name*

Example:

```
switch#show key chain bgp-keys
```

Confirms that the keychain no longer exists in running configuration.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure a primary key and enable the AES password encryption feature

Secure device authentication by using AES password encryption (Type-6).

You can configure a primary key for Type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Beginning with Cisco NX-OS Release 10.3(3)F, Type-6 encryption is supported for RPM legacy keychain.

Follow these steps to configure a primary key and enable the AES password encryption feature:

Procedure

Step 1 Configures a primary key (`Master Key`) to be used with the AES password encryption feature using the [**no**] **key config-key ascii** [<new_key> old <old_master_key>] command.

Example:

```
switch#key config-key ascii
New Master Key:
Retype Master Key:

{"actionLSubj": {"attributes": {"dn": "sys/action/lsubj-[sys/passwdenc]"}
"children": [{"smartcardPasswdEncryptMasterKeyConfigLTask": {"attributes": {"adminSt":
"start",
"dn": "sys/action/lsubj-[sys/passwdenc]/smartcardPasswdEncryptMasterKeyConfigLTask",
"key": "ciscociscociscocisco",
"oldKey": "test1test1test1test1",
"delete": "no",
"freq": "one-shot"}}}]}}
```

The primary key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the primary key at any time.

If you enable the AES password encryption feature before configuring a primary key, a message appears stating that password encryption will not take place unless a primary key is configured. If a primary key is already configured, you are prompted to enter the current primary key before entering a new primary key.

Note
Starting with Cisco NX-OS Release 10.3(2)F, you can configure primary key using DME payload and non-interactive mode.

Step 2 Enter the global configuration mode. **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Step 3 Enable or disable the AES password encryption feature using the [**no**] **feature password encryption aes** command.

Example:

```
switch(config)#feature password encryption aes
```

.

Step 4 Convert existing plain or weakly encrypted passwords to Type-6 encrypted passwords using the **encryption re-encrypt obfuscated** command.

Example:

```
switch(config)#encryption re-encrypt obfuscated
```

Step 5 (Optional) Display the configuration status of the AES password encryption feature and the primary key using the **show encryption service stat** command.

Example:

```
switch#show encryption service stat
```

Step 6 Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Note
This command is necessary to synchronize the primary key in the running configuration and the startup configuration.

Related Topics

AES password encryption and primary encryption keys, on page 547
AES password encryption and primary encryption keys, on page 547
Configure text for a key, on page 562
Configure accept and send lifetimes for a key, on page 564

Configure text for a key

You can configure the text for a key. The text is the shared secret. The device stores the text in a secure format.

For MACsec and RPM legacy keychain, the text is encrypted and stored in Type-6 format if AES password encryption feature is enabled and primary key configured otherwise it will be stored in Type-7 encrypted format.

By default, accept and send lifetimes for a key are infinite, which means that the key is always valid. After you configure the text for a key, configure the accept and send lifetimes for the key.

Before you begin

Determine the text for the key. You can enter the text as unencrypted text or in the encrypted form that Cisco NX-OS uses to display key text when you use the **show key chain** command. Using the encrypted form is particularly helpful if you are creating key text to match a key as shown in the **show key chain** command output from another device.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **key chain** *name*

Example:

```
switch(config)#key chain bgp-keys
switch(config-keychain)#
```

Enters keychain configuration mode for the keychain that you specified.

Step 3 **key** *key-ID*

Example:

```
switch(config-keychain)#key 13
switch(config-keychain-key)#
```

Enters key configuration mode for the key that you specified. The *key-ID* argument must be a whole number between 0 and 65535.

Step 4 **key-string** [*encryption-type*] *text-string*

Example:

```
switch(config-keychain-key)#key-string 0 AS3cureStr1ng
```

Configures the text string for the key. The *text-string* argument is alphanumeric, case-sensitive, and supports special characters.

The *encryption-type* argument can be one of the following values:

- 0—The *text-string* argument that you enter is unencrypted text. This is the default.
- 6—Beginning with Cisco NX-OS Release 10.3(3)F, the Cisco proprietary (Type-6 encrypted) method is supported on Cisco Nexus 9000 Series platform switches.
- 7—The *text-string* argument that you enter is encrypted. The encryption method is a Cisco proprietary method. This option is useful when you are entering a text string based on the encrypted output of a **show key chain** command that you ran on another Cisco NX-OS device.

The **key-string** command has limitations on using the following special characters in the *text-string*:

Table 114: Configure text for a key

Special Character	Description	Comments
|	Vertical bar or pipe	Unsupported
>	Greater than	Unsupported
\	Backslash	Unsupported
(	Left parenthesis	Unsupported
'	Apostrophe	Unsupported
"	Quotation mark	Unsupported
?	Question mark	Supported. E a question m

For more information on the special characters usage in commands, see Understanding the Command-Line Interface section.

Step 5 (Optional) **show key chain** *name* [**mode decrypt**]

Example:

```
switch#show key chain bgp-keys
```

Shows the keychain configuration, including the key text configuration. The mode decrypt option, which can be used by a device administrator only, displays the keys in cleartext.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Configure a primary key and enable the AES password encryption feature, on page 550

Configure accept and send lifetimes for a key

You can configure the accept lifetime and send lifetime for a key. By default, accept and send lifetimes for a key are infinite, which means that the key is always valid.



Note We recommend that you configure the keys in a keychain to have overlapping lifetimes. This practice prevents loss of key-secured communication due to moments where no key is active.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **key chain** *name*

Example:

```
switch(config)#key chain bgp-keys
switch(config-keychain)#
```

Enters keychain configuration mode for the keychain that you specified.

Step 3 **key** *key-ID*

Example:

```
switch(config-keychain)#key 13
switch(config-keychain-key)#
```

Enters key configuration mode for the key that you specified.

Step 4 **accept-lifetime** [**local**] *start-time* [**duration** *duration-value* | **infinite** | *end-time*]

Example:

```
switch(config-keychain-key)#accept-lifetime 00:00:00 Jun 13 2013 23:59:59 Sep 12 2013
```

Configures an accept lifetime for the key. By default, the device treats the *start-time* and *end-time* arguments as UTC. If you specify the **local** keyword, the device treats these times as local times.

The *start-time* argument is the time of day and date that the key becomes active.

Specify the end of the lifetime with one of the following options:

- **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
- **infinite**—The accept lifetime of the key never expires.
- *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive.

Step 5 **send-lifetime** [**local**] *start-time* [**duration** *duration-value* | **infinite** | *end-time*]

Example:

```
switch(config-keychain-key)#send-lifetime 00:00:00 Jun 13 2013 23:59:59 Aug 12 2013
```

Configures a send lifetime for the key. By default, the device treats the *start-time* and *end-time* arguments as UTC. If you specify the **local** keyword, the device treats these times as local times.

The *start-time* argument is the time of day and date that the key becomes active.

You can specify the end of the send lifetime with one of the following options:

- **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
- **infinite**—The send lifetime of the key never expires.
- *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive.

Step 6 (Optional) **show key chain** *name* [**mode decrypt**]

Example:

```
switch#show key chain bgp-keys
```

Shows the keychain configuration, including the key text configuration. The mode decrypt option, which can be used by a device administrator only, displays the keys in cleartext.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Related Topics

Configure a primary key and enable the AES password encryption feature, on page 550

Configure a key for OSPFv2 cryptographic authentication

You can configure message digest 5 (MD5) or hash-based message authentication code secure hash algorithm (HMAC-SHA) authentication for OSPFv2.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **key chain** *name*

Example:

```
switch(config)#key chain bgp-keys
switch(config-keychain)#
```

Enters keychain configuration mode for the keychain that you specified.

Step 3 **key** *key-ID*

Example:

```
switch(config-keychain)#key 13
switch(config-keychain-key)#
```

Enters key configuration mode for the key that you specified. The *key-ID* argument must be a whole number between 0 and 65535.

Note
For OSPFv2, the key identifier in the key key-id command supports values from 0 to 255 only.

Step 4 [**no**] **cryptographic-algorithm** {**HMAC-SHA-1** | **HMAC-SHA-256** | **HMAC-SHA-384** | **HMAC-SHA-512** | **MD5**}

Example:

```
switch(config-keychain-key)#cryptographic-algorithm HMAC-SHA-1
```

Configures the OSPFv2 cryptographic algorithm to be used for the specified key. You can configure only one cryptographic algorithm per key.

Step 5 (Optional) **show key chain** *name*

Example:

```
switch#show key chain bgp-keys
```

Shows the keychain configuration.

Step 6 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Determining active key lifetimes

Use this reference to determine which keys within a key chain have active accept or send lifetimes.

Table 115: Active key lifetime command

Command	Purpose
show key chain	Displays the key chains configured on the device.

Keychain management configuration verification commands

Use this reference to verify the Keychain management configuration.

To verify keychain management configuration information, use the following command:

Table 116: Keychain management show commands

Command	Purpose
show key chain *name*	Displays the keychains configured on the device.

Configuration example for keychain management

This example shows how to configure a keychain named "ospf-keys". Each key text string is encrypted. The keys are configured to use MD5 as their cryptographic algorithm. Each key has longer accept lifetimes than send lifetimes, resulting in overlap between a pair of keys. In this example, there is configured overlap between key 1 and key 2, as well as key 2 and key 3. This prevents a period of time in which there are no active keys, helping to avoid a disruption in communication of the underlying protocol:

```
key chain ospf-keys
  key 1
    key-string 7 070c285f4d0658544541
    accept-lifetime local 00:00:00 May 13 2024  12:00:00 Sep 14 2024
    send-lifetime local 00:00:00 May 13 2024  00:00:00 Sep 14 2024
    cryptographic-algorithm MD5
  key 2
    key-string 7 070c285f4d0658574446
    accept-lifetime local 00:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    send-lifetime local 10:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    cryptographic-algorithm MD5
  key 3
    key-string 7 070c285fad0622474941
    accept-lifetime local 00:00:00 Jan 15 2025  12:00:00 Jun 15 2025
```

```
    send-lifetime local 10:00:00 Jan 15 2025  12:00:00 Jun 15 2025
    cryptographic-algorithm MD5
```

This example shows how to configure a keychain named "bgp-keys" with Type-6 encryption. This encryption mode is available when feature password encryption aes is enabled:

Configuration example for keychain management command examples

```
key chain bgp-keys
  key 1
    key-string 6
JDYkbN6ZTz3Hqrv5ZWliyxqlYiQXYc0wWpOnK7epMGoHK6qVJPeJtSYAGhQ9V+QKG4ZrcWeuunTtAA==
    accept-lifetime local 00:00:00 May 13 2024  12:00:00 Sep 14 2024
    send-lifetime local 00:00:00 May 13 2024  00:00:00 Sep 14 2024
  key 2
    key-string 6
JDYkO6Di45Bu1ikPja/r8VJNoSTa4I4QMxtzzG3DQzal9G9LJA6F1WNGX8GRgn95SPuf4naoTZCtAA==
    accept-lifetime local 00:00:00 Sep 13 2024  12:00:00 Jan 15 2025
    send-lifetime local 10:00:00 Sep 13 2024  12:00:00 Jan 15 2025
  key 3
    key-string 6
JDYk8DJ15ZdOQ/O7vnj2M92lRiR2x8VrL0Muj/30TNlIK5f+JMFEHoWy0Rfuy827G/H10w2it7eVAA==
    accept-lifetime local 00:00:00 Jan 15 2025  12:00:00 Jun 15 2025
    send-lifetime local 10:00:00 Jan 15 2025  12:00:00 Jun 15 2025
```

Where to go next

For information about routing features that use keychains, see the

Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

Additional references for keychain management

Use this reference to find additional references for keychain management.

Related Documents

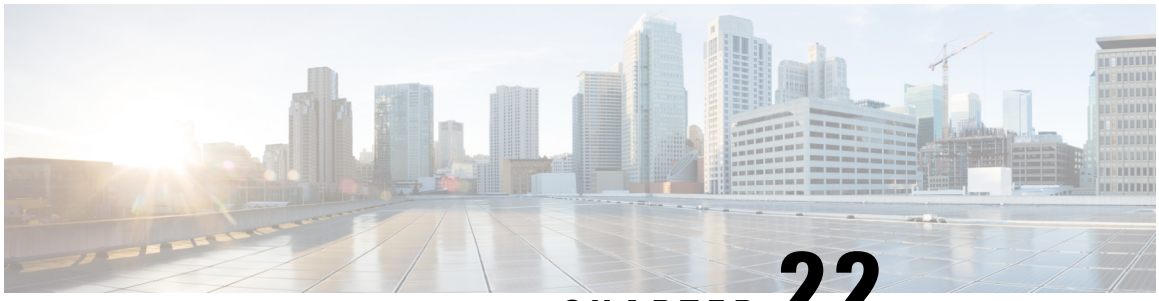
Table 117: Related documents

Related Topic	Document Title
Border Gateway Protocol	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*
OSPFv2	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards

Table 118: Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—



CHAPTER 22

Configuring Traffic Storm Control

This chapter describes how to configure traffic storm control on the Cisco NX-OS device.

This chapter includes the following sections:

- About traffic storm control, on page 569
- Licensing requirements for traffic storm control, on page 571
- Guidelines and limitations for traffic storm control, on page 571
- Default settings for traffic storm control, on page 575
- Configure traffic storm control for one-level threshold, on page 575
- Prioritizing storm-control policer over the CoPP policer, on page 577
- Configure traffic storm control for two-level threshold, on page 577
- Traffic storm control configuration verification commands, on page 579
- Monitoring traffic storm control counters, on page 579
- Configuration examples for traffic storm control, on page 580
- System log examples for traffic storm control, on page 581
- Additional references for traffic storm control, on page 581

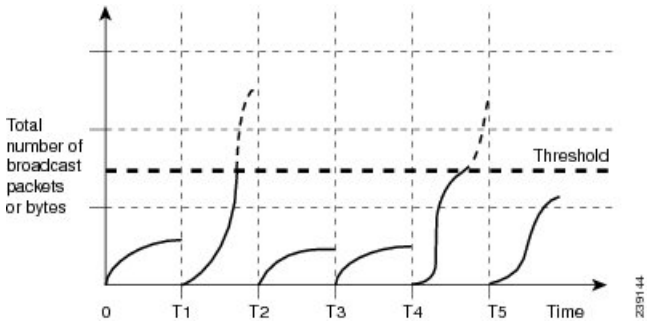
About traffic storm control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use the traffic storm control feature to prevent disruptions on Layer 2 or Layer 3 ports by a broadcast, multicast, or unicast traffic storm on physical interfaces.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, and unicast traffic over a 3.9-millisecond interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

This table shows the broadcast traffic patterns on a Layer 2 or Layer 3 interface over a given interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 20: Broadcast Suppression



The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of granularity. A higher threshold allows more packets to pass through.

Traffic storm control on the Cisco Nexus 9000v device is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from a Layer 2 or Layer 3 interface to the switching bus. Using the Individual/Group bit in the packet destination address, the circuitry determines if the packet is unicast or broadcast, tracks the current count of packets within the 3.9-millisecond interval, and filters out subsequent packets when a threshold is reached.

Traffic storm control uses a bandwidth-based method to measure traffic. You set the percentage of total available bandwidth that the controlled traffic can use. Because packets do not arrive at uniform intervals, the 3.9-millisecond interval can affect the behavior of traffic storm control.

The following are examples of how traffic storm control operation is affected

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within the 3.9-millisecond interval, traffic storm control drops all broadcast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and the combined broadcast and multicast traffic exceeds the level within the 3.9-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within the 3.9-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within the 3.9-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.

When the traffic exceeds the configured level, you can configure traffic storm control to perform the following optional corrective actions :

- Shut down—When ingress traffic exceeds the traffic storm control level that is configured on a port, traffic storm control puts the port into the error-disabled state. To reenable this port, you can use either the **shutdown** and **no shutdown** options on the configured interface, or the error-disable detection and recovery feature. You are recommended to use the **errdisable recovery cause storm-control** command for error-disable detection and recovery along with the **errdisable recovery interval** command for defining the recovery interval. The interval can range between 30 and 65535 seconds.
- Trap—You can configure traffic storm control to generate an SNMP trap when ingress traffic exceeds the configured traffic storm control level. The SNMP trap action is enabled by default. However, storm

control traps are not rate-limited by default. You can control the number of traps generated per minute by using the **snmp-server enable traps storm-control trap-rate** command.

By default, Cisco NX-OS takes no corrective action when traffic exceeds the configured level.

Licensing requirements for traffic storm control

The following table shows the licensing requirements for this feature:

Table 119: Licensing requirements for traffic storm control

Product	License Requirement
Cisco NX-OS	Traffic storm control requires no license. Any feature not included in a license package is bundled with the nx-os image and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and limitations for traffic storm control

Use this reference to review the guidelines and limitations for traffic storm control.

Configuration and threshold guidelines

Traffic storm control has the following configuration guidelines and limitations:

- The storm control feature does not work if you enable storm control on an interface where sFlow is also enabled.
- Storm control PPS option was supported only on Cisco Nexus 9300-FX2 platform switches. Beginning with Cisco NX-OS Release 10.3(2)F, it is also supported on Cisco Nexus 9300-FX3, 9300-GX, and 9300-GX2 platform switches.
- For Cisco Nexus NFE2-enabled devices, you can use the storm control-cpu to control the number of ARP packets sent to the CPU.
- Storm control can be configured on physical, port-channel, and breakout interfaces.
- Specify the traffic storm control level as a percentage of the total interface bandwidth:
  - The pps range can be from 0 to 200000000.
  - The optional fraction of a level can be from 0 to 99.
  - 100 percent means no traffic storm control.
  - 0.0 percent suppresses all traffic.
- For Cisco Nexus 9500 Series switches with 9400 Series line cards, and Cisco Nexus 9300 Series switches, you can use the storm control CLI to specify bandwidth level either as a percentage of port capacity or packets-per-second.

Packets-per-second behavior and release support

- Beginning with Cisco Nexus Release 9.2(1), the error margin is greater than 1% when you configure the storm control packets-per-second value as follows:
  - Traffic period < 60 s
  - Storm control pps <1000
  - Storm control pps <5 is not supported.
  - For 5-1000 pps, 20 additional pps is required to hit storm control in <60 s.
  - For >1000 pps, 2.5-3% additional pps is required to hit storm control in <60 s.
- Beginning with Cisco Nexus Release 9.2(1), you can use the percentage of port capacity or packets-per-second for the Cisco Nexus 9336C-FX2, Cisco Nexus 93300YC-FX2, and Cisco Nexus 93240YC-FX2-Z switches.

Hardware and statistics limitations

- If you have configured an SVI for the VLAN on Cisco Nexus 9200, 9300-EX platform switches, or on the N9K-X9700-FX3 line cards, storm control broadcast does not work for ARP traffic (ARP request).
- Local link and hardware limitations prevent storm-control drops from being counted separately. Instead, storm-control drops are counted with other drops in the discards counter.
- Because of hardware limitations and the method by which packets of different sizes are counted, the traffic storm control level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.
- Due to a hardware limitation, the output for the **show interface counters storm-control** command does not show ARP suppression when storm control is configured and the interface is actually suppressing ARP broadcast traffic. This limitation can lead to the configured action not being triggered but the incoming ARP broadcast traffic being correctly storm suppressed.
- Due to a hardware limitation, storm control is not supported for 400G ports beyond 70% of the port bandwidth in Cisco Nexus GX series platform switches.
- Due to a hardware limitation, the packet drop counter cannot distinguish between packet drops caused by a traffic storm and packet drops caused by other discarded input frames. This limitation can lead to the configured action being triggered even in the absence of a traffic storm.
- Due to a hardware limitation, storm suppression packet statistics are not supported on uplink ports.
- Due to a hardware limitation, storm suppression packet statistics do not include broadcast traffic on VLANs with an active switched virtual interface (SVI).
- Due to a design limitation, storm suppression packet statistics do not work if the configured level is 0.0, which is meant to suppress all incoming storm packets.

Interface and platform support limitations

- Traffic storm control is supported on the Cisco Nexus 9300 Series switches and the Cisco Nexus 9500 Series switches with the 9700-EX/ FX line card.

- Traffic storm control is not supported on Cisco N9K-M4PC-CFP2.
- Traffic storm control is not supported on either Host Interface (HIF) or Network Interface (NIF) FEX interfaces.
- When an interface is shut down, either as a result of storm-control errdisable or due to an administrative shutdown, the Rx storm suppression drop count may appear slightly lower than the actual number of packets dropped in hardware. This discrepancy can occur if the last update from the hardware or from internal system processes, such as the tahusd daemon, is missed as the interface goes down. The suppressed packet count is fully updated once the interface returns to the up state.
- Traffic storm control is only for ingress traffic, specifically for unknown unicast, unknown multicast, and broadcast traffic.



Note On Cisco Nexus 9000 Series switches, traffic storm control applies to unknown unicast traffic and not known unicast traffic.

- When port-channel members are error disabled due to a configured action, all individual member ports should be flapped to recover from the error disabled state.
- In Cisco Nexus Release 9.2(1), the traffic storm control feature is not supported on Cisco Nexus 9500 platform switches with the N9K-X96136YC-R line card and N9K-C9504-FM-R fabric module.
- Beginning with Cisco Nexus Release 9.2(4), the traffic storm control feature is supported on Cisco Nexus 9500 platform switches with the N9K-X96136YC-R line card and N9K-C9504-FM-R fabric module. Traffic storm control counters do not increment when the interface is flooded with the broadcast traffic.
- Beginning with Cisco Nexus Release 9.3(2), the traffic storm control feature with only rate-limiting is supported on Cisco Nexus 9500 platform switches with the N9K-X96136YC-R, N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards, and N9K-C9504-FM-R and N9K-C9508-FM-R fabric modules. Traffic storm control counters and storm-control action are not supported.

Cisco Nexus 9200 series switch limitations

- Traffic storm control with unknown multicast traffic is not supported.
- Packet-based statistics are not supported for traffic storm control as the policer supports only byte-based statistics.
- Traffic storm control is not supported for copy-to-CPU packets.

Two-level threshold and logging limitations

- Beginning with Cisco NX-OS Release 10.1(2), the storm control feature is supported on the N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches.
- Beginning with Cisco Nexus Release 10.1(2), for Cisco Nexus N9300-FX and N9300-FX2 series switches, you can configure a two-level threshold and logging support for Broadcast, Unknown Unicast, and Multicast (BUM) traffic, and also set trap or shutdown action for each threshold level. The existing storm control configuration is now used only for one-level threshold.

- The following guidelines and limitations apply to the two-level threshold and logging support for BUM traffic feature for Cisco Nexus 10.1(2) release:
  - The new traffic storm control feature in Cisco Nexus Release 10.1(2) supports a maximum of 62 ports (as a single slice) on Cisco Nexus N9300-FX and a total of 124 ports (as two slices) on Cisco Nexus N9300-FX2.
  - Traffic storm control supports devices that are only in one storm control mode at a time, either one-level or two-level threshold. It does not support a mix of one-level threshold and two-level threshold storm control mode across ports at a time.
  - Traffic storm control monitors traffic statistics and generates system log for each level (lower and higher) and traffic type (unknown unicast, multicast, and broadcast) from Cisco Nexus Release 10.1(2).
  - The two-level threshold traffic storm control feature requires carving of a new Ternary Content Addressable Memory (TCAM) region with a fixed size of 512, and a reload of the device.
  - Traffic storm control for two-level threshold cannot coexist with the L2 Netflow feature, that is, presence of config layer2-switched flow monitor CLI, because of TCAM resource limitation.
  - The two-level threshold feature for traffic storm control does not support non-IP MC flood traffic (packet without an IP header) and packets-per-second mode.
  - Traffic storm control is not supported on Generic Online Diagnostics (GOLD) packets and sub-interface level.
  - If you were on a prior release, have upgraded to 10.1(2), and want to use the two-level storm control feature, then make sure that you configure the switch with the new storm control commands.
  - If you have configured the two-level storm control feature in version 10.1(2), and you want to downgrade to a previous version, then the new feature does not support downgrade. To downgrade, remove the configuration.

Layer 3 and later release support

- Beginning from Cisco Nexus Release 10.2(1), storm control does not allow multiple action configurations on an interface. If the previous action value is overwritten, the latest configured action value is used.
- Beginning with Cisco NX-OS Release 10.2(2)F, the storm control feature is supported on Cisco N9K-9332D-GX2B platform switches.
- Beginning with Cisco NX-OS Release 10.3(2)F, traffic storm control is supported on Layer 3 interfaces, and the following guidelines and limitations are applicable:
  - Traffic storm control supports devices that are only on one-level threshold storm control mode.
  - Layer 3 packets destined for the control plane such as ARP broadcast are not suppressed by storm control and are policed by the CoPP policer. However, storm control violation actions are triggered.
  - This feature is supported only on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches, and Cisco Nexus 9500 Series switches with FX and GX line cards.
- Beginning with Cisco NX-OS Release 10.3(3)F, the **system storm-control priority-policy drop-l3** command is introduced to prioritize storm control drop over the CoPP policer, and the following guidelines and limitations are applicable:

- This feature applies to Layer 3 control frames.
- This feature is supported only on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches, and Cisco Nexus 9500 Series switches with FX and GX line cards.
- This feature is applicable only for one-level threshold traffic storm control.

N9300 series smart switch limitations

- Beginning with Cisco NX-OS Release 10.6(2)F, the traffic storm control feature is supported on the following switches.
  - N9324C-SE1U
  - N9348Y2C6D-SE1U
- Supports only one-level threshold (1R2C).
- Does not support two-level threshold (2R3C).
- Does not support PPS.
- Supported interfaces are Layer 2 ports (physical, breakout, and port channel).
- Beginning with Cisco NX-OS Release 10.6(2)F, these commands are deprecated on N9300 Series smart switches:
  - **system storm control multi-threshold**
  - **storm-control multi unicast** {**level1** *level-value %* | **level2** *level-value %*}
  - **storm-control multi action1 trap action2 shutdown**

Default settings for traffic storm control

Use this reference to review the default settings for traffic storm control.

Table 120: Default traffic storm control parameters

Parameters	Default
Traffic storm control	Disabled
Threshold percentage	100

Configure traffic storm control for one-level threshold

You can set the percentage of total available bandwidth that the controlled traffic can use for one-level threshold.

Note

- Traffic storm control uses a 3.9-millisecond interval that can affect the behavior of traffic storm control.
- You must carve the n9k-arp-acl TCAM region before setting storm-control-cpu rate on port-channel. For information on configuring the TCAM region size, see the *Configuring ACL TCAM Region Sizes* section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}

Example:

```
switch(config)#interface ethernet 1/1
switch(config-if)#
```

Enters interface configuration mode.

Step 3 [**no**] **storm-control** {**broadcast** | **multicast** | **unicast**} **level** {*<level-value %>* |} **pps** < *pps-value* > }

Example:

```
switch(config-if)#storm-control unicast level 40
```

Example:

```
switch(config-if)#storm-control broadcast level pps 8000
```

Configures traffic storm control for traffic on the interface. You can also configure bandwidth level as a percentage either of port capacity or packets-per-second. The default state is disabled.

Step 4 [**no**] **storm-control action trap**

Example:

```
switch(config-if)#storm-control action trap
```

Generates an SNMP trap (defined in CISCO-PORT-STORM-CONTROL-MIB) and a syslog message when the traffic storm control limit is reached.

Step 5 **exit**

Example:

```
switch(config-if)#exit
switch(config)#
```

Exits interface configuration mode.

Step 6 (Optional) **show running-config interface** {**ethernet** *slot*/*port* | **port-channel** *number*}

Example:

```
switch#show running-config interface ethernet 1/1
```

Displays the traffic storm control configuration.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Prioritizing storm-control policer over the CoPP policer

By default, the Layer 3 control packets such as ARP broadcast are dropped by the storm-control policer, but policed by the CoPP policer and sent to CPU. This is because, for control packets, the storm-control policer has a lower priority compared to the CoPP policer.

Enabling the prioritize storm policer over the CoPP policer feature allows for increasing the priority of the storm-control policer over the CoPP policer for Layer 3 control packets. Consequently, the Layer 3 control packets such as ARP broadcast are dropped completely and not policed further by the CoPP policer, but policed by the storm-control policer.

Note When this feature is enabled, and if the incoming traffic is above the configured threshold value for a specific traffic type (multicast and broadcast), then the storm-control policer may not have control on the packets that get dropped. In this scenario, even the genuine Layer 3 control packets of that specific traffic type (multicast and broadcast) may get dropped.

Use the following command to prioritize the storm-control policer over the CoPP policer:

[**no**] **system storm-control priority-policy drop-l3**

Use the **no** form of this command to disable this feature.

Configure traffic storm control for two-level threshold

You can set the percentage of total available bandwidth that the controlled traffic can use for two-level threshold.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **system storm control multi-threshold**

Example:

```
switch(config)#system storm control multi-threshold
```

Enables global support for two-level storm control thresholds. This command is required only for configuring two-level threshold.

Step 3 **hardware access-list tcam region ing-storm-control 512**

Example:

```
switch(config)#hardware access-list tcam region ing-storm-control 512
```

Carves a new TCAM region with a fixed size of 512 for the two-level threshold.

After running the command, make sure that you reload the device.

Step 4 **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}

Example:

```
switch(config)#interface ethernet 1/1
switch(config-if)#
```

Enters interface configuration mode.

Step 5 [**no**] **storm-control multi unicast** {**level1** <*level-value %*> | **level2** <*level-value %*>}

Example:

```
switch(config-if)#storm-control multi unicast level1 5 level2 10
```

Configures traffic storm control for traffic on the interface for two-level threshold.

You can also configure bandwidth level as a percentage of port capacity. The default state is disabled.

Step 6 [**no**] **storm-control multi action1**{**trap** | **shutdown**} **action2**{**trap** | **shutdown**}

Example:

```
switch(config-if)#storm-control multi action1 trap action2 shutdown
```

Generates the following:

- An SNMP trap (defined in CISCO-PORT-STORM-CONTROL-MIB) to monitor the storm control.
- A syslog message when the traffic storm control limit is reached.

You can also configure the trap or shutdown action for the lower and higher level of storm control threshold. However, if you configure shutdown on lower threshold (level1) for a port, you must configure shutdown for higher threshold (level2) for that port.

Step 7 **exit**

Example:

```
switch(config-if)#exit
switch(config)#exit
switch#
```

Exits interface configuration mode.

Step 8 (Optional) **show running-config interface** {**ethernet** *slot*/*port* | **port-channel** *number*}

Example:

```
switch#show running-config interface ethernet 1/1
```

Displays the traffic storm control configuration.

Step 9 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Traffic storm control configuration verification commands

Use this reference to verify the Traffic storm control configuration.

To display traffic storm control configuration information, perform one of the following tasks:

Table 121: Traffic storm control configuration verification commands

Command	Purpose
show running-config interface	Displays the traffic storm control configuration.
show access-list storm-control arp-stats interface [ethernet] | **port-channel**] *number*	Displays the storm control statistics for arp packets on the interface.

Monitoring traffic storm control counters

This reference provides the commands required to monitor traffic storm control counters maintained by the Cisco NX-OS device.

Table 122: Monitoring traffic storm control counters

Command	Purpose
The following row is applicable only to one-level threshold.	
show interface [**ethernet** *slot* / *port* | **port-channel** *number*] **counters storm-control**	Displays the traffic storm control counters.
The following rows are applicable only to two-level threshold.	
show interface [**ethernet** *slot* / *port* | **port-channel** *number*] **counters storm-control multi-threshold**	Displays the list of the configured storm control values for all interfaces.

Command	Purpose
show interface [**ethernet** *slot* / *port* | **port-channel** *number*] **counters storm-control multi-threshold**	Displays the list of the configured storm control values for the interface.
show interface [**ethernet** *slot* / *port* | **port-channel** *number*] **counters storm-control multi-threshold unicast**	Displays the list of the unicast drops for both level1 and level2.
show interface [**ethernet** *slot* / *port* | **port-channel** *number*] **counters storm-control multi-threshold broadcast**	Displays the list of the broadcast drops for both level1 and level2.
show interface [**ethernet** *slot* / *port* | **port-channel** *number*] **counters storm-control multi-threshold multicast**	Displays the list of the multicast drops for both level1 and level2.

Configuration examples for traffic storm control

The following example shows how to configure traffic storm control for one-level threshold:

```
switch#configure terminal
switch(config)#interface Ethernet1/1
switch(config-if)#storm-control broadcast level 40
switch(config-if)#storm-control multicast level 40
switch(config-if)#storm-control unicast level 40
switch(config)#storm-control-cpu arp rate 150
```

The following example shows how to configure traffic storm control for two-level threshold:

```
switch#configure terminal
switch(config)#system storm control multi-threshold
switch(config)#hardware access-list tcam region ing-storm-control 512
switch(config)#interface Ethernet1/1
switch(config-if)#storm-control multi broadcast level1 5 level2 10
switch(config-if)#storm-control multi multicast level1 5 level2 10
switch(config-if)#storm-control multi unicast level1 5 level2 10
switch(config-if)#storm-control multi action1 trap action2 shutdown
```

The following example checks the programmed configured rate and the statistics of dropped ARP packets:

```
switch#show access-list storm-control-cpu arp-stats
interface port-channel 132
slot  1
=======
------------------------------------------------------------------------
ARP Policer Entry Statistics
------------------------------------------------------------------------
Interface port-channel132:
----------
Member Interface   Entry-ID  Rate     RedPacket Count     GreenPacket Count
-------------------------------------------------------------------------------
Ethernet1/35        3976        50           0                0
-------------------------------------------------------------------------------

slot  7
=======
------------------------------------------------------------------------
ARP Policer Entry Statistics
------------------------------------------------------------------------
Interface port-channel132:
```

```
----------
Member Interface   Entry-ID  Rate     RedPacket Count     GreenPacket Count
--------------------------------------------------------------------------------
```

System log examples for traffic storm control

The following example shows the system log for traffic storm control with one-level threshold:

```
%ETHPORT-5-STORM_CONTROL_ABOVE_THRESHOLD: Traffic in port Ethernet1/5 exceeds the configured
 threshold , action - Trap
```

The following example shows the system log for traffic storm control with two-level threshold:

```
%ETHPORT-5-STORM_CONTROL_ABOVE_THRESHOLD: Traffic in port Ethernet1/5 exceeds the configured
 Broadcast threshold level1[10%], action - Trap

%ETHPORT-5-STORM_CONTROL_ABOVE_THRESHOLD: Traffic in port Ethernet1/5 exceeds the configured
 Broadcast threshold level1[15%], action - Shutdown
```



Note The system log message includes the specific traffic type that exceeded the threshold and the level at which the traffic type reached the storm control action on an interface.

Additional references for traffic storm control

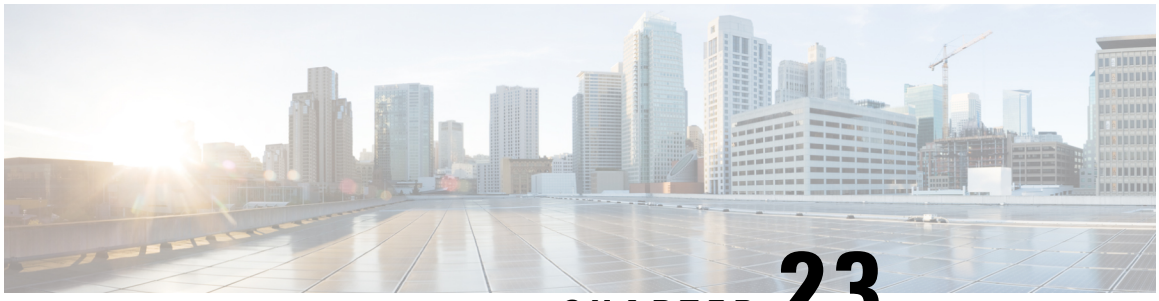
Use this reference to find additional references for traffic storm control.

This section includes additional information related to implementing traffic storm control.

Related Documents

Table 123: Related documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*



CHAPTER 23

Configuring Unicast RPF

This chapter describes how to configure unicast reverse path forwarding (uRPF) on Cisco NX-OS devices.

This chapter includes the following sections:

- About unicast RPF, on page 583
- Guidelines and limitations for unicast RPF, on page 585
- Unicast RPF parameter settings, on page 587
- Configure unicast RPF for Cisco Nexus 9500 switches with -R line cards, on page 588
- Configure unicast RPF for Cisco Nexus 9300 switches, on page 589
- Configuration examples for unicast RPF, on page 591
- Unicast RPF configuration verification, on page 592
- Reference materials for unicast RPF features, on page 592

About unicast RPF

Unicast RPF is a network security feature that

- discards IPv4 and IPv6 packets lacking a verifiable source address,
- protects against attacks using forged or spoofed addresses, and
- verifies packet sources by performing a reverse-path lookup in the routing table (FIB).

Attackers use methods like Smurf or Tribal Flood Network (TFN) to exploit forged or rapidly changing source IPv4 or IPv6 addresses, making it difficult for you to locate or filter these attacks. Enable Unicast RPF to help prevent these attacks by ensuring that only packets with valid source addresses matching the IP routing table are forwarded.

When you enable unicast RPF on an interface, the switch inspects all ingress packets. The switch confirms that the source address and interface are in the routing table and that the source matches the interface the packet arrived on. This process uses the Forwarding Information Base (FIB). Unicast RPF applies only to the ingress interface at the upstream end of a connection.

Unicast RPF verifies that any packet received has arrived on the best return path to its source by performing a reverse lookup in the FIB. If the packet comes from a valid reverse path, the switch forwards it; otherwise, the switch drops the packet to prevent attacks involving spoofed IP addresses.

Note Unicast RPF is an ingress function and is applied only on the ingress interface of a switch at the upstream end of a connection.

Note With unicast RPF, all equal-cost “best” return paths are considered valid. Unicast RPF works where multiple return paths exist if each path is equal to the others in routing cost (such as number of hops or weights) and the route is in the FIB. Unicast RPF also functions where Variants of Enhanced Interior Gateway Routing Protocol (EIGRP) are used, and unequal candidate paths to the source IP address may exist.

Unicast RPF can prevent DoS attacks such as Smurf or TFN, where attackers use forged or changing source IP addresses to evade detection.

How unicast RPF works

Review these key principles for implementing unicast RPF.

- Receive the packet at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). A matching route must exist in the FIB. Add routes to the FIB using static routes, network statements, or dynamic routing.
- Ensure that the IP source addresses at the receiving interface match the routing entry for the interface.
- Apply unicast RPF as an input function on the input interface at the upstream end of a connection.

Caution Use optional BGP attributes, such as weight and local preference, with caution to prevent attackers from changing the best path back to the source address. These changes can disrupt unicast RPF.

Summary

You can use unicast RPF for downstream networks, even if the downstream network has other connections to the Internet.

Workflow

When a packet is received at the interface where you have configured unicast RPF and ACLs, the Cisco NX-OS software performs these actions:

1. Check the input ACLs on the inbound interface.
2. Use unicast RPF to verify that the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.
3. Conduct a FIB lookup for packet forwarding.
4. Check the output ACLs on the outbound interface.

5. Forward the packet.

Guidelines and limitations for unicast RPF

Use this reference to review the guidelines and limitations for unicast RPF.

Unicast RPF (uRPF) has the following configuration guidelines and limitations:

Supported platforms

- uRPF is supported for the following platforms:
  - Cisco Nexus 9500 Series switches with N9K-X9636C-R and N9K-X9636Q-R line cards
  - Cisco Nexus 9500 Series switches with N9K-X9636C-RX line cards
  - Cisco Nexus 9300 platform switches (excluding the 9300-FXP switches)
- Beginning with Cisco NX-OS Release 10.1(2), uRPF is supported on:
  - Cisco Nexus 9300-GX/GX2 series switches and Cisco Nexus 9500 series switches with FX linecards (for IPv4 and IPv6)
  - Cisco Nexus 9500 series switches with EX linecards (for IPv4 only)
  - ToR and EoR switches that support vPC
- Beginning with Cisco NX-OS Release 9.2(1), uRPF is supported on:
  - Cisco Nexus 9300-EX Series switches (for IPv4 only)
  - Cisco Nexus 9300-FX/FX2 Series switches (for IPv4 and IPv6)
- Beginning with Cisco NX-OS Release 9.3(5), uRPF is supported on Cisco Nexus 9300-FX3 platform switches (for IPv4 and IPv6).
- Beginning with Cisco Nexus Release 9.3(1), uRPF is supported on Cisco Nexus 9500 Series switches with the family of modular EX/FX line cards (see Cisco Nexus 9500 Cloud-Scale Line Cards and Fabric Modules Data Sheet).



Note uRPF on the modular EX/ FX line cards is supported only in DUAL STACK MCAST routing mode. Specify the following configuration before enabling uRPF: `system routing template-dual-stack-mcast`. Refer to the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* on how to configure DUAL STACK MCAST routing mode.

From Cisco NX-OS Release 10.1(2), uRPF on the modular EX/ FX line cards is supported in default routing mode, too.

Configuration guidelines and limitations

- You must apply uRPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.

- The further downstream that you apply uRPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying uRPF on an aggregation device helps to mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying uRPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying uRPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy uRPF across Internet, intranet, and extranet resources mean the better the chances of mitigating large-scale network disruptions throughout the Internet community and of tracing the source of an attack.
- uRPF won't inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. Configure uRPF at a home gateway so that uRPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.
- You can use uRPF in any "single-homed" environment where there is only one access point out of the network or one upstream connection. Networks that have one access point provide symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet.
- Don't use uRPF on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, which means that multiple routes to the source of a packet exist. You should configure uRPF only where there is natural or configured symmetry.
- uRPF allows packets with 192.0.2.1 source and 192.0.2.2 destination to pass so that the Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP) can operate correctly.
- When uRPF is enabled, the amount of static routes to null0 the switch can install is limited to the value of "Max V4 Ucast DA TCAM table entries" in "show hardware internal forwarding table utilization".

R Series line card support

The following guidelines and limitations apply only to Cisco Nexus 9500 Series switches with a N9K-X9636C-R, N9K-X9636C-RX, or N9K-X9636Q-R line card:

- Beginning with Cisco NX-OS Release 9.2(1), for N9K-X9636C-R and N9K-X96136YC-R switches, you can configure only one version of the available IPv4 and IPv6 Unicast RPF command on an interface. However, this enables Unicast RPF for both IPv4 and IPv6.
- For strict uRPF to work, enable it on the ingress interface and the interface where the source IP address is learned.
- The switch hardware does not implement strict uRPF per the configured routing interface.
- Strict uRPF is implemented per learned route on strict uRPF-enabled interfaces.
- If a route is resolved as ECMP, strict uRPF falls back to loose mode.
- Because of the hardware limitation on the trap resolution, uRPF might not be applied on supervisor-bound packets via inband.
- For IP traffic, enable IPv4 and IPv6 configurations simultaneously.
- Due to hardware limitations, the N9K-X9636C-R, N9K-X9636C-RX, and N9K-X9636Q-R line cards support only the following combinations:

Table 124: Guidelines and limitations for unicast RPF

uRPF Configuration		Applied Traffic Check on Source IP Address			
IPv4	IPv6	IP Unipath	IP ECMP	MPLS Encap/VPN/ECMP	Unipath MPLS VPN for N9K-X9636C-RX Line Card
Disable	Disable	Allow	Allow	Allow	Allow
Loose	Loose	uRPF loose	uRPF loose	uRPF loose	uRPF strict
Strict	Strict	uRPF strict	uRPF loose	uRPF loose	uRPF strict

- Strict uRPF discards the ICMPv6 NA packets even if the destined interface receives them for the following Cisco NX-OS devices:
  - Line cards: N9K-X9564PX, N9K-X9564TX, N9K-X9536PQ, X9408PC-CFP2, X9464TX, X9464TX2
  - Uplink modules: N9K-M12PQ
  - Switches: 93128TX, 9396PX, 9396TX, 9372PX, 9372PX-E, 3164Q, 31128PQ

Cisco Nexus switch support guidelines and limitations

- Strict uRPF blocks the ICMP traffic destined to the interface through VxLAN for the following platforms:
  - Cisco Nexus 9200 platform switches
  - Cisco Nexus 9300-EX/FX/GX platform switches
  - Nexus 9500 switches with N9K- X9700-EX and N9K-X9700-FX line cards
- If Strict uRPF is configured, append the following commands for urpf strict mode to work for unresolved host behind a subnet:
  - **no system multicast dcs-check**
  - **hardware profile multicast max-limit lpm-entries 0**

Unicast RPF parameter settings

This table presents the default settings for Unicast RPF parameters.

Table 125: Default unicast RPF parameter settings

Parameters	Default
Unicast RPF	Disabled

Configure unicast RPF for Cisco Nexus 9500 switches with -R line cards

You can configure unicast RPF on an ingress interface for Cisco Nexus 9500 Series switches with an -R line card.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter interface configuration mode using the **interface ethernet** *slot* / *port* command.

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

Step 3 Specify an IPv4 or IPv6 address for the interface using the {**ip** | **ipv6**} **address** *ip-address/length* command.

Example:

```
switch(config-if)#ip address 172.23.231.240/23
```

Step 4 Configure unicast RPF on the interface for both IPv4 and IPv6 using the {**ip** | **ipv6**} **verify unicast source reachable-via any** command.

Example:

```
switch(config-if)#ip verify unicast source reachable-via any
```

Note
When you enable uRPF for IPv4 or IPv6 (using the **ip** or **ipv6** keywords), uRPF is enabled for both IPv4 and IPv6.

Step 5 (Optional) Display the IP information for an interface using the **show ip interface ethernet** *slot* / *port* command.

Example:

```
switch#show ip interface ethernet 2/3
```

Step 6 (Optional) Display the configuration for an interface in the running configuration using the **show running-config interface ethernet** *slot* / *port* command.

Example:

```
switch#show running-config interface ethernet 2/3
```

Step 7 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command,

Example:

```
switch#copy running-config startup-config
```

Configure unicast RPF for Cisco Nexus 9300 switches

You can configure one of the following Unicast RPF modes on an ingress interface for Cisco Nexus 9300 platform switches (excluding the 9300-FXP switches) running Cisco NX-OS Release 9.2(1) or a later release.

- Strict Unicast RPF mode

  A strict mode check is successful when Unicast RPF finds a match in the FIB for the packet source address and the ingress interface through which the packet is received matches one of the Unicast RPF interfaces in the FIB match. If this check fails, the packet is discarded. You can use this type of Unicast RPF check where packet flows are expected to be symmetrical.

- Loose Unicast RPF mode

  A loose mode check is successful when a lookup of a packet source address in the FIB returns a match and the FIB result indicates that the source is reachable through at least one real interface. The ingress interface through which the packet is received is not required to match any of the interfaces in the FIB result.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **system urpf disable**

Example:

```
switch(config)#no system urpf disable
```

Enables Unicast RPF on the switch.

Note
You must reload the Cisco NX-OS box to apply the Unicast RPF configuration.

Step 3 **interface ethernet** *slot* / *port*

Example:

```
switch(config)#interface ethernet 2/3
switch(config-if)#
```

Specifies an Ethernet interface and enters interface configuration mode.

Step 4 {**ip** | **ipv6**} **address** *ip-address/length*

Example:

```
switch(config-if)#ip address 172.23.231.240/23
```

Specifies an IPv4 or IPv6 address for the interface.

Step 5 {**ip** | **ipv6**} **verify unicast source reachable-via** {**any** [**allow-default**] | **rx**}

Example:

```
switch(config-if)#ip verify unicast source reachable-via any
```

Configures Unicast RPF on the interface for both IPv4 and IPv6.

You can enable IPv4 and IPv6 uRPF separately for the Cisco Nexus 9300-EX Series switches (for IPv4) and on Cisco Nexus 9300-FX/FX2 Series switches.

Note
When you enable Unicast RPF for IPv4 or IPv6 (using the **ip** or **ipv6** keyword), Unicast RPF is enabled for both IPv4 and IPv6.

You can configure only one version of the available IPv4 and IPv6 Unicast RPF command on an interface. When you configure one version, all the mode changes must be done by this version and all other versions will be blocked by that interface.

- The **any** keyword specifies loose Unicast RPF.
- If you specify the **allow-default** keyword, the source address lookup can match the default route and use that for verification.

  Note
  The **allow-default** keyword is not applicable in the ALPM routing mode.

  Note
  The source address lookup (in case of a loose Unicast RPF check) does not match the default route if you do not specify the **allow-default** keyword.

- The **rx** keyword specifies strict Unicast RPF.

Step 6 **exit**

Example:

```
switch(config-if)#exit
switch(config)#
```

Exits interface configuration mode.

Step 7 (Optional) **show ip interface ethernet** *slot* / *port*

Example:

```
switch#show ip interface ethernet 1/54 | grep -i "unicast reverse path forwarding"
IP unicast reverse path forwarding: none
```

Displays the IP information for an interface and verifies if Unicast RPF is enabled.

Step 8 (Optional) **show running-config interface ethernet** *slot* / *port*

Example:

```
switch#show running-config interface ethernet 2/3
```

Displays the configuration for an interface in the running configuration.

Step 9 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configuration examples for unicast RPF

You can enable loose and strict unicast Reverse Path Forwarding (RPF) on Nexus 9500 and 9300 Series switches. Find examples for both Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) packets.

Loose unicast RPF for IPv4 packets on a Nexus 9500 series switch with an -r line card

Configure loose unicast Reverse Path Forwarding for IPv4 packets on your Nexus 9500 Series switch with an R line card.

```
interface Ethernet2/3
ip address 172.16.0.0/12
ip verify unicast source reachable-via any
```

Loose unicast RPF for IPv6 packets on a Nexus 9500 series switch with an -r line card

Configure loose unicast Reverse Path Forwarding (RPF) for IPv6 packets on your Nexus 9500 Series switch with an R line card.

```
interface Ethernet2/1
ipv6 address 2001:DB8::/32
ipv6 verify unicast source reachable-via any
```

Loose unicast RPF for IPv4 packets on a Nexus 9300 platform switch

Configure strict unicast Reverse Path Forwarding (RPF) for IPv4 packets on your Nexus 9300 platform switch.

```
no system urpf disable
interface Ethernet2/3
ip address 172.16.0.0/12
ip verify unicast source reachable-via any
```

Loose unicast RPF for IPv6 packets on a Nexus 9300 platform switch

Configure loose unicast Reverse Path Forwarding (RPF) for IPv6 packets on your Nexus 9300 platform switch.

```
no system urpf disable
interface Ethernet2/1
ipv6 address 2001:DB8::/32
ipv6 verify unicast source reachable-via any
```

Strict unicast RPF for IPv4 packets on a Nexus 9300 platform switch

Configure strict unicast Reverse Path Forwarding (RPF) for IPv4 packets on your Nexus 9300 platform switch.

```
no system urpf disable
interface Ethernet2/2
```

```
ip address 172.16.0.0/12
ip verify unicast source reachable-via rx
```

Strict unicast RPF for IPv6 packets on a Nexus 9300 platform switch

Configure strict unicast Reverse Path Forwarding (RPF) for IPv6 packets on your Nexus 9300 platform switch.

```
no system urpf disable
interface Ethernet2/4
ipv6 address 2001:DB8::/32
ipv6 verify unicast source reachable-via rx
```

Unicast RPF configuration verification

To display unicast RPF configuration information, use the table.

The table lists commands and their purposes for verifying Unicast RPF configuration.

Table 126: Commands for verifying unicast RPF configuration

Command	Purpose
show running-config interface ethernet *slot* / *port*	Displays the interface configuration in the running configuration.
show running-config ip [**all**]	Displays the IPv4 configuration in the running configuration.
show running-config ipv6 [**all**]	Displays the IPv6 configuration in the running configuration.
show startup-config interface ethernet *slot* / *port*	Displays the interface configuration in the startup configuration.
show startup-config ip	Displays the IP configuration in the startup configuration.

Reference materials for unicast RPF features

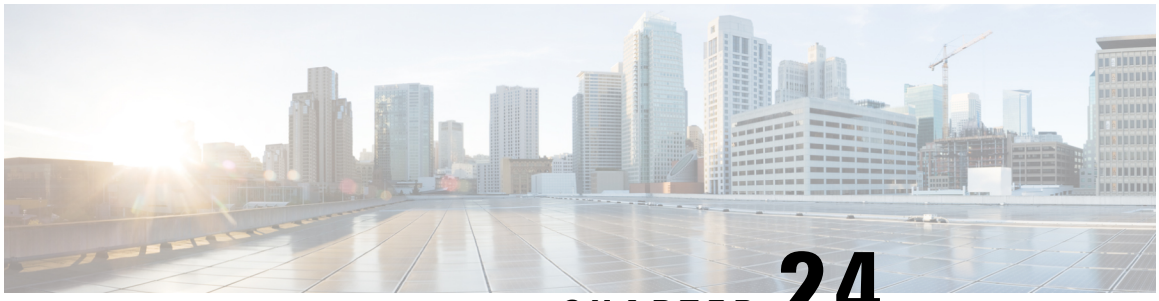
This section includes additional information related to implementing unicast RPF.

Related topic

Table 127: Additional information and related documents for implementing unicast RPF

Related Topic	Document Title
Data Management Engine (DME)-ized commands	For more information, see the Cisco Nexus 3000 and 9000 Series NX-API REST SDK User Guide and API Reference

Related Topic	Document Title
MPLS VPN	For more information, see the Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide



CHAPTER 24

Configuring Switchport Blocking

This chapter describes how to configure switchport blocking on the Cisco NX-OS device.

This chapter includes the following sections:

- About switchport blocking, on page 595
- Guidelines and limitations for switchport blocking, on page 595
- Default settings for switchport blocking, on page 596
- Configure switchport blocking, on page 596
- Switchport blocking configuration verification commands, on page 597
- Configuration example for switchport blocking, on page 597

About switchport blocking

Occasionally, unknown multicast or unicast traffic is flooded to a switch port because a MAC address has timed out or has not been learned by the switch. Security issues could arise if unknown multicast and unicast traffic is forwarded to a switch port. You can enable switchport blocking to guarantee that no multicast or unicast traffic is flooded to the port.

Guidelines and limitations for switchport blocking

Use this reference to review the guidelines and limitations for switchport blocking.

Switchport blocking has these configuration guidelines and limitations:

- Switchport blocking applies only to egress ports while traffic storm control applies only to ingress ports.
- Switchport blocking is supported on all switched ports (including PVLAN ports) and is applied to all VLANs on which the port is forwarding.
- Switchport blocking is not supported for FEX ports.
- When you block unknown multicast or unicast traffic for a port channel, it is blocked on all ports in the port-channel group.
- Switchport blocking provides only a single level of control, preventing the flooding of all unknown egress multicast or unicast packets on the specified port.

- Switchport blocking drops control packets that originate from the CPU on Cisco Nexus 9500 Series switches. It does not drop packets on Cisco Nexus 9300 Series switches.

Default settings for switchport blocking

Use this reference to review the default settings for switchport blocking.

This table lists the default settings for switchport blocking parameters.

Table 128: Default switchport blocking parameters

Parameters	Default
Switchport blocking	Disabled

Configure switchport blocking

By default, the switch floods packets with unknown destination MAC addresses to all ports. To prevent this behavior, configure a port to block unknown multicast or unicast packets.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter interface configuration mode.

Example:

```
switch(config)#interface ethernet 1/1
switch(config-if)#
```

Step 3 Prevent the flooding of unknown multicast or unicast packets on the specified interface.

Example:

```
switch(config-if)#switchport block unicast
```

Use the **no** form of this command to resume normal forwarding on the port.

Step 4 (Optional) Display the switchport blocking configuration.

Example:

```
switch#show interface ethernet 1/1 switchport
```

Step 5 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Switchport blocking configuration verification commands

Use this reference to verify the Switchport blocking configuration.

To display switchport blocking configuration information, perform one of the following tasks:

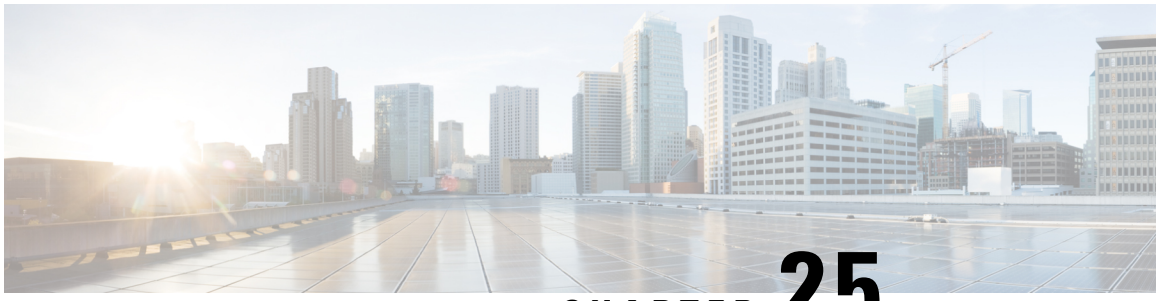
Table 129: Switchport blocking configuration verification commands

Command	Purpose
show interface switchport	Displays the switchport blocking configuration for all interfaces.
show interface {**ethernet** *slot* / *port* | **port-channel** *number*} **switchport**	Displays the switchport blocking configuration for the specified interface.
show running-config interface [**ethernet** *slot* / *port* | **port-channel** *number*]	Displays the switchport blocking configuration in the running configuration.

Configuration example for switchport blocking

The following example shows how to block multicast and unicast flooding on Ethernet interface 1/2 and how to verify the configuration:

```
switch#configure terminal
switch(config)#interface ethernet 1/2
switch(config-if)#switchport block multicast
switch(config-if)#switchport block unicast
switch#show running-config interface ethernet 1/2
!Command: show running-config interface Ethernet1/2
!Time: Wed Apr 15 16:25:48 2015
version 79.2(1)
interface Ethernet1/2
switchport
switchport block multicast
switchport block unicast
```



CHAPTER 25

Configure Control Plane Policing

This chapter describes how to configure Control Plane Policing on Cisco NX-OS devices.

- About CoPP, on page 599
- CoPP configuration guidelines, on page 617
- Default settings for CoPP, on page 621
- Configuring CoPP, on page 622
- Protocol ACL filtering added for egress CoPP, on page 629
- Copp configuration verification commands, on page 632
- Display the CoPP configuration status, on page 634
- Monitor CoPP, on page 634
- Monitoring CoPP with SNMP, on page 634
- Clear the CoPP statistics, on page 635
- Configuration examples for CoPP, on page 635
- Additional references for CoPP, on page 638

About CoPP

Control Plane Policing (CoPP) protects the control plane and separates it from the data plane, which ensures network stability, reachability, and packet delivery.

This feature allows a policy map to be applied to the control plane. This policy map looks like a normal QoS policy and is applied to all traffic entering the switch from a non-management port. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

The Cisco NX-OS device provides CoPP to prevent DoS attacks from impacting performance. Such attacks, whether inadvertent or malicious, typically involve high rates of traffic directed to the supervisor module or CPU.

The supervisor module divides the traffic that it manages into three functional components or planes:

- Data plane—Handles all the data traffic. The basic functionality of a Cisco NX-OS device is to forward packets from one interface to another. The packets that are not meant for the switch itself are called the transit packets. These packets are handled by the data plane.
- Control plane—Handles all routing protocol control traffic. These protocols, such as the Border Gateway Protocol (BGP) and the Open Shortest Path First (OSPF) Protocol, send control packets between devices. These packets are destined to router addresses and are called control plane packets.

- Management plane—Runs the components meant for Cisco NX-OS device management purposes such as the command-line interface (CLI) and Simple Network Management Protocol (SNMP).

The supervisor module has both the management plane and control plane and is critical to the operation of the network. Any disruption or attacks to the supervisor module will result in serious network outages. Excessive traffic to the supervisor module could overload and slow down the performance of the entire Cisco NX-OS device. Additionally, a DoS attack on the supervisor module could generate IP traffic streams to the control plane at a very high rate. This would force the control plane to spend significant time handling these packets and prevent it from processing genuine traffic.

Examples of DoS attacks include:

- Internet Control Message Protocol (ICMP) echo requests
- IP fragments
- TCP SYN flooding

These attacks can impact the device performance and have these negative effects:

- reduced service quality (such as poor voice, video, or critical applications traffic)
- high route processor or switch processor CPU utilization
- route flaps due to loss of routing protocol updates or keepalives
- unstable Layer 2 topology
- slow or unresponsive interactive sessions with the CLI
- processor resource exhaustion, including memory and buffer depletion
- indiscriminate drops of incoming packets



Caution It is important to ensure that you protect the supervisor module from accidental or malicious attacks by configuring control plane protection.

Control plane protection

To protect the control plane, the Cisco NX-OS device segregates packets destined for the control plane into separate classes. After these classes are identified, the Cisco NX-OS device polices the packets to prevent the supervisor module from being overwhelmed.

Control plan packet types

Different types of packets can reach the control plane.

- Receive packets: packets that have the destination address of a router. The destination address can be a Layer 2 address (such as a router MAC address) or a Layer 3 address (such as the IP address of a router interface). These packets include router updates and keepalive messages. Multicast packets can also be in this category where packets are sent to multicast addresses that are used by a router.

- Exception packets: packets that need special handling by the supervisor module. For example, if a destination address is not present in the Forwarding Information Base (FIB) and results in a miss, the supervisor module sends an ICMP unreachable packet back to the sender. Another example is a packet with IP options set.

  These exceptions are possible from line cards only:

  - match exception ip option
  - match exception ipv6 option
  - match exception ttl-failure

  These exceptions are possible from fabric modules only:

  - match exception ipv6 icmp unreachable
  - match exception ip icmp unreachable

  These exceptions are possible from line cards and fabric modules:

  - match exception mtu-failure

- Redirected packets: packets that are redirected to the supervisor module.
- Glean packets: if a Layer 2 MAC address for a destination IP address is not present in the FIB, the supervisor module receives the packet and sends an ARP request to the host.

All of these different packets could be maliciously used to attack the control plane and overwhelm the Cisco NX-OS device. CoPP classifies these packets into different classes and provides a mechanism to control the rate at which the supervisor module receives each packet.

Classification for CoPP

To ensure effective protection, the device classifies packets that reach the supervisor modules. This classification allows you to apply different rate control policies based on the packet type. For example, you might be less strict with protocol packets such as Hello messages and more strict with packets sent to the supervisor module due to specific IP options. You configure packet classifications and rate control policies using class maps and policy maps.

Egress CoPP

Beginning with Cisco NX-OS Release 10.2(3)F, egress CoPP is supported on the Nexus 93180YC-EX, Nexus 93180YC-FX, Nexus 93240YC-FX2, Nexus 93360YC-FX2, Nexus 9336C-FX2, Nexus 9336C-FX2-E, Nexus 93180YC-FX3, N9K-C9316D-GX, N9K-C93600CD-GX, Nexus 9364C-GX, N9K-C9332D-GX2B , Nexus 9364C and Nexus 9332C CloudScale switches.

Egress CoPP can be applied on top of custom/default CoPP policy.

Rate controlling mechanisms

Once the packets are classified, the device has mechanisms to control how quickly packets reach the supervisor module. The two primary mechanisms are policing and rate limiting.

Using hardware policers, you can define actions for traffic based on whether it meets specific conditions. These actions include transmitting, marking down, or dropping the packet.

You can configure these parameters for policing:

- Committed information rate (CIR): Desired bandwidth, specified as a bit rate or a percentage of the link rate.
- Committed burst (BC): Size of a traffic burst that can exceed the CIR within a given unit of time and not impact scheduling.

In addition, you can configure separate actions, such as transmit or drop, for traffic that conforms or violates policy.

For more information on policing parameters, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

Dynamic and static CoPP ACLs

CoPP access control lists (ACLs) are classified as either dynamic or static. Cisco Nexus 9300 and 9500 Series switches use only dynamic CoPP ACLs. Cisco Nexus 9200 Series switches use both dynamic and static CoPP ACLs.

Dynamic CoPP ACLs work only for Forwarding Information Base (FIB)-based supervisor redirected packets. Static CoPP ACLs work for ACL-based supervisor redirected packets. Dynamic CoPP ACLs are supported for myIP and link-local multicast traffic. Static CoPP ACLs are supported for all other types of traffic.

Static CoPP ACLs are identified by a substring. Any ACL that has one of these substrings is categorized as a static CoPP ACL.

MAC-based static CoPP ACL substrings:

- acl-mac-cdp-udld-vtp
- acl-mac-cfsoe
- acl-mac-dot1x
- acl-mac-l2-tunnel
- acl-mac-l3-isis
- acl-mac-lacp
- acl-mac-lldp
- acl-mac-sdp-srp
- acl-mac-stp

Dynamic and static CoPP ACLs:

- acl-mac-undesirable

Protocol-based static CoPP ACL substrings:

- acl-dhcp
- acl-dhcp-relay-response
- acl-dhcp6
- acl-dhcp6-relay-response

- acl-ptp

Multicast-based static CoPP ACL substrings:

- acl-igmp

For more information on static CoPP ACLs, see CoPP configuration guidelines, on page 617 .

Default policing policies

When you start your Cisco NX-OS device for the first time, the software installs the default copp-system-p-policy-strict policy. This policy protects the supervisor module from DoS attacks. You can set the protection level by choosing a CoPP policy option from the initial setup utility:

- **Strict**—This policy is 1 rate and 2 color.
- **Moderate**—This policy is 1 rate and 2 color. The important class burst size is greater than that of the strict policy but less than that of the lenient policy.
- **Lenient**—This policy is 1 rate and 2 color. The important class burst size is greater than that of the moderate policy but less than that of the dense policy.
- **Dense**—This policy is 1 rate and 2 color. The policer CIR values are lower than those in the strict policy.
- **Skip**—No control plane policy is applied. (Cisco does not recommend using the Skip option, as it may impact the network control plane.

If you do not select an option or do not run the setup utility, the software applies strict policing. We recommend that you start with the strict policy and later modify the CoPP policies as required.

Note Strict policing is not applied by default when using POAP, so you must configure a CoPP policy.

The copp-system-p-policy policy has optimized values suitable for basic device operations. You must add specific class and access-control list (ACL) rules that meet your DoS protection requirements. The default CoPP policy does not change when you upgrade the software.

Caution If you select the skip option and do not configure CoPP protection, your Cisco NX-OS device can be vulnerable to DoS attacks.

You can reassign the default CoPP policy by runnign the **setup** command at the CLI prompt or by using the **copp profile** command.

Related Topics

Change or reapply the default CoPP policy, on page 628

Default class maps

The copp-system-class-critical class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-critical
  match access-group name copp-system-p-acl-bgp
  match access-group name copp-system-p-acl-rip
```

```
  match access-group name copp-system-p-acl-vpc
  match access-group name copp-system-p-acl-bgp6
  match access-group name copp-system-p-acl-ospf
  match access-group name copp-system-p-acl-rip6
  match access-group name copp-system-p-acl-eigrp
  match access-group name copp-system-p-acl-ospf6
  match access-group name copp-system-p-acl-eigrp6
  match access-group name copp-system-p-acl-auto-rp
  match access-group name copp-system-p-acl-mac-l3-isis
```

The copp-system-class-exception class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-exception
  match exception ip option
  match exception ip icmp unreachable
  match exception ipv6 option
  match exception ipv6 icmp unreachable
```

The copp-system-class-exception-diag class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-exception-diag
  match exception ttl-failure
  match exception mtu-failure
```

The copp-system-class-important class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-important
  match access-group name copp-system-p-acl-hsrp
  match access-group name copp-system-p-acl-vrrp
  match access-group name copp-system-p-acl-hsrp6
  match access-group name copp-system-p-acl-vrrp6
  match access-group name copp-system-p-acl-mac-lldp
```

The copp-system-class-l2-default class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-l2-default
  match access-group name copp-system-p-acl-mac-undesirable
```

The copp-system-class-l2-unpoliced class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-l2-unpoliced
  match access-group name copp-system-p-acl-mac-stp
  match access-group name copp-system-p-acl-mac-lacp
  match access-group name copp-system-p-acl-mac-cfsoe
  match access-group name copp-system-p-acl-mac-sdp-srp
  match access-group name copp-system-p-acl-mac-l2-tunnel
  match access-group name copp-system-p-acl-mac-cdp-udld-vtp
```

The copp-system-class-l3mc-data class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-l3mc-data
  match exception multicast rpf-failure
  match exception multicast dest-miss
```

The copp-system-class-l3uc-data class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-l3uc-data
  match exception glean
```

The copp-system-class-management class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-management
  match access-group name copp-system-p-acl-ftp
  match access-group name copp-system-p-acl-ntp
  match access-group name copp-system-p-acl-ssh
  match access-group name copp-system-p-acl-http
  match access-group name copp-system-p-acl-ntp6
  match access-group name copp-system-p-acl-sftp
```

```
  match access-group name copp-system-p-acl-snmp
  match access-group name copp-system-p-acl-ssh6
  match access-group name copp-system-p-acl-tftp
  match access-group name copp-system-p-acl-https
  match access-group name copp-system-p-acl-snmp6
  match access-group name copp-system-p-acl-tftp6
  match access-group name copp-system-p-acl-radius
  match access-group name copp-system-p-acl-tacacs
  match access-group name copp-system-p-acl-telnet
  match access-group name copp-system-p-acl-radius6
  match access-group name copp-system-p-acl-tacacs6
  match access-group name copp-system-p-acl-telnet6
```

The copp-system-class-monitoring class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-monitoring
  match access-group name copp-system-p-acl-icmp
  match access-group name copp-system-p-acl-icmp6
  match access-group name copp-system-p-acl-traceroute
```

The copp-system-class-multicast-host class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-multicast-host
  match access-group name copp-system-p-acl-mld
```

The copp-system-class-multicast-router class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-multicast-router
  match access-group name copp-system-p-acl-pim
  match access-group name copp-system-p-acl-msdp
  match access-group name copp-system-p-acl-pim6
  match access-group name copp-system-p-acl-pim-reg
  match access-group name copp-system-p-acl-pim6-reg
  match access-group name copp-system-p-acl-pim-mdt-join
```

The copp-system-class-nat-flow class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-nat-flow
  match exception nat-flow
```

The copp-system-class-ndp class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-ndp
  match access-group name copp-system-p-acl-ndp
```

The copp-system-class-normal class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-normal
  match access-group name copp-system-p-acl-mac-dot1x
  match protocol arp
```

The copp-system-class-normal-dhcp class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-normal-dhcp
  match access-group name copp-system-p-acl-dhcp
  match access-group name copp-system-p-acl-dhcp6
```

The copp-system-class-normal-dhcp-relay-response class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-normal-dhcp-relay-response
  match access-group name copp-system-p-acl-dhcp-relay-response
  match access-group name copp-system-p-acl-dhcp6-relay-response
```

The copp-system-class-normal-igmp class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-normal-igmp
  match access-group name copp-system-p-acl-igmp
```

The copp-system-class-redirect class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-redirect
  match access-group name copp-system-p-acl-ptp
```

The copp-system-class-undesirable class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-undesirable
  match access-group name copp-system-p-acl-undesirable
  match exception multicast sg-rpf-failure
```

The copp-system-class-fcoe class uses this configuration.

```
class-map type control-plane match-any copp-system-p-class-fcoe
  match access-group name copp-system-p-acl-mac-fcoe
```



Note The copp-system-class-fcoe class is not supported for Cisco Nexus 9200 Series switches.

Strict default CoPP policy

On Cisco Nexus 9200 Series switches, the strict CoPP policy has the following configuration:



Note The CIR value for copied or custom CoPP profiles which are pre-existing before upgrade to a new image will have the same CIR values as before. The new CoPP profiles which are copied from default profiles in the new image will have a new CIR value.

```
policy-map type control-plane copp-system-p-policy-strict
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 2200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 32000 bytes conform transmit violate drop
```

```
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 32000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 32000 bytes conform transmit violate drop
```

On Nexus 9300 and 9500 Series switches, the strict CoPP policy has the following configuration:

```
switch#show policy-map type control-plane name copp-system-p-policy-strict
  policy-map type control-plane copp-system-p-policy-strict
    class copp-system-p-class-l3uc-data
      set cos 1
      police cir 800 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-critical
      set cos 7
      police cir 36000 kbps bc 1280000 bytes conform transmit violate drop
    class copp-system-p-class-important
      set cos 6
      police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
    class copp-system-p-class-openflow
      set cos 5
      police cir 1000 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-multicast-router
      set cos 6
      police cir 2600 kbps bc 128000 bytes conform transmit violate drop
    class copp-system-p-class-multicast-host
      set cos 1
      police cir 1000 kbps bc 128000 bytes conform transmit violate drop
    class copp-system-p-class-l3mc-data
      set cos 1
      police cir 2400 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-normal
      set cos 1
      police cir 2200 kbps bc 128000 bytes conform transmit violate drop
    class copp-system-p-class-ndp
      set cos 6
      police cir 1400 kbps bc 32000 bytes conform transmit violate drop
```

```
    class copp-system-p-class-normal-dhcp
      set cos 1
      police cir 1300 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-normal-dhcp-relay-response
      set cos 1
      police cir 1500 kbps bc 64000 bytes conform transmit violate drop
    class copp-system-p-class-normal-igmp
      set cos 3
      police cir 3000 kbps bc 64000 bytes conform transmit violate drop
    class copp-system-p-class-redirect
      set cos 1
      police cir 1800 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-exception
      set cos 1
      police cir 150 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-exception-diag
      set cos 1
      police cir 150 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-management
      set cos 2
      police cir 36000 kbps bc 512000 bytes conform transmit violate drop
    class copp-system-p-class-monitoring
      set cos 1
      police cir 360 kbps bc 128000 bytes conform transmit violate drop
    class copp-system-p-class-l2-unpoliced
      set cos 7
      police cir 50 mbps bc 8192000 bytes conform transmit violate drop
    class copp-system-p-class-undesirable
      set cos 0
      police cir 200 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-fcoe
      set cos 6
      police cir 30000 kbps bc 825000 bytes conform transmit violate drop
    class copp-system-p-class-nat-flow
      set cos 7
      police cir 800 kbps bc 64000 bytes conform transmit violate drop
    class copp-system-p-class-l3mcv6-data
      set cos 1
      police cir 2400 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-undesirablev6
      set cos 0
      police cir 200 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-l2-default
      set cos 0
      police cir 400 kbps bc 32000 bytes conform transmit violate drop
    class class-default
      set cos 0
      police cir 400 kbps bc 32000 bytes conform transmit violate drop
switch#show policy-map type control-plane name copp-system-p-policy-strict
  policy-map type control-plane copp-system-p-policy-strict
    class copp-system-p-class-l3uc-data
      set cos 1
      police cir 250 pps bc 256 packets conform transmit violate drop
    class copp-system-p-class-critical
      set cos 7
      police cir 19000 pps bc 1024 packets conform transmit violate drop
    class copp-system-p-class-important
      set cos 6
      police cir 6000 pps bc 1024 packets conform transmit violate drop
    class copp-system-p-class-openflow
      set cos 5
      police cir 2000 pps bc 256 packets conform transmit violate drop
    class copp-system-p-class-multicast-router
      set cos 6
```

```
  police cir 3000 pps bc 1024 packets conform transmit violate drop
class copp-system-p-class-multicast-host
  set cos 1
  police cir 2000 pps bc 1024 packets conform transmit violate drop
class copp-system-p-class-l3mc-data
  set cos 1
  police cir 3000 pps bc 1024 packets conform transmit violate drop
class copp-system-p-class-normal
  set cos 1
  police cir 2500 pps bc 512 packets conform transmit violate drop
class copp-system-p-class-ndp
  set cos 6
  police cir 1500 pps bc 512 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp
  set cos 1
  police cir 750 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp-relay-response
  set cos 1
  police cir 750 pps bc 512 packets conform transmit violate drop
class copp-system-p-class-normal-igmp
  set cos 3
  police cir 6000 pps bc 512 packets conform transmit violate drop
class copp-system-p-class-redirect
  set cos 1
  police cir 1500 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-exception
  set cos 1
  police cir 125 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-exception-diag
  set cos 1
  police cir 125 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-management
  set cos 2
  police cir 5000 pps bc 1024 packets conform transmit violate drop
class copp-system-p-class-monitoring
  set cos 1
  police cir 375 pps bc 1024 packets conform transmit violate drop
class copp-system-p-class-l2-unpoliced
  set cos 7
  police cir 64000 pps bc 16384 packets conform transmit violate drop
class copp-system-p-class-undesirable
  set cos 0
  police cir 125 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-fcoe
  set cos 6
  police cir 1000 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-nat-flow
  set cos 7
  police cir 125 pps bc 256 packets conform transmit violate drop
class copp-system-p-class-l3mcv6-data
  set cos 1
  police cir 3000 pps bc 1024 packets conform transmit violate drop
class copp-system-p-class-l2-default
  set cos 0
  police cir 125 pps bc 256 packets conform transmit violate drop
class class-default
  set cos 0
  police cir 125 pps bc 256 packets conform transmit violate drop
```



Note The values in the examples are for reference only. Actual CoPP rates may vary based on the switches deployed.

Moderate default CoPP policy

On Cisco Nexus 9200 Series switches, the moderate CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-moderate
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 1920000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1920000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 96000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
```

```
    police cir 400 kbps bc 48000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 48000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the moderate CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-moderate
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 3000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 96 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 300 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 15 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-fcoe
```

```
    set cos 6
    police cir 1500 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 50 pps bc 48 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 48 packets conform transmit violate drop
```

Lenient default CoPP policy

On Cisco Nexus 9200 Series switches, the lenient CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-lenient
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 2560000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 2560000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
```

```
    police cir 150 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 64000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 64000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series switches, the lenient CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-lenient
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 3000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 64 packets conform transmit violate drop
```

```
  class copp-system-p-class-monitoring
    set cos 1
    police cir 300 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 15 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-fcoe
    set cos 6
    police cir 1500 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 50 pps bc 64 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 64 packets conform transmit violate drop
```

Dense default CoPP policy

On Cisco Nexus 9200 Series switches, the dense CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-dense
      class copp-system-p-class-l3uc-data
      set cos 1
      police cir 800 kbps bc 32000 bytes conform transmit violate drop
      class copp-system-p-class-critical
      set cos 7
      police cir 4500 kbps bc 1280000 bytes conform transmit violate drop
      class copp-system-p-class-important
      set cos 6
      police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
      class copp-system-p-class-multicast-router
      set cos 6
      police cir 370 kbps bc 128000 bytes conform transmit violate drop
      class copp-system-p-class-management
      set cos 2
      police cir 2500 kbps bc 128000 bytes conform transmit violate drop
      class copp-system-p-class-multicast-host
      set cos 2
      police cir 300 kbps bc 128000 bytes conform transmit violate drop
      class copp-system-p-class-l3mc-data
      set cos 1
      police cir 600 kbps bc 32000 bytes conform transmit violate drop
      class copp-system-p-class-normal
      set cos 1
      police cir 1400 kbps bc 128000 bytes conform transmit violate drop
      class copp-system-p-class-ndp
      set cos 1
      police cir 350 kbps bc 32000 bytes conform transmit violate drop
      class copp-system-p-class-normal-dhcp
      set cos 1
      police cir 750 kbps bc 128000 bytes conform transmit violate drop
      class copp-system-p-class-normal-dhcp-relay-response
      set cos 1
      police cir 750 kbps bc 128000 bytes conform transmit violate drop
      class copp-system-p-class-normal-igmp
      set cos 3
      police cir 1400 kbps bc 128000 bytes conform transmit violate drop
      class copp-system-p-class-redirect
```

```
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-exception
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-exception-diag
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 128000 bytes conform transmit violate drop
    class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
    class copp-system-p-class-undesirable
    set cos 0
    police cir 100 kbps bc 32000 bytes conform transmit violate drop
    class copp-system-p-class-l2-default
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
    class class-default
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series switches, the dense CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-dense
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 2500 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 2
    police cir 1000 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 1200 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 750 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 1
    police cir 750 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 150 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 2500 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
```

```
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception-diag
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-monitoring
  set cos 1
  police cir 50 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l2-unpoliced
  set cos 7
  police cir 20000 pps bc 8192 packets conform transmit violate drop
class copp-system-p-class-undesirable
  set cos 0
  police cir 15 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-fcoe
  set cos 6
  police cir 750 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l2-default
  set cos 0
  police cir 25 pps bc 32 packets conform transmit violate drop
class class-default
  set cos 0
  police cir 25 pps bc 32 packets conform transmit violate drop
```

Packets per second credit limit

The aggregate packets per second (PPS) for a given policy (sum of PPS of each class part of the policy) is capped by an upper PPS Credit Limit (PCL). If an increase in PPS of a given class causes a PCL exceed, the configuration is rejected. To increase the desired PPS, the additional PPS beyond PCL should be decreased from other class(es).

Modular QoS Command-Line interface

CoPP uses the Modular Quality of Service Command-Line Interface (MQC). MQC is a CLI structure that allows you to define a traffic class. You can also create a traffic policy (policy map) and attach the traffic policy to an interface. The traffic policy contains the CoPP feature, which is applied to the traffic class.

Procedure

Step 1 Define a traffic class using the **class-map** command.

Example:

```
class-map type control-plane copp-sample-class
```

A traffic class is used to classify traffic. This example shows how to create a new class-map called copp-sample-class:

Step 2 Create a traffic policy using the **policy-map** command.

A traffic policy (policy map) contains a traffic class and one or more CoPP features that will be applied to the traffic class. The CoPP features in the traffic policy specify how to treat the classified traffic.

Step 3 Attach the traffic policy (policy map) to the control plane using the **control-plane** and **service-policy** commands.

Example:

```
control-plane service-policy input copp-system-policy
```

This example shows how to attach the policy map to the control plane:

Note
The copp-system-policy is always configured and applied. There is no need to use this command explicitly.

Copp and the management interface

The Cisco NX-OS device supports only hardware-based CoPP, which does not support the management interface (mgmt0). The out-of-band mgmt0 interface connects directly to the CPU and does not pass through the in-band traffic hardware where CoPP is implemented.

On the mgmt0 interface, ACLs can be configured to give or deny access to a particular type of traffic.

Related Topics

Configure IP ACLs, on page 307
Configuring MAC ACLs, on page 393

CoPP configuration guidelines

Use this reference to review copp configuration guidelines.

A CoPP policy is a set of configuration rules that

- manage control plane traffic by classifying and policing traffic to the CPU,
- enforce protections against unwanted or excessive control traffic, and
- ensure critical management or routing functions are preserved on the device.

CoPP has these configuration guidelines and limitations:

- We recommend that you use the strict default CoPP policy initially and then later modify the CoPP policies that are based on the data center and application requirements.
- Customizing CoPP is an ongoing process. You must configure CoPP for the protocols and features used in your environment and the supervisor features required by the server environment. As protocols and features change, modify CoPP accordingly.
- We recommend that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to modify the CoPP policies.
- All traffic not specified in other class maps is placed in the default class. Monitor drops in this class and investigate whether they result from unwanted traffic or from a feature that needs to be configured.
- All broadcast traffic is sent through CoPP logic to determine which packets, such as ARP and DHCP, must be redirected with an access control list (ACL) to the router processor. Broadcast traffic that does not need to be redirected is matched against CoPP logic. Both conforming and violated packets are

counted in the hardware but are not sent to the CPU. Broadcast traffic that must be sent to the CPU and that which does not must be separated into different classes.

- After you have configured CoPP, delete anything that is not being used, such as old class maps and unused routing protocols.
- Ensure that the CoPP policy does not filter critical traffic, such as routing protocols or interactive access. Filtering this traffic could prevent remote access and require a console connection.
- Perform CoPP policy parameter changes only during a maintenance window. Modifying the CIR or committed burst (BC) value of an existing CoPP policy class while the switch is carrying production traffic can cause a transient disruption to LACP, port-channel, and vPC operation.

Switch specific limitations

- CoPP processing comprises of 2 stages: In the first stage, the actual packet size is reused in each class policy, however when the packet enters the second stage, an internal header of 44 bytes is added. This causes an alteration in the conform or violation policies of all the CoPP classes. This limitation is applicable to Cisco Nexus 9300-FX, Nexus 9300-FX2, Nexus 9364C, Nexus 9332C, and 9300-GX platform switches.
- First-generation Cisco Nexus 9000 Series switches (non - EX/ FX/FX2), do not support source-based CoPP. This limitation does not exist for cloud scale ASIC-based Cisco Nexus switches.
- The **match-all** option is not supported in CoPP class-map and it always defaults to the **match-any** option.
- The Cisco NX-OS software does not support egress CoPP or silent mode. CoPP is supported only on ingress (you cannot use the **service-policy output copp** command to the control plane interface).
- The copp-system-class-fcoe class is not supported for Cisco Nexus 9200 Series switches.
- Cloudscale IPv6 link-local BGP support requires carving > 512 ing-sup TCAM region (this requires a reload to take effect).
- To avoid traffic loss during traffic impact, configure the CoPP class normal CIR value to 2200 kbps on Cisco Nexus 9300 GX/FX/FX2/FX3, 9504-FM-G, and 9508-FM-G switches and X9716D-GX line cards.
- Beginning with Cisco NX-OS Release 10.3(2)F, source IP based filtering in CoPP is supported on Cisco Nexus 9504 and 9508 modular chassis with R/RX line cards.



Note For IPv6, source IP based filtering is supported up to 24b MSB.

Configuration limitations

- You can use the access control entry (ACE) hit counters in the hardware only for ACL logic. Use the software ACE hit counters and the **show access-lists** and **show policy-map type control-plane** commands to evaluate CPU traffic.
- The Cisco NX-OS device hardware performs CoPP on a per-forwarding-engine basis. CoPP does not support distributed policing. Therefore, you should choose rates so that the aggregate traffic does not overwhelm the supervisor module.

- If multiple flows map to the same class, individual flow statistics will not be available.
- You cannot disable CoPP. If you attempt to disable it, packets are rate limited at 50 packets per seconds .
- Skip CoPP policy option has been removed from the Cisco NX-OS initial setup utility because using it can impact the control plane of the network.
- Cisco Nexus 9200 Series switches support CoPP policer rates only in multiples of 10 kbps. If a rate is configured that is not a multiple of 10 kbps, the rate is rounded down. For example, the switch uses 50 kbps if a rate of 55 kbps is configured. (The **show policy-map type control-plane** command shows the user configured rate. See Copp configuration verification commands, on page 632 for more information.)
- For Cisco Nexus 9200 Series switches, ip icmp redirect, IPv6 icmp redirect, ip ICMP unreachable, ipv6 icmp unreachable, and mtu-failure use the same TCAM entry, and they will all be classified to the class map where the first exception is present in the policy. In the CoPP strict profile, they are classified to the class-exception class map. In a different CoPP policy, if the first exception is in a different class map (for example, class-exception-diag), the rest of the exceptions will be classified to the same class map.

Static CoPP ACLs limitations

- These guidelines and limitations apply to static CoPP ACLs:
  - Only Cisco Nexus 9200 Series switches use static CoPP ACLs.
  - Static CoPP ACLs can be remapped to a different CoPP class.
  - Access control entries (ACEs) cannot be modified or removed for static CoPP ACLs.
  - If a CoPP ACL has a static ACL substring, it maps to that type of traffic. For example, if the ACL includes the acl-mac-stp substring, STP traffic classifies to the class map for that ACL.
  - Static CoPP ACLs take priority over dynamic CoPP ACLs, regardless of their position in the CoPP policy, the order in which they are configured, and how they appear in the output of the **show policy-map type control-plane** command.
  - You must have static CoPP ACLs in the CoPP policy. Otherwise, the CoPP policy is rejected.

Protocol ACL filtering feature support and limitations

- Beginning with Cisco Nexus Release 9.2(2), Cisco Nexus 9300-EX, Cisco Nexus 9300-FX Series switches and Cisco Nexus 9500 platform switches support protocol ACL filtering. In this release, IPv6 ACL is not supported.
- Beginning with Cisco NX-OS Release 9.2(3), IPv6 ACL is supported for dynamic CoPP on the Cisco Nexus 9300-EX, Cisco Nexus 9300-FX Series switches, and Cisco Nexus 9500 platform switches.
- The protocol ACL filtering for egress CoPP has the following limitations:
  - Once the egress CoPP ACL is defined, you cannot add or remove an existing rule. This is applicable for all class-maps and policy-maps attached to the egress CoPP ACLs.
  - You cannot override the existing egress CoPP with a new policy. You must remove the existing egress CoPP before you add a new policy.
  - The deny action is not applicable.

- Every entry is programmed in TCAM and uses a different TCAM space if two MAC or IP ACLs with the same entries are created and bound to either the same or a different class-map.
- The maximum TCAM carving supported for the egress CoPP is 128 entries (24 entries are reserved and the remaining 104 entries are for egress CoPP, which are all double wide), which can be any of 52 (Ipv4, mac, Ipv6) entries.
- Policer can be used to drop the traffic completely, with cir and burst as 0.
- SNMP MIB is not supported.

CoPP ACL feature support and limitations

- When a packet meets multiple exception conditions, CoPP matches the packet based on the order in which the CoPP ACLs are configured and matches it only against a single class. This is an expected CoPP behavior.

  Beginning with Cisco NX-OS Release 9.3(4), the UC FIB MISS exception is counted against the CoPP class (copp-system-p-class-exception). Therefore, if a packet has both, the TTL (accounted user class copp-system-p-class-exception-diag) and the UC FIM MISS exceptions, it is accounted against the UC FIB MISS exception. This behavior occurs because the order of the CoPP classes where the copp-system-p-class-exception class has an order higher than the copp-system-p-class-exception-diag class. For NX-OS releases earlier to NX-OS Release 9.3(4), the UC FIB MISS exception was not explicitly handled by the CoPP rules.
- Beginning with Cisco NX-OS Release 10.1(2), CoPP is supported on the Cisco Nexus X9624D-R2 line cards and 9508-FM-R2 switches.
- Beginning with Cisco NX-OS Release 10.1(2), CoPP is supported on the Cisco Nexus 9364D-GX2A and 9332D-GX2B switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on the Cisco Nexus 9332D-H2R switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, CoPP ACL is supported on the Cisco Nexus 93400LD-H1, and 93108TC-FX3 switches.
- Beginning with Cisco Nexus Release 10.4(3)F, CoPP ACL is supported on the Cisco Nexus 9364C-H1 switches.

ISSU limitations

- If you upgrade from a Cisco NX-OS release that supports the CoPP feature to a Cisco NX-OS release that supports the CoPP feature with other classes for new protocols, you must either run the setup utility using the **setup** command or use the **copp profile** command for the new CoPP classes to be available.
- Before you downgrade from a Cisco NX-OS release that supports the CoPP feature to an earlier Cisco NX-OS release that supports the CoPP feature, you should verify compatibility using the **show incompatibility nxos bootflash:** *filename* command. If an incompatibility exists, disable any features that are incompatible with the downgrade image before downgrading the software.
- If there are changes in the CoPP profile in the target image of a non-disruptive upgrade operation, then a reload is required in the target image, post upgrade, for those changes to take effect.

Copp guidelines and limitations for Cisco Nexus 9800 series switches

- Beginning with Cisco NX-OS Release 10.3(1)F , CoPP ACL is supported on Cisco Nexus 9808 switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.
- Cisco Nexus 9808 /9804 switches have the following limitations for SUP CoPP ACL support:
  - Policer rate is in multiples of 161 PPS at Stage-1.
  - There is no shaper in Stage 0.
  - Stage-2 output is at LC/Module level, and Stage-3 output is at SUP/CPU level.
  - Fabrics/FMs are not involved in in-band path.
  - CoPP policy for Stage-1, Stage-2, and Stage-3 are in PPS.
  - CoPP Stage3 stats gets reset to zero after system switchover.
  - Only policer rate changes are supported in Custom CoPP.

CoPP guidelines and limitations for Cisco Nexus 9364E-SG2 switches

- User-defined MAC access lists and source MAC addresses (SMACs) are not supported within Custom CoPP. However, MAC ACLs with well-known destination MAC addresses (DMACs) from the default profile are supported.
- All Bridge Protocol Data Units (BPDUs) are mapped to a single policer; therefore, distinct policing configurations for protocols such as Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), and Spanning Tree Protocol (STP) are not supported.
- CoPP Consistency checker is not supported for Custom CoPP.
- Only a single burst value of 126 packets is supported.
- Destination IP match does not support IPv4 or IPv6 link-local multicast addresses.

Default settings for CoPP

Use this reference to review the default settings for CoPP.

This table lists the default settings for CoPP parameters.

Table 130: Default CoPP parameters settings

Parameters	Default
Default policy	Strict

Parameters	Default
Default policy	9 policy entries **Note** The maximum number of supported policies with associated class maps is 128.
Scale factor value	1.00

Configuring CoPP

This section describes how to configure CoPP.

Configure a control plane class map

You must configure control plane class maps for control plane policies.

You can classify traffic by matching packets based on existing ACLs. The permit and deny ACL keywords are ignored in the matching.

You can configure policies for IP version 4 (IPv4) and IP version 6 (IPv6) packets.

Before you begin

Ensure that you have configured the IP ACLs if you want to use ACE hit counters in the class maps.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify a control plane class map and enters class map configuration mode.

Example:

```
switch(config)#class-map type control-plane ClassMapA
switch(config-cmap)#
```

The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive.

Note
You cannot use class-default, match-all, or match-any as class map names.

Step 3 (Optional) Specify matching for an IP ACL.

Example:

```
switch(config-cmap)#match access-group name MyAccessList
```

Note
The permit and deny ACL keywords are ignored in the CoPP matching.

Step 4 (Optional) Specify matching for IPv4 or IPv6 ICMP redirect exception packets.

Example:

```
switch(config-cmap)#match exception ip icmp redirect
```

Step 5 (Optional) Specify matching for IPv4 or IPv6 ICMP unreachable exception packets.

Example:

```
switch(config-cmap)#match exception ip icmp unreachable
```

Step 6 (Optional) Specify matching for IPv4 or IPv6 option exception packets.

Example:

```
switch(config-cmap)#match exception ip option
```

Step 7 Specify matching for IP Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP) packets.

Example:

```
switch(config-cmap)#match protocol arp
```

Step 8 Complete the remaining control plane class map steps.

a) Exit class map configuration mode.

Example:

```
switch(config-cmap)#exit
switch(config)#
```

b) (Optional) Display the control plane class map configuration.

Example:

```
switch#show class-map type control-plane
```

c) (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Configure a control plane policy map

You must configure a policy map for CoPP, which includes policing parameters. If you do not configure a policer for a class, this default is configured:

- 50 packets per second (pps) with a burst of 32 packets (for Cisco Nexus 9300 and 9500 Series switches)
- 150 kilobits per second (kbps) with a burst of 32,000 bytes (for Cisco Nexus 9200 Series switches)

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Specify a control plane policy map to enter policy map configuration mode.

Example:

```
switch(config)#policy-map type control-plane ClassMapA
switch(config-pmap)#
```

The policy map name can have a maximum of 64 characters and is case sensitive.

Step 3 Specify a control plane class map name or the class-default to enter control plane class configuration mode.

Example:

```
switch(config-pmap)#class ClassMapA
switch(config-pmap-c)#
```

The class-default class map is always at the end of the class map list for a policy map.

Enter one of the following commands:

- **police** [**cir**] {*cir-rate* [*rate-type*]}
- **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]
- **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**]

Example:

```
switch(config-pmap-c)#police cir 52000 bc 1000 packets
```

Example:

```
switch(config-pmap-c)#police cir 3400 kbps bc 200 kbytes
```

Specifies the committed information rate (CIR). The rate range is as follows:

- 0 1 to 268435456 pps (for Cisco Nexus 9300 and 9500 Series switches)
- 0 to 80000000000 bps/gbps/kbps/mbps (for Cisco Nexus 9200 Series switches)

Note
The CIR rate range starts with 0. In previous releases, the CIR rate range starts with 1. A value of 0 drops the packet.

The committed burst (BC) range is as follows:

- 1 to 1073741 packets (for Cisco Nexus 9300 and 9500 Series switches)
- 1 to 512000000 bytes/kbytes/mbytes (for Cisco Nexus 9200 Series switches)

The **conform transmit** action transmits the packet.

Note
You can specify the BC and conform action for the same CIR.

Step 4 (Optional) Specify the threshold value for dropped packets and generates a syslog if the drop count exceeds the configured threshold.

Example:

```
switch(config-pmap-c)#logging drop threshold 100
```

The range for the *drop-count* argument is from 1 to 8000000000 bytes. The range for the *syslog-level* argument is from 1 to 7, and the default level is 4.

Step 5 (Optional) Specify the 802.1Q class of service (CoS) value.

Example:

```
switch(config-pmap-c)#set cos 1
```

The range is from 0 to 7. The default value is 0.

Step 6 Exit the policy map class configuration mode and policy map configuration mode.

Example:

```
switch(config-pmap-c)#exit
switch(config-pmap)#exit
switch(config)#
```

Step 7 (Optional) Display the control plane policy map configuration.

Example:

```
switch#show policy-map type control-plane
```

Step 8 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure a control plane class map, on page 622

Configure the control plane service policy

You can configure one or more policy maps for the CoPP service policy.

Note When you try to change the CoPP policy and apply a custom CoPP policy, it is configured in the hardware as non-atomic, and the following system message appears:

```
This operation can cause disruption of control traffic. Proceed (y/n)?  [no]y
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT24-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT23-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT21-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT25-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT26-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT22-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT4-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
```

Before you begin

Ensure that you have configured a control plane policy map.

Procedure

Step 1 Enters global configuration mode and then, enter control plane configuration mode.

Example:

```
switch#configure terminal
switch(config)#control-plane
switch(config-cp)#
```

Step 2 Specifies a policy map for the input traffic.

Example:

```
switch(config-cp)#service-policy input PolicyMapA
```

Repeat this step if you have more than one policy map.

You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 50 packets per seconds .

Step 3 Exit control plane configuration mode.

Example:

```
switch(config-cp)#exit
switch(config)#
```

Step 4 (Optional) Display the CoPP configuration.

Example:

```
switch#show running-config copp
```

Step 5 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Related Topics

Configure a control plane policy map, on page 623

Configure the CoPP scale factor per line card

You can configure the CoPP scale factor per line card.

The scale factor configuration is used to scale the policer rate of the applied CoPP policy for a particular line card. The accepted value is from 0.10 to 2.00. You can increase or reduce the policer rate for a particular line card without changing the current CoPP policy. The changes are effective immediately, so you do not need to reapply the CoPP policy.

Procedure

Step 1 Enters global configuration mode and then enter the control plane configuration mode.

Example:

```
switch#configure terminal
switch(config)#control-plane
switch(config-cp)#
```

Step 2 Configure the policer rate per line card.

Example:

```
switch(config-cp)#scale-factor 1.10 module 1-2
```

The allowed scale factor value is from 0.10 to 2.00. When the scale factor value is configured, the policing values are multiplied by the corresponding scale factor value of the module, and it is programmed in the particular module.

To revert to the default scale factor value of 1.00, use the **no scale-factor** *value* **module** *multiple-module-range* command, or explicitly set the default scale factor value to 1.00 using the **scale-factor 1 module** *multiple-module-range* command.

Step 3 (Optional) Display the applied scale factor values when a CoPP policy is applied.

Example:

```
switch#show policy-map interface control-plane
```

Step 4 (Optional) Copy the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Change or reapply the default CoPP policy

You can change to a different default CoPP policy, or you can reapply the same default CoPP policy.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Apply the CoPP best practice policy.

Example:

```
switch(config)#copp profile moderate
```

You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 50 packets per seconds .

Step 3 (Optional) Display the CoPP status, including the last configuration operation and its status.

Example:

```
switch#show copp status
```

This command also enables you to verify that the CoPP best practice policy is attached to the control plane.

Step 4 (Optional) Display the CoPP configuration in the running configuration.

Example:

```
switch#show running-config copp
```

Related Topics

Changing or reapplying the default CoPP policy using the setup utility, on page 637

Copying the CoPP best practice policy

The CoPP best practice policy is read-only. If you want to modify its configuration, you must copy it.

Procedure

Step 1 Create a copy of the CoPP best practice policy.

Example:

```
switch#copp copy profile strict prefix abc
```

CoPP renames all class maps and policy maps with the specified prefix or suffix.

Step 2 (Optional) Display the CoPP status, including the last configuration operation and its status.

Example:

```
switch#show copp status
```

This command also enables you to verify that the copied policy is not attached to the control plane.

Step 3 (Optional) Display the CoPP configuration in the running configuration, including the copied policy configuration.

Example:

```
switch#show running-config copp
```

Protocol ACL filtering added for egress CoPP

The protocol ACL filtering for egress CoPP enables the NX-OS swtich to filter all traffic to control plane based on the host MAC, IPv4, and IPv6 address.

Configure ARP ACL filtering for egress CoPP

Before you begin

Ensure that you have configured a control plane policy map.

Procedure

Step 1 Enters global configuration mode and configure the size of the CoPP TCAM region.

Example:

```
switch#hardware access-list tcam region erg-copp 128
```

Step 2 Copy the running configuration to the startup configuration and reload the device.

Example:

```
switch#copy running-config startup-config
switch#reload
```

Note
The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules.

Step 3 Enter the AMC ACL configuration sub-mode.

Example:

```
switch#configure terminal
switch#(config)#mac access-list mac-foo-1
switch(config-mac-acl)#
```

Step 4 Specify a control plane class map and enters class map configuration mode.

Example:

```
switch(config)#class-map type control-plane match-any c-map2
switch(config-cmap)#match access-group name IP-foo-1
```

The default class matching is match-any. The name can be a maximum of 64 characters long and is case-sensitive.

Step 5 Specify a control plane policy map and enters policy map configuration mode.

Example:

```
switch(config)#policy-map type control-plane PolicyMap1
switch(config-pmap)#
```

The policy map name can have a maximum of 64 characters and is case-sensitive.

Step 6 Specify a control plane class map name or the class default and enters control plane class configuration mode.

Example:

```
switch(config-pmap)#class c-map2
switch(config-pmap-c)#
```

The class-default class map is always at the end of the class map list for a policy map.

Enter one of these commands:

- **police** [**cir**] {*cir-rate* [*rate-type*]}
- **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]
- **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**]

Example:

```
switch(config-pmap-c)#police cir 52000 bc 1000 packets
```

Specifies the committed information rate (CIR). The rate range is as follows:

The committed burst (BC) range is as follows:

Step 7 Enter the control plane dynamic configuration mode.

Example:

```
switch(config)#control-plane dynamic
switch(config-cp-dyn)#
```

Step 8 Specify a policy map for the input traffic.

Example:

```
switch(config-cp-dyn)#service-policy-dynamic input PolicyMap1
```

Configure IP ACL filtering for egress CoPP

Before you begin

Ensure that you have configured a control plane policy map.

Procedure

Step 1 Configure the size of the egress CoPP TCAM region.

Example:

```
switch#hardware access-list tcam region erg-copp 128
```

Step 2 Copy the running configuration to the startup configuration and reload the device.

Example:

```
switch#copy running-config startup-config
switch#reload
```

Note
The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules.

Step 3 **ip access-list IP-foo-1**

Example:

```
switch#configure terminal
switch(config)#ip access-list IP-foo-1
switch(config-acl)#
```

Step 4 **permit tcp access-list IP-foo-1** *eq bgp*

Example:

```
switch(config-acl)#10 permit tcp 10.1.1.1/32 10.1.1.2/32 eq bgp
```

Step 5 Specify a control plane class map and enters class map configuration mode.

Example:

```
switch(config)#class-map type control-plane match-any c-map2
switch(config-cmap)#
```

The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive.

Step 6 **match access-group name** *access-list-name*

Example:

```
switch(config-cmap)#match access-group name IP-foo-1
```

Step 7 Specify a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case sensitive.

Example:

```
switch(config)#policy-map type control-plane ClassMapA
switch(config-pmap)#
```

Step 8 Specify a control plane class map name or the class default and enters control plane class configuration mode.

Example:

```
switch(config-pmap)#class ClassMap2
switch(config-pmap-c)#
```

The class-default class map is always at the end of the class map list for a policy map.

Enter one of these commands.

- **police** [**cir**] {*cir-rate* [*rate-type*]}
- **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]
- **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**]

Example:

```
switch(config-pmap-c)#police cir 52000 bc 1000 packets
```

Example:

```
switch(config-pmap-c)#police cir 3400 kbps bc 200 kbytes
```

Specifies the committed information rate (CIR). The rate range is as follows:

The committed burst (BC) range is as follows:

Step 9 Enter the control plane dynamic configuration mode and specify a policy map for the input traffic.

Example:

```
switch(config)#control-plane dynamic
switch(config-cp-dyn)#service-policy-dynamic input PolicyMap1
```

Copp configuration verification commands

Use this reference to verify the Copp configuration.

To display CoPP configuration information, use one of these tasks.

Table 131: Copp configuration verification commands

Command	Purpose
show policy-map type control-plane [**expand**] [**name** *policy-map-name*]	isplays the control plane policy map with associated class maps and CIR and BC values.
show policy-map interface control-plane	Displays the policy values with associated class maps and the number of drops for each policy or class map. When a CoPP policy is applied, the display includes scale factor values. If the scale factor value is set to the default (1.00), it is omitted from the display. **Note** The scale factor modifies the CIR and BC values internally for each module. The display only shows the configured CIR and BC values, while the actual value applied on a module is calculated by multiplying the scale factor with the configured value.

Command	Purpose
show class-map type control-plane [*class-map-name*]	Displays the control plane class map configuration, including the ACLs that are bound to this class map.
show copp diff profile {**strict** | **moderate** | **lenient** | **dense** } [**prior-ver**] **profile** {**strict** | **moderate** | **lenient** | **dense** } **show copp diff profile**	Displays the difference between two CoPP best practice policies. If you do not include the **prior-ver** option, this command displays the difference between two currently applied default CoPP best practice policies, such as the currently applied strict and currently applied moderate policies. When you include the prior-ver option, this command displays the difference between a currently applied default CoPP best practice policy and a previously applied default CoPP best practice policy (such as the currently applied strict and the previously applied lenient policies).
show copp profile {**strict** | **moderate** | **lenient** | **dense** }	Displays the details of the CoPP best practice policy, along with the classes and policer values.
show running-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the running configuration. Using the **all** keyword shows both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show running-config copp [**all**]	Displays the CoPP configuration stored in the running configuration.
show startup-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the startup configuration. Using the **all** keyword shows both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Display the CoPP configuration status

Procedure

Display the configuration status for the CoPP feature.

Example:

```
switch#show copp status
```

Monitor CoPP

Procedure

Display packet-level statistics for all classes that are part of the applied CoPP policy.

Example:

```
switch#show policy-map interface control-plane
```

Statistics are specified in terms of OutPackets (packets admitted to the control plane) and DropPackets (packets dropped because of rate limiting).

Example:

This example shows how to monitor CoPP:

```
switch#show policy-map interface control-plane
Control Plane
Service-policy  input: copp-system-p-policy-strict
class-map copp-system-p-class-critical (match-any)
set cos 7
police cir 19000 pps , bc 128 packets
module 4 :
transmitted 373977 packets;
dropped 0 packets;
```

Monitoring CoPP with SNMP

Beginning with Cisco Nexus Release 9.2(3), CoPP supports the Cisco class-based QoS MIB (cbQoSMIB). You can now monitor all CoPP elements using SNMP, but you cannot modify them. This feature applies only to policies and their subelements—such as classes, match rules, and set actions—attached to the control plane. You cannot view elements of policies that are not in service on the control plane through SNMP.

These cbQoSMIB tables are supported:

- ccbQosServicePolicy
- cbQosInterfacePolicy
- cbQosObjects
- cbQosPolicyMapCfg
- cbQosClassMapCfg
- cbQosMatchStmtCfg
- cbQosPoliceCfg
- cbQosSetCfg

Note SNMP MIB is not supported for Dynamic CoPP.

Clear the CoPP statistics

To clear the CoPP statistics, perform these steps.

Procedure

Step 1 (Optional) Display the currently applied CoPP policy and per-class statistics.

Example:

```
switch#show policy-map interface control-plane
```

Step 2 Clear the CoPP statistics.

Example:

```
switch#clear copp statistics
```

Example:

This example shows how to clear the CoPP statistics for your installation:

```
switch#show policy-map interface control-plane
switch#clear copp statistics
```

Configuration examples for CoPP

This section includes example CoPP configurations.

Copp configuration example

The following example shows how to configure CoPP using IP ACLs and MAC ACLs:

```
configure terminal
ip access-list copp-system-p-acl-igmp
 permit igmp any 10.0.0.0/24

ip access-list copp-system-p-acl-msdp
 permit tcp any any eq 639

mac access-list copp-system-p-acl-arp
 permit any any 0x0806

ip access-list copp-system-p-acl-tacas
 permit udp any any eq 49

ip access-list copp-system-p-acl-ntp
 permit udp any 10.0.1.1/23 eq 123

ip access-list copp-system-p-acl-icmp
 permit icmp any any

class-map type control-plane match-any copp-system-p-class-critical
 match access-group name copp-system-p-acl-igmp
 match access-group name copp-system-p-acl-msdp

class-map type control-plane match-any copp-system-p-class-important
 match access-group name copp-system-p-acl-gre

class-map type control-plane match-any copp-system-p-class-normal
 match access-group name copp-system-p-acl-icmp
 match exception ip icmp redirect
 match exception ip icmp unreachable
 match exception ip option
 match redirect arp-inspect
 match redirect dhcp-snoop

policy-map type control-plane copp-system-p-policy
 class copp-system-p-class-critical
  police cir 2000 kbps bc 1500 bytes pir 3000 kbps be 1500 bytes conform transmit exceed
transmit violate drop
 class copp-system-p-class-important
  police cir 1000 kbps bc 1500 bytes pir 1500 kbps be 1500 bytes conform transmit exceed
transmit violate drop
 class copp-system-p-class-normal
  police cir 400 kbps bc 1500 bytes pir 600 kbps be 1500 bytes conform transmit exceed
transmit violate drop
 class class-default
  police cir 200 kbps bc 1500 bytes pir 300 kbps be 1500 bytes conform transmit exceed
transmit violate drop

control-plane
 service-policy input copp-system-p-policy
```

Create CoPP class and associate ACL:

```
class-map type control-plane copp-arp-class
 match access-group name copp-arp-acl
```

Add the class to the CoPP policy:

```
policy-map type control-plane copp-system-policy
 class copp-arp-class
  police pps 500
```

The following example shows to customize COPP limit:

```
copp copy profile strict suffix CUSTOMIZED-COPP
policy-map type control-plane copp-policy-strict-CUSTOMIZED-COPP
 class copp-class-redirect-CUSTOMIZED-COPP
  police cir 1500 mbps bc 125 mbytes conform transmit violate drop
control-plane
 service-policy input copp-policy-strict-CUSTOMIZED-COPP
```

Changing or reapplying the default CoPP policy using the setup utility

The following example shows how to change or reapply the default CoPP policy using the setup utility.

```
switch# setup
---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes

Do you want to enforce secure password standard (yes/no) [y]: <CR>

Create another login account (yes/no) [n]: n

Configure read-only SNMP community string (yes/no) [n]: n

Configure read-write SNMP community string (yes/no) [n]: n

Enter the switch name : <CR>

Enable license grace period? (yes/no) [n]: n

Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n

Configure the default gateway? (yes/no) [y]: n

Configure advanced IP options? (yes/no) [n]: <CR>

Enable the telnet service? (yes/no) [n]: y

Enable the ssh service? (yes/no) [y]: <CR>

Type of ssh key you would like to generate (dsa/rsa) : <CR>

Configure the ntp server? (yes/no) [n]: n

Configure default interface layer (L3/L2) [L3]: <CR>

Configure default switchport interface state (shut/noshut) [shut]: <CR>

Configure best practices CoPP profile (strict/moderate/lenient/dense/skip) [strict]: strict

Configure CMP processor on current sup (slot 6)? (yes/no) [y]: n
```

```
Configure CMP processor on redundant sup (slot 5)? (yes/no) [y]: n

The following configuration will be applied:
  password strength-check
  no license grace-period
  no telnet server enable
  no system default switchport
  system default switchport shutdown
  policy-map type control-plane copp-system-p-policy

Would you like to edit the configuration? (yes/no) [n]: <CR>

Use this configuration and save it? (yes/no) [y]: y

switch#
```

Changing CoPP policy limit

The following example shows to change CoPP limit to set PTP state stable across PTP interfaces.

```
copp copy profile strict suffix CUSTOMIZED-COPP

policy-map type control-plane copp-policy-strict-CUSTOMIZED-COPP
  class copp-class-redirect-CUSTOMIZED-COPP
    police cir 1500 mbps bc 125 mbytes conform transmit violate drop

control-plane
  service-policy input copp-policy-strict-CUSTOMIZED-COPP
```

Additional references for CoPP

This section offers more information about implementing CoPP.

Related Documents

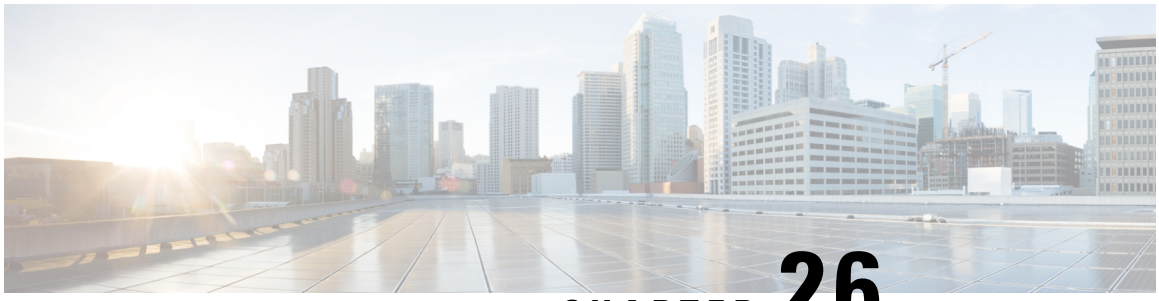
Table 132: Related documents

Related Topic	Document title
Licensing	*Cisco NX-OS Licensing Guide*

Standards

Table 133: Standards

Standards	Title
RFC 2698	A Two Rate Three Color Marker



CHAPTER 26

Configuring Rate Limits

This chapter describes how to configure rate limits for supervisor-bound traffic on Cisco NX-OS devices.

This chapter includes the following sections:

- About rate limits, on page 639
- Guidelines and limitations for rate limits, on page 640
- Default settings for rate limits, on page 641
- Configure rate limits, on page 641
- Monitor rate limits, on page 643
- Clear the rate limit statistics, on page 643
- Rate limit configuration verification commands, on page 644
- Configuration examples for rate limits, on page 644
- Additional references for rate limits, on page 645

About rate limits

Rate limits can prevent redirected packets for exceptions from overwhelming the supervisor module on a Cisco NX-OS device.

You can configure rate limits for the following types of redirected packets:

- Access-list log packets
- Bidirectional Forwarding Detection (BFD) packets
- Catch-all exception traffic
- Fabric Extender (FEX) traffic
- Layer 3 glean packets
- Layer 3 multicast data packets
- SPAN egress traffic

For Cisco Nexus 9200, 9332C, 9364C, 9300-EX, 9300-FX/FXP/FX2/FX3, and 9300-GX platform switches and Cisco Nexus 9500 platform switches with -EX/ FX line cards, the CoPP policer rate is kilo bits per second. For other Cisco Nexus 9000 Series switches, the CoPP policer rate is in packets per second; However, it is kilo bits per second for SPAN egress traffic.

Guidelines and limitations for rate limits

Use this reference to review the guidelines and limitations for rate limits.

Rate limits has the following configuration guidelines and limitations:

- You can set rate limits for supervisor-bound exception and redirected traffic. Use control plane policing (CoPP) for other types of supervisor-bound traffic.

Note Hardware rate-limiters protect the supervisor CPU from excessive inbound traffic. The traffic rate allowed by the hardware rate-limiters is configured globally and applied to each individual I/O module. The resulting allowed rate depends on the number of I/O modules in the system. CoPP provides more granular supervisor CPU protection by utilizing the modular quality-of-service CLI (MQC).

- You can configure a hardware rate-limiter to show statistics for outbound traffic on SPAN egress ports. This rate-limiter is supported on all Cisco Nexus 9000, 9300, and 9500 Series switches .
- The rate-limiter on egress ports is limited per pipe on the Cisco Nexus 9300 and 9500 Series switches . The rate-limiter on egress ports is limited per slice on the Cisco Nexus 9200 and 9300-EX Series switches.
- Cisco Nexus 9300 and 9500 Series switches support both local and ERSPAN. However, the rate-limiter only applies to ERSPAN. You must configure e-racl ACL TCAM region to enable the rate-limiter on these switches. For more information, see the Configuring ACL TCAM Region Sizes section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.
- For Cisco Nexus 9200 and 9300-EX Series switches and the N9K-X9736C-EX, N9K-97160YC-EX, N9K-X9732C-EX, N9K-X9732C-EXM line cards, the SPAN egress rate-limiter applies to both ERSPAN and local SPAN. You do not require special TCAM carving to use the rate-limiter on these devices.
- For Cisco Nexus 92160YC-X, 92304QC, 9272Q, 9232C, 92300YC, 9348GC-FXP, 93108TC-FX, 93180YC-FX Series switches , you should not configure both, sFlow and ERSPAN.
- Logging rate-limit is enabled by default. No default configuration is shown up in **show running-config** and in **show running-config all** . Use **show logging** cli to check if rate-limit is enabled. It has a dedicated field to verify if rate-limit is enabled or disabled.

  Once no logging rate-limit config is applied, it appears in the running-config and displayed in show logging output.
- The **rate-limit cpu direction** {**input** | **output** | **both**} **pps** *packets* **action log** command is not supported on Cisco Nexus 9000 Series switches.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default settings for rate limits

Use this reference to review the default settings for rate limits.

The following table displays default settings for rate limits.

Table 134: Default rate limits parameters settings

Parameters	Default
Access-list log packets rate limit	100 packets per second
BFD packets rate limit	10000 packets per second
Exception packets rate limit	50 packets per second
FEX packets rate limit	12000 packets per second
Layer 3 glean packets rate limit	100 packets per second
Layer 3 multicast data packets rate limit	3000 packets per second
SPAN egress rate limit	No limit
R-L Class packets rate limit	100 packets per second
SPAN packets rate limit	50 packets per second
sFLOW packets rate limit	40000 kilobits per second
VXLAN-OAM packets rate limit	1000 packets per second
100M-ethports packets rate limit	10000 packets per second
SPAN egress disabled dot1x packets rate limit	3000 packets per second
MPLS-OAM packets rate limit	300 packets per second
Netflow packets rate limit	120000 packets per second
SSX packets rate limit	120000 packets per second
UCS-mgmt packets rate limit	120000 packets per second
MDNS packets rate limit	1024 packets per second

Configure rate limits

You can set rate limits on supervisor-bound traffic.

Procedure

Step 1 Use the **hardware rate-limiter access-list-log** {*packets* | **disable**} [**module** *module* [**port** *start end*]] to configure rate limits for packets that are copied to the supervisor module for access list logging. The range is 0–10000.

Example:

```
switch#configure terminal
switch(config)# hardware rate-limiter access-list-log 200
```

Step 2 Use the **hardware rate-limiter bfd** *packets* [**module** *module* [**port** *start end*]] command to configure rate limits for bidirectional forwarding detection (BFD) packets. The range is 0–10000.

Example:

```
switch(config)#hardware rate-limiter bfd 500
```

Step 3 Use the **hardware rate-limiter exception** *packets* [**module** *module* [**port** *start end*]] command to configure rate limits for any exception traffic in the system that is not classified by the Control Plane Policing (CoPP) policy. The range is 0–10000.

Example:

```
switch(config)#hardware rate-limiter exception 500
```

Step 4 use the **hardware rate-limiter fex** *packets* [**module** *module* [**port** *start end*]] command to configure rate limits for supervisor-bound FEX traffic. The range is 0–10000.

Example:

```
switch(config)#hardware rate-limiter fex 500
```

Step 5 Use the **hardware rate-limiter layer-3 glean** *packets* [**module** *module* [**port** *start end*]] command to configure rate limits for Layer 3 glean packets. The range is 0–10000.

Example:

```
switch(config)#hardware rate-limiter layer-3 glean 500
```

A node receiving traffic for a particular destination might be unable to forward traffic because it is unaware of the rewrite information or the physical layer interface behind which the destination resides. During this time, it is possible to install a glean entry in the data path for that destination. Because this might not be a pointer to the global punt adjacency, a reserved module or port value is used to punt such packets to the supervisor. This glean rate can be controlled using the given rate limiter.

Note
The CoPP policy controls the rate of glean packets that are forwarded to CPU due to hit of global punt adjacency. The Layer 3 glean hardware rate-limiter limits the number of glean packets that are redirected to CPU by sup-redirect acess-list. This is used in special cases such as, in the VXLAN environment when the packet is received from an unknown VTEP.

Step 6 Use the **hardware rate-limiter layer-3 multicast local-groups** *packets* [**module** *module* [**port** *start end*]] command to configure rate limits for Layer 3 multicast data packets that are punted for initiating a shortest-path tree (SPT) join. The range is 0–10000.

Example:

```
switch(config)#hardware rate-limiter layer-3 multicast local-groups 300
```

Step 7 Use the **hardware rate-limiter span-egress** *rate* [**module** *module*] command to configure rate limits for SPAN for egress traffic. The range is 0–100000000.

Example:

```
switch(config)#hardware rate-limiter span-egress 123
```

Note
You should not configure both sFlow and the SPAN egress rate-limiter.

Step 8 (Optional) Use the **show hardware rate-limiter** [**access-list-log | bfd |exception | fex | layer-3 glean | layer-3 multicast local-groups |**] | **module** *module*] command to display the rate limit configuration. The module range is 1–30.

Example:

```
switch#show hardware rate-limiter
```

Step 9 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration. the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Monitor rate limits

You can monitor rate limits.

Procedure

Use the **show hardware rate-limiter** [**access-list-log** | **bfd** | **exception** | **fex** | **layer-3 glean** | **layer-3 multicast local-groups** | **span-egress** | **module** *module*] command to display the rate limit statistics.

Example:

```
switch#show hardware rate-limiter access-list-log
```

Clear the rate limit statistics

You can clear the rate limit statistics.

Procedure

Use the **clear hardware rate-limiter** {**all** | **access-list-log** | **bfd** | **exception** | **fex** | **layer-3 glean** | **layer-3 multicast local-groups** | **span-egress** [**module** *module*]} command to clear the rate limit statistics.

Example:

```
switch#clear hardware rate-limiter access-list-log
```

Rate limit configuration verification commands

Use this reference to verify the Rate limit configuration.

To display the rate limit configuration information, perform the following tasks:

Table 135: Rate limit configuration verification commands

Command	Purpose
show hardware rate-limiter [**access-list-log** | **bfd** | **exception** | **fex** | **layer-3 glean** | **layer-3 multicast local-groups** | **span-egress** | **module** *module*]	Displays the rate limit configuration.

Configuration examples for rate limits

The following example shows how to configure rate limits for packets copied to the supervisor module for access list logging:

```
switch#configure terminal
switch(config)#hardware rate-limiter access-list-log 200
switch(config)#exit
switch#show hardware rate-limiter access-list-log
Units for Config: kilo bits per second
Allowed, Dropped & Total: aggregated since last clear counters
Module: 4
R-L Class           Config           Allowed         Dropped             Total
+------------------+--------+---------------+---------------+-----------------
+
access-list-log         100               0               0                 0
Port group with configuration same as default configuration
Eth4/1-36
Module: 22
R-L Class           Config           Allowed         Dropped             Total
+------------------+--------+---------------+---------------+-----------------
+
access-list-log         100               0               0                 0
Port group with configuration same as default configuration
Eth22/1-0
```

The following example shows how the SPAN egress rate limiter might be in conflict with sFlow:

```
switch#configure terminal
switch(config)#hardware rate-limiter span-egress 123
Warning: This span-egress rate-limiter might affect functionality of sFlow
switch(config)#exit
switch#show hardware rate-limiter span-egress
Units for Config: kilo bits per second
Allowed, Dropped & Total: aggregated since Module: 1
R-L Class          Config            Allowed        Dropped           Total
+----------------+----------+--------------+--------------+----------------+
```

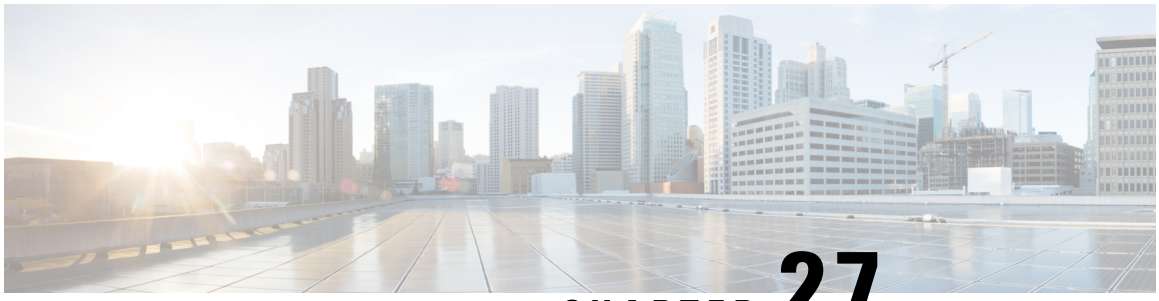
```
L3 glean                100             0               0               0
L3 mcast loc-grp       3000             0               0               0
access-list-log         100             0               0               0
bfd                   10000             0               0               0
exception                50             0               0               0
fex                    3000             0               0               0
span                     50             0               0               0
dpss                   6400             0               0               0
span-egress             123             0               0               0
<<configured
```

Additional references for rate limits

This section includes additional information related to implementing rate limits.

Table 136: Related documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*



CHAPTER 27

Configure MACsec

This chapter describes how to configure MACsec on Cisco NX-OS devices.

- About MACsec, on page 647
- Licensing requirements for MACsec, on page 648
- Guidelines and limitations for MACsec, on page 648
- Enable MACsec, on page 655
- Disable MACsec, on page 656
- Configure a MACsec keychain and keys, on page 657
- MACsec packet-number exhaustion, on page 659
- Configure MACsec fallback key, on page 659
- Configure a MACsec policy, on page 660
- Configure MACsec EAP, on page 662
- QKD integration with SKIP on MACsec, on page 663
- About Configurable EAPOL Destination and Ethernet Type, on page 671
- MACsec configuration verification commands, on page 673
- Display MACsec Statistics, on page 676
- Configuration Example for MACsec, on page 679
- XML output for MACsec show commands, on page 683
- Macsec supported MIBs and download locations, on page 690
- Related documentation, on page 690

About MACsec

Media Access Control Security (MACsec) an IEEE 802.1AE along with MACsec Key Agreement (MKA) protocol provide secure communications on Ethernet links. It offers the following :

- Provides line rate encryption capabilities.
- Helps to ensure data confidentiality by providing strong encryption at Layer 2.
- Provides integrity checking to help ensure that data cannot be modified in transit.
- Can be selectively enabled using a centralized policy to help ensure that it is enforced where required while allowing non-MACsec-capable components to access the network.

- Encrypts packets on a hop-by-hop basis at Layer 2, allowing the network to inspect, monitor, mark, and forward traffic according to your existing policies (unlike end-to-end Layer 3 encryption techniques that hide the contents of packets from the network devices they cross).

Key Lifetime and Hitless Key Rollover

A MACsec keychain can have multiple pre-shared keys (PSKs), each configured with a key ID and an optional lifetime. A key lifetime specifies at which time the key activates and expires. In the absence of a lifetime configuration, the default lifetime is unlimited. When a lifetime is configured, MKA rolls over to the next configured pre-shared key in the keychain after the lifetime is expired. The time zone of the key can be local or UTC. The default time zone is UTC.

To configure a MACsec keychain, see Configure a MACsec keychain and keys, on page 657.

A key can roll over to a second key within the same keychain by configuring the second key (in the keychain) and configuring a lifetime for the first key. When the lifetime of the first key expires, it automatically rolls over to the next key in the list. If the same key is configured on both sides of the link at the same time, then the key rollover is hitless (that is, the key rolls over without traffic interruption).

Fallback Key

A MACsec session can fail due to a key/key name (CKN) mismatch or a finite key duration between the switch and a peer. If a MACsec session does fail, a fallback session can take over if a fallback key is configured. A fallback session prevents downtime due to primary session failure and allows a user time to fix the key issue causing the failure. A fallback key also provides a backup session if the primary session fails to start. This feature is optional.

To configure a MACsec fallback key, see Configure MACsec fallback key, on page 659.

Licensing requirements for MACsec

Use this reference to review licensing requirements for MACsec.

Table 137: Licensing requirements for MACsec

Product	License Requirement
Cisco NX-OS	MACsec requires a Security license. Consumption of the security license by MACsec requires a macsec applied to at least one interface. For a complete explanation of the Cisco NX-OS licensing scheme and obtain and apply licenses, see the Cisco NX-OS Licensing Guide .

Guidelines and limitations for MACsec

Review MACsec requirements, supported platforms, and operational limitations.

Interface and protocol behavior

- MACsec is supported on the following interface types:

- Layer 2 switch ports (access and trunk)
- Layer 3 routed interfaces (no subinterfaces)

Note Enabling MACsec on the Layer 3 routed interface also enables encryption on all the subinterfaces that are defined under that interface. However, selectively enabling MACsec on a subset of subinterfaces of the same Layer 3 routed interface is not supported.

- Layer 2 and Layer 3-port channels (no subinterfaces)

- Beginning with Cisco Nexus Release 10.2(1)F, Secure Channel Identifier (SCI) can be disabled from MACSec security tag (SecTAG) on Cisco Nexus 9000 ToR switches.
  - It is supported in FX2 and FX3 platforms.
  - It is supported in FX platforms with XPN cipher suites only
- When the Cisco Nexus ToR switches are downgraded from Cisco NX-OS Release 9.3.7 to Cisco NX-OS Release 9.3.6 and below releases, MACsec is not supported.
- MKA is the only supported key exchange protocol for MACsec. The Security Association Protocol (SAP) is not supported.
- Link-level flow control (LLFC) and priority flow control (PFC) are not supported with MACsec.
- Multiple MACsec peers (different SCI values) for the same interface are not supported.
- You can retain the MACsec configuration when you disable MACsec using the **macsec shutdown** command.

Cloud Scale switch support by release

Table 138: MACsec support by Cloud Scale switch and release

Minimum release	Switch or line card	MACsec support
9.2(1)	Cisco Nexus 93180YC-FX	MACsec is supported.
9.3(1)	Cisco Nexus 9364C	Ports 49–64 support MACsec. Configuring or removing MACsec causes a port flap regardless of the security-policy type.
9.3(1)	Cisco Nexus 9332C	Ports 25–32 support MACsec. Configuring or removing MACsec causes a port flap regardless of the security-policy type.
9.3(1)	Cisco Nexus 9348GC-FXP	Ports 49–54 support MACsec.
9.3(3)	Cisco Nexus 93216TC-FX2 and Cisco Nexus 93360YC-FX2	MACsec is supported.
9.3(5)	Cisco Nexus 93180YC-FX3S	All ports support MACsec.

Minimum release	Switch or line card	MACsec support
9.3(5)	Cisco Nexus X9732C-FX and X9788TC-FX line cards	MACsec is supported.
9.3(7)	Cisco Nexus 9364C and 9336C-FX2	QSA: MACsec is supported.
10.1(1)	Cisco Nexus 9336C-FX2 and 9336C-FX2-E, and 9364C	QSA: MACsec is supported.
10.1(1)	Cisco Nexus 9336C-FX2-E	MACsec is supported.
10.1(2)	Cisco Nexus 9300-FX3 platform switches	QSA: MACsec is supported.
10.2(1)F	Cisco Nexus X9716D-GX	MACsec is supported.
10.2(1q)F	Cisco Nexus 9332D-GX2B	Ports 25–32 support MACsec.
10.2(2)F	Cisco Nexus N9K-C9348D-GX2A	Ports 1–48 support MACsec.
10.2(2)F	Cisco Nexus X9736C-FX and X9736Q-FX line cards	10G QSA: MACsec is supported.
10.2(2)F	Cisco Nexus 9364D-GX2A	Ports 1–16 support MACsec.
10.3(1)F	N9K-C93180YC-FX3H	MACsec is supported on all 24 ports.
10.3(2)F	Cisco Nexus 9408 with X9400-16W and X9400-8D LEM modules	All supported links support MACsec.
10.4(1)F	Cisco Nexus 9348GC-FX3 and 9348GC-FX3PH	Ports 49–54 support MACsec.
10.4(1)F	Cisco Nexus 9332D-H2R	Front-panel ports 1–32 support MACsec. Ethernet1/33 and Ethernet1/34 do not support MACsec.
10.4(2)F	Cisco Nexus 93400LD-H1	All ports support MACsec.
10.4(2)F	Cisco Nexus 93108TC-FX3	Ports 49–54 support MACsec.
10.4(3)F	Cisco Nexus 9364C-H1	Ports 49–64 support MACsec.

Silicon One switch support and limitations

When you configure MACsec on Silicon One switches, review these guidelines and limitations.

Table 139: MACsec support by Silicon One switch and release

Switch family	Switch	Minimum release	MACsec support
Nexus 9300-SE1	N9396Y12C-SE1 N9396T12C-SE1	10.6(2)F	All 96 10G/25G SFP28 ports and 12 40G/100G QSFP28 ports support MACsec.

Port-speed support and limitations

- On Cisco Nexus 9300-FX switches and 9700-FX line cards:
  - MACsec is not supported on 1G ports.
  - Additionally, if a MAC block contains any 1G ports, MACsec is not supported on any port within that MAC block.
  - When using 1G optics on MACsec capable 9700-FX line cards, it is recommended to change diagnostics mode to 'minimal'. For more information on diagnostics, see the Bootup Diagnostics section in the **Cisco Nexus 9000 Series NX-OS System Management Configuration Guide**.
- On Cisco Nexus 9300-FX2 switches:
  - MACsec is supported on 1G ports, provided the ports are non-BV or non-retimer ports.

    For details about retimer ports, refer to the *Supported Retimer Ports* section in the **Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide**.
- Beginning with Cisco NX-OS Release 10.1(1) , the Cisco Nexus 93180YC-FX3, and 93108TC-FX3P switches support MACsec on all port speeds including 1G and 10G port speeds.
- MACsec is supported on Cisco Nexus 93240YC-FX2, 9336C-FX2, 93108TC-FX, 93180YC-FX switches and the X9736C-FX, and X9732C-EXM line cards.
-

MACsec and high-power optics limitations

On some Cisco Nexus 9000 platforms, MACsec and High-Power Optics (HPO) are subject to platform-specific power and credit limits. MACsec and HPO resources cannot borrow credits from each other after either resource type is in use. If MACsec is already in use, HPO cannot borrow MACsec credits. If HPO is already in use, MACsec cannot borrow HPO credits.

Table 140: Supported MACsec and high-power optics combinations

Platform	Supported MACsec and HPO combinations	Limit/error behavior
N9K-C9364D-GX2A	Up to 16 MACsec ports with no HPO; up to 8 MACsec ports with up to 8 HPO ports; up to 13 HPO ports with no MACsec. HPO is supported on odd-numbered 400G ports.	Additional MACsec or HPO is blocked with `macSecPowerInsufficient` or `hpoPowerInsufficient`. Unsupported HPO ports or excess HPO usage can cause `hpoXcvrUnsupported` and err-disable the port.
N9K-C9332D-GX2B	Up to 8 MACsec ports with no HPO; up to 4 MACsec ports with up to 4 HPO ports; up to 8 HPO ports with no MACsec.	Same limit behavior.
N9K-C9348D-GX2A	48 MACsec ports and up to 24 HPO ports are supported together. HPO is supported only on the listed supported ports.	The 25th HPO port or HPO on an unsupported port is not supported and can trigger `hpoXcvrUnsupported`.
N9K-C9408	64 MACsec ports and up to 32 HPO ports are supported together.	The 33rd HPO port is not supported.

For more information about port and HPO details, see Hardware Installation Guide.

When the platform limit is reached, the CLI displays one of these error messages to indicate which resource limit or port restriction caused the configuration to fail: `macSecPowerInsufficient`, `hpoPowerInsufficient`, or `hpoXcvrUnsupported`.

Session and policy behavior

- MACsec sessions are liberal in accepting packets from a key server whose latest Rx and latest Tx flags have been retired after Tx SA installation for the first time. The MACsec session then converges into a secure state.
- Beginning with Cisco NX-OS Release 9.2(1), the following configurations are allowed:
  - Allowing MACSec policy to be modified while the policy is referenced by an interface.
  - Allowing different MACsec policies across different lanes of a breakout port.
- Beginning with Cisco Nexus Release 9.3(1), you cannot apply MACsec configuration directly on port-channel interface. However, you can apply MACsec configurations directly on port-channel member ports. This applies to both NX-OS and vPC port-channels.

MACsec policy limitations

- BPDU packets can be transmitted before a MACsec session becomes secure.

Platform and cipher-suite limitations

- On the Cisco Nexus 9332D-GX2B , 9364D-GX2A and 9348D-GX2A switches and Cisco Nexus X9836DM-A line card, when MACsec is either configured or unconfigured on a port, a port-flap occurs irrespective of MACsec security-policy type.
- Beginning with Cisco Nexus Release 10.3(3)F, the cipher key enforcement feature provides the option to define the supported cipher suites from the most preferred to the least preferred on the Cisco Nexus 9332D-GX2B, 9336C-FX2, 93180YC-FX, and 93180YC-FX3 switches with following limitations:
  - Cipher Key Enforcement feature will work effectively only if it is prioritized as key server, else it will be in **init** or **pending** state of the session.
  - Cipher Key Enforcement feature is only supported for direct connections between 2 peers. If MKA session is with multiple peers, this feature is not expected to work properly.
  - During a peer cipher suite allowance change, session may not secure on the most preferred supported cipher suite.
  - When changing cipher from any to an enforced-peer cipher on a policy that is used on any secured MACsec session, it is recommended to flap the port after changing the cipher to reach expected behavior. If flapping is not done, the session shows secured on the switch while peer session shows pending on unsupported ciphers. It could also cause session to not get secured immediately even if supported cipher is present in enforce-peer cipher suites.
  - The Allowed Peer Cipher Suites (APSC) can not be empty or cannot have duplicates.
  - Cipher-suite and cipher-suite enforce-peer commands cannot coexist under the same policy.

- While waiting for SAK Cipher-Enforcing Timer to timeout to try the next cipher suite in line, the data and control traffic might face one way traffic interruptions even with should secure mode. The interruption will only recover when session is secured.

Keychain limitations

- You cannot overwrite the octet string for a MACsec key. Instead, you must create a new key or a new keychain.
- A new key in the keychain is configured when you enter *end* or *exit* . The default timeout for editor mode is 6 seconds. If the key is not configured with the key octet string or/and the send lifetime within the 6-second window, incomplete information may be used to bring up the MACsec session and could result in the session being stuck in an Authorization Pending state. If the MACsec sessions are not converged after the configuration is complete, you might be advised to shut/no shut the ports.
- For a given keychain, key activation times should overlap to avoid any period of time when no key is activated. If a time period occurs during which no key is activated, session negotiation fails and traffic drops can occur. The key with the latest start time among the currently active keys takes precedence for a MACsec key rollover.
- When the command `key-chain macsec-psk no-show` is configured, the key-octet-string is masked with asterisks in the running configuration and in saved configuration files. Consequently, if you perform a reload ASCII with this command enabled, the key-octet-string configuration will not be retained. To avoid loss of the key configuration, remove the command `key-chain macsec-psk no-show` before performing an ASCII reload or ISSU/ISSD, and reconfigure it afterward. This limitation may also apply to ISSU/ISSD and configure replace.

Fallback limitations

- If a MACsec session is secured on an old primary key, it does not go to a fallback session in case of mismatched latest active primary key. So the session remains secured on the old primary key and will show as rekeying on the old CA under status. And the MACsec session on the new key on primary PSK will be in init state.
- Use only one key with infinite lifetime in the fallback key chain. Multiple keys are not supported.
- The key ID (CKN) used in the fallback key chain must not match any of the key IDs (CKNs) used in the primary key chain.
- Once configured, fallback configuration on an interface cannot be removed, unless the complete MACsec configuration on the interface is removed.

Layer 2 tunneling limitations

- MACsec is not supported on ports configured for dot1q tunneling (switchport mode dot1q-tunnel) or L2TP.
- L2TP does not work if STP is enabled on trunk ports for non-native VLANs.

EAPOL configuration limitations

- In Cisco NX-OS Release 9.3(1), EAPOL configuration is not supported on Cisco Nexus 9332C and 9364C Series switches.
- Within the same slice of the forwarding engine, EAPOL ethertype and dot1q ethertype cannot have the same value.
- For enabling EAPOL configuration, the range of ethernet type between 0 to 0x599 is invalid.
- For enabling EAPOL configuration, on N9K-X9836DM-A line card, the only supported EAPOL mac addresses are range 0x0180c2000000 to 0x0180c20000ff.
- While configuring EAPOL packets, the following combinations must not be used:
  - Mac address 0100.0ccd.cdd0 with any ethertype
  - Any mac address with Ether types: 0xfff0, 0x800, 0x86dd
  - The default destination MAC address, 0180.c200.0003 with the default Ethernet type, 0x888e
  - Different EAPOL DMAC addresses on both MACsec peers. The MACsec session works only if the MACsec peer is sending MKAPDUs with the DMAC configured locally.
- Beginning with Cisco NX-OS Release 10.2(1)F, EAPOL is supported on Cisco Nexus 9300-FX3 Series switches.

Interoperability limitations

- Interoperability of N9K-X9732C-EXM and other peer switches (other Cisco and non-Cisco switches) is supported only with the XPN cipher suite.
- MACsec peers must run the same Cisco NX-OS release in order to use the AES_128_CMAC cryptographic algorithm. For interoperability between previous releases and Cisco NX-OS Release 9.2(1), you must use keys with the AES_256_CMAC cryptographic algorithm.
- For interoperability between previous releases and Cisco NX-OS Release 9.2(1), pad the MACsec key with zeros if it is less than 32 octets.
- On any Cisco NX-OS switch, you can configure only one unique combination of an alternate MAC address and Ethernet type on all interfaces.
- When you attempt to downgrade from Cisco NX-OS Release 9.3(1) to a Cisco NX-OS release without per port channel member MACsec configuration support, when the switch has MACsec configurations on members of the same port channel interface that are different from each other, you may see the following error message:

```
Asymmetric macsec config is present on port-channel members. Please use
symmetric macsec config across members to perform Non-disruptive ISSU.
```

- Software support for MACsec is not available on the Cisco Nexus X9400-22L LEM card.

Statistics limitations

- Few CRC errors should occur during the transition between MACsec and non-MACsec mode (regular port shut/no shut).

- Secy statistics are cumulative and polled every 30 seconds.
- The IEEE8021-SECY-MIB OIDs secyRxSAStatsOKPkts, secyTxSAStatsProtectedPkts, and secyTxSAStatsEncryptedPkts can carry only up to 32 bits of counter values, but the traffic may exceed 32 bits.
- On Cisco Nexus 9300-FX3 platform switches, the **show macsec secy statistics** command supports rate statistics and the following rate related "CISCO-SECY-EXT-MIB" OIDs beginning with Cisco NX-OS Release 10.4(2)F.
  - cseSecyIfRxUncontrolledPktRate,
  - cseSecyIfRxControlledPktRate,
  - cseSecyIfTxUncontrolledPktRate,
  - cseSecyIfTxControlledPktRate
  - cseSecyIfRxControlledOctetRate
  - cseSecyIfTxControlledOctetRate
  - cseSecyIfRxUnControlledOctetRate
  - cseSecyIfTxUnControlledOctetRate

Nondisruptive ISSU and chassis limitations

For the MACsec policy commands used for nondisruptive ISSU, see the *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide*.

- MACsec is not supported on Cisco Nexus 9500 Series switches with Nexus 9600-R/R2 line cards.

Enable MACsec

Before you can access the MACsec and MKA commands, you must enable the MACsec feature.

Procedure

Step 1 **configure terminal**

Enters global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 **feature macsec**

Enables MACsec and MKA on the device.

Example:

```
switch(config)#feature macsec
```

Step 3 (Optional) **copy running-config startup-config**

Copies the running configuration to the startup configuration.

Example:

```
switch#copy running-config startup-config
```

Disable MACsec

Beginning with Cisco NX-OS Release 9.2(1), disabling the MACsec feature only deactivates this feature and does not remove the associated MACsec configurations.

Disabling MACsec has the following conditions:

- MACsec shutdown is global command and is not available at the interface level.
- The macsec shutdown, show macsec mka session/summary, show macsec mka session detail, and show macsec mka/secy statistics commands will display the 'Macsec is shutdown' message. However, the show macsec policy and show key chain commands will display the output.
- Consecutive MACsec status changes from macsec shutdown to no macsec shutdown and vice versa needs a 30 seconds time interval in between the status change.

Procedure

Step 1 **configure terminal**

Enters global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 **macsec shutdown**

Disables the MACsec configuration on the device. The **no** option restores the MACsec feature.

Example:

```
switch(config)#macsec shutdown
```

Step 3 (Optional) **copy running-config startup-config**

Copies the running configuration to the startup configuration. This step is required only if you want to retain the MACsec in the shutdown state after the switch reload.

Example:

```
switch#copy running-config startup-config
```

Configure a MACsec keychain and keys

You can create a MACsec keychain and keys on the device.

Note Only MACsec keychains will result in converged MKA sessions.

Before you begin

Make sure that MACsec is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 (Optional) [**no**] **key-chain macsec-psk no-show**

Example:

```
switch(config)#key-chain macsec-psk no-show
```

Hides the encrypted key octet string in the output of the **show running-config** and **show startup-config** commands by replacing the string with a wildcard character. By default, PSK keys are displayed in encrypted format and can be easily decrypted. This command applies only to MACsec keychains.

Note
The key-octet-string is hidden (with asterisks) when you save the configuration to a file. As a result, if you perform an ASCII reload or config replace with this command configured, the 'key-octet-string' configuration will not be retained.

Step 3 **key chain** *name* **macsec**

Example:

```
switch(config)#key chain 1 macsec
switch(config-macseckeychain)#
```

Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.

Step 4 **key** *key-id*

Example:

```
switch(config-macseckeychain)#key 1000
switch(config-macseckeychain-macseckey)#
```

Creates a MACsec key and enters MACsec key configuration mode. The range is from 1 to 32 octets, and the maximum size is 64.

Note
The key must consist of an even number of characters.

Step 5 **key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** | **AES_256_CMAC**}

Example:

```
switch(config-macseckeychain-macseckey)#key-octet-string
abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789
cryptographic-algorithm AES_256_CMAC
```

Configures the octet string for the key. The *octet-string* argument can contain up to 64 hexadecimal characters. The octet key is encoded internally, so the key in clear text does not appear in the output of the **show running-config macsec** command.

The key octet string includes the following:

- 0 Encryption Type - No encryption (default)
- 6 Encryption Type - Proprietary (Type-6 encrypted). For more information, see Enable Type-6 encryption on MACsec keys, on page 552 .
- 7 Encryption Type - Proprietary WORD key octet string with maximum 64 characters

Note
MACsec peers must run the same Cisco NX-OS release in order to use the AES_128_CMAC cryptographic algorithm. To interoperate between previous releases and Cisco NX-OS Release 7.0(3)I7(2) or a later release, you must use keys with the AES_256_CMAC cryptographic algorithm.

Step 6 **send-lifetime** *start-time* **duration** *duration*

Example:

```
switch(config-macseckeychain-macseckey)#send-lifetime 00:00:00 Oct 04 2016 duration 100000
```

Configures a send lifetime for the key. By default, the device treats the start time as UTC.

The *start-time* argument is the time of day and date that the key becomes active. The *duration* argument is the length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).

Step 7 (Optional) **show key chain** *name*

Example:

```
switch#show key chain 1
```

Displays the keychain configuration.

Step 8 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

MACsec packet-number exhaustion

Every MACsec frame contains a 32-bit packet number (PN), and it is unique for a given Security Association Key (SAK). Upon PN exhaustion (after reaching 75% of 2^{32}- 1), SAK rekey takes place automatically to refresh the data plane keys and the PN will wrap around.

For example, on 10G full line rate at 64 bytes, the SAK rekey will occur every 216 seconds due to PN exhaustion.

This is applicable when using GCM-AES-PN-128 or GCM-AES-PN-256 cipher-suites.

When GCM-AES-XPN-128 or GCM-AES-XPN-256 cipher-suite is used, the SAK rekey happens automatically when reaching 75% of 2^{64}- 1, which will take several years to exhaust the packet numbering. The cipher-suite is configurable under the MACsec policy and the operational cipher-suite is determined by the key-server device.

It is recommended to use the XPN ciphersuite on the N9K-X9732C-EXM line card.

Configure MACsec fallback key

Beginning with Cisco NX-OS Release 9.2(1), you can configure a fallback key on the device to initiate a backup session if the primary session fails as a result of a key/key name (CKN) mismatch or a finite key duration between the switch and peer.

Before you begin

Make sure that MACsec is enabled and a primary and fallback keychain and key ID are configured. See Configuring a MACsec Keychain and Keys.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters the global configuration mode.

Step 2 **interface** *name*

Example:

```
switch(config)#interface ethernet 1/1
switch(config-if)#
```

Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port.

Step 3 **macsec keychain** *keychain-name* **policy** *policy-name* **fallback-keychain** *keychain-name*

Example:

```
switch(config-if)#macsec keychain kc2 policy abc fallback-keychain fb_kc2
```

Specifies the fallback keychain to use after a MACsec session failure due to a key/key ID mismatch or a key expiration.

Note
The fallback key ID should not match any key ID from a primary keychain.

Fallback keychain configuration for each interface can be changed on the corresponding interface, without removing the MACsec configuration, by reissuing the same command with the fallback keychain name changed.

Note
The command must be entered exactly the same as the existing configuration command for the interface, except for the fallback keychain name.

See Configuring a MACsec Keychain and Keys.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure a MACsec policy

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Make sure that MACsec is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **macsec policy** *name*

Example:

```
switch(config)#macsec policy abc
switch(config-macsec-policy)#
```

Creates a MACsec policy.

Step 3 (Optional) **[no] cipher-suite** {{**enforce-peer** *allowed-peer-cipher-suite1* [*allowed-peer-cipher-suite2* [*allowed-peer-cipher-suite3* [*allowed-peer-cipher-suite4*]]]} | *suite*}

Example:

```
switch(config-macsec-policy)#cipher-suite enforce-peer GCM-AES-XPN-256 GCM-AES-XPN-128
```

Configures the sequence for the following cipher suites from the most preferred to the least preferred. The session gets secured on the most preferred cipher suite that is supported by the peer.: GCM-AES-128, GCM-AES-256, GCM-AES-XPN-128, or GCM-AES-XPN-256.

To unconfigure, you can either use the **no** form or overwrite the existing sequence with the required sequence preference.

Note

- For this feature to work, ensure that the Cisco NX-OS switch is set as key server.
- If the peer supports cipher suites that are not included in the set of cipher suites defined in the **cipher-suite enforce-peer** command, the MKA session state will not be secured instead it will be pending.

Step 4 (Optional) **[no] include-sci**

Example:

```
switch(config-macsec-policy)#no include-sci
```

Disables SCI in SecTAG. By default, SCI is always enabled.

Note
To prevent packet drops, ensure that SCI tagging settings are consistent at both ingress and egress points.

Step 5 (Optional) **key-server-priority** *number*

Example:

```
switch(config-macsec-policy)#key-server-priority 0
```

Configures the key server priority to break the tie between peers during a key exchange. The range is from 0 (highest) and 255 (lowest), and the default value is 16.

Step 6 (Optional) **security-policy** *name*

Example:

```
switch(config-macsec-policy)#security-policy should-secure
```

Configures one of the following security policies to define the handling of data and control packets:

- **must-secure** —Packets not carrying MACsec headers will be dropped.
- **should-secure** —Packets not carrying MACsec headers will be permitted. This is the default value.

Step 7 Complete the remaining required MACsec policy steps.

a) (Optional) **window-size** *number*

Example:

```
switch(config-macsec-policy)#window-size 512
```

Configures the replay protection window such that the secured interface will not accept any packet that is less than the configured window size. The range is from 0 to 596000000.

b) (Optional) **sak-expiry-time** *time*

Example:

```
switch(config-macsec-policy)#sak-expiry-time 100
```

Configures the time in seconds to force an SAK rekey. This command can be used to change the session key to a predictable time interval. The default is 0.

c) (Optional) **conf-offset** *name*

Example:

```
switch(config-macsec-policy)#conf-offset CONF-OFFSET-0
```

Configures one of the following confidentiality offsets in the Layer 2 frame, where encryption begins: CONF-OFFSET-0, CONF-OFFSET-30, or CONF-OFFSET-50. This command might be necessary for intermediate switches to use packet headers {dmac, smac, etype} like MPLS tags.

d) (Optional) **show macsec policy**

Example:

```
switch#show macsec policy
```

Displays the MACsec policy configuration.

e) (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Configure MACsec EAP

Beginning with Cisco NX-OS Release 10.4(1)F, you can use MACsec EAP profile for 802.1X authentication.

Before you begin

- Enable the 802.1X feature on the Cisco NX-OS device.
- Configure MACsec command which specifies should-secure (default) or must-secure macsec policy

Procedure

Step 1 **configure terminal**

Enters global configuration mode.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 **interface ethernet** *slot / port*

Example:

```
switch(config)#interface ethernet 1/30
switch(config-if)#
```

Selects the interface to configure and enters interface configuration mode.

Step 3 **[no] macsec eap policy** *policy name*

Example:

```
switch(config-if)#macsec eap policy P1
switch(config-eap-profile)#
```

Creates the MACsec eap profile.

The **no** form of the command is used to disable the MACsec eap profile.

Step 4 **[no] dot1x supplicant eap profile** *eap profile name* }

Enters global configuration mode.

Example:

```
switch(config-if)#dot1x supplicant eap profile TLS
```

Configures the eap profile to be used by the supplicant.

QKD integration with SKIP on MACsec

Recent advancements in quantum computing have exposed vulnerabilities in various cryptographic algorithms, making them unsecured for future applications. The RSA (integer factorization) and DHE (discrete logarithms) public key algorithms, which rely on computational complexity, are now at risk of being solved by quantum computers using Shor's or Grover's algorithm.

As a result, establishing a shared secret key between communicating parties has become a significant challenge. To avoid this issue, you can configure quantum-safe algorithms or implement a Quantum Key Distribution (QKD).

About QKD integration with secure key integration protocol

Integrating Secure Key Integration Protocol (SKIP) protocol to the switches empowers to establish communication with external quantum devices. This advancement allows for the utilization of Quantum Key Distribution (QKD) devices in the exchange of MACsec encryption keys between switches.

QKD operates on the principles of quantum physics, utilizing the quantum state of photons to encode and share information through an optical link. Additionally, an authenticated classical channel is used for sharing measurements. The change in quantum states helps the two end parties of the communication channel to identify any interception of their key.

QKD is a secured key exchange mechanism against quantum attacks even in the future advancements in cryptanalysis or quantum computing. QKD doesn’t require continual updates based on discovered vulnerabilities.

Guidelines and limitations

Use this reference to review the guidelines and limitations.

Guidelines and limitations information

The integration of QKD with SKIP for MACsec communication has the following guidelines and limitations:

- Feature limitations
  - Ensure that the QKD server is accessible through the management interface if switches have an HTTPS connection that is established with it.
  - The SKIP protocol is available only on the interfaces that support MACsec encryption.
  - MACsec PPK session and EAP-TLS sessions are not supported on the same interface.
  - A switch can connect to only one QKD server and one QKD profile per switch.
  - For SKIP protocol, only a single remoteSystemID is supported.
  - For QKD connectivity, IPv6 is not supported.
  - MACsec peers exchanging QKD Keys must be Cisco NX-OS switches.
- Configuration limitations
  - You can use the SKIP protocol only in a point-to-point MACsec link encryption scenario.
  - If MACsec peers are connected to two different QKD servers, the QKD servers synchronize the keys to establish an MKA session. This synchronization ensures that the MACsec key (CKN) and key-string (CAK) are the same at both ends.
  - To establish a secure Transport Layer Security (TLS) connection and enable mutual authentication, you must install trustpoint certificates on the switch. These certificates allow the switch to obtain keys from the server. For more information, see chapter Configuring PKI in *Cisco Nexus 9000 Series NX-OS Security Configuration Guide.*
  - QKD MACsec session fails due to configuration or invalid key errors. There is no FALLBACK mechanism for the preshared key (PSK).
  - Once the MACsec session is established, any modifications to the QKD profile will cause traffic loss in **MUST SECURE MACSEC** mode.
  - Once the MACsec session is up with QKD keys derived from KME Server, trustpoint modifications that are part of the QKD profile will not have any effect on the current sessions.
  - The remoteSystemID attribute is mandatory for the capability response.

QKD MACsec fallback to PSK support

Supported platforms and releases for SKIP

Table 141: CloudScale switches

Platform	Release
• N9K-C9348GC-FXP • N9K-C93216TC-FX2 • N9K-C93360YC-FX2 • N9K-C9336C-FX2 • N9K-C9348GC-FX3 • N9K-C9348D-GX2A • N9K-C9332D-H2R	10.4(3)F

Configuring point-to-point MACsec link encryption using SKIP

In point-to-point MACsec Link Encryption, secure encryption is established by using SKIP in switches. This encryption is set up between two interfaces in peer switches and requires the assistance of a QKD device network. Instead of the switches network, the QKD network shares the MACsec encryption key. Therefore, when a switch is required to create a MACsec link between peer switch interfaces, it contacts the external QKD device and requests the key. The external QKD device then generates a key pair consisting of the key ID and the key.

The Key ID acts as the unique ID string for the key (Shared Secret). The QKD device shares both the key ID and Key with the switch, while the switch only shares the key ID with its peer. The Peer switch uses this Key ID to retrieve encryption keys from its QKD device. Hence, Quantum networks always securely communicate encryption keys.

Enable postquantum cryptography

Postquantum cryptography (PQC) provides enhanced security against future quantum-based threats. Enabling this feature ensures that your network communications are secured with advanced, quantum-resistant algorithms.

Follow these steps to enable postquantum cryptography on your switch:
Before you begin

- Configure MACsec Pre-Shared Key (PSK).
- Configure MACsec in the PPK mode.
- An external QKD device network.
- Add the QKD server CA to the trustpoint in the switch and import the QKD server root CA certificate to the switch.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **feature cryptopqc**

Example:

```
switch(config)#feature cryptopqc
```

Enables post quantum cryptography (cryptopqc) on the switch.

Step 3 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Enable MACsec and MKA features

Use this task when you require secure Layer 2 encryption on your network to protect data in transit between network devices.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **feature macsec**

Example:

```
switch(config)#feature macsec
```

Enables MACsec and MKA on the switch.

Step 3 (Optional) copy running-config startup-config

Example:

```
switch#copy running-config startup-config
```

Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configure quantum key distribution profile

This task enables you to create and configure a QKD profile on a Cisco switch. The configuration specifies the QKD profile name, assigns the remote KME server and port, and links the necessary certificate trustpoint for transport security.

Procedure

Step 1 **crypto qkd profile** *name*

Example:

```
switch(config)#crypto qkd profile ppk1
switch(config-crypto-qkd-profile)#
```

Creates a QKD profile with name ppk1.

Step 2 **kme server** *hostname/IP* **port** *portnumber*

Example:

```
switch(config-crypto-qkd-profile)#kme server 192.0.2.1 port 6000
```

Configures Key Management Engine (KME)/QKD server IP and TCP port number.

Note
Port number is optional. By default, port number will be 443.

Step 3 **transport tls authentication-type trustpoint** *trustpoint name*

Example:

```
switch(config-crypto-qkd-profile)#transport tls authentication-type trustpoint tp1
```

Configures the CA (Certificate Authority) trustpoint. To create a trustpoint, refer to Configuring PKI section.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configure a MACsec policy and assign a PPK crypto-QKD profile

Enabling MACsec on your network switch helps protect Ethernet traffic by encrypting data at Layer 2. Assigning a PPK (Pre-shared Primary Key) crypto-QKD profile to a MACsec policy ensures secure key

management and automated encryption key exchanges between devices. This configuration is essential for organizations that require secure network communications and compliance with data protection standards.

Procedure

Step 1 **macsec policy** *name*

Example:

```
switch(config)#macsec policy test-policy
```

Creates a MACsec policy.

Step 2 **ppk crypto-qkd-profile** *name*

Example:

```
switch(config-macsec-policy)#ppk crypto-qkd-profile ppk1
```

Configures the PPK profile name.

Step 3 (Optional) copy running-config startup-config

Example:

```
switch#copy running-config startup-config
```

Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuration examples

The following examples shows configuration of QKD profile and display of the configured details:

Configuration examples of QKD profile

- Configuring QKD profile

```
switch(config)#feature cryptopqc
switch(config)#crypto qkd profile ppk1
switch(config-crypto-qkd-profile)#kme server 192.0.2.1 port 5000
switch(config-crypto-qkd-profile)#transport tls authentication-type trustpoint tp1
switch(config-crypto-qkd-profile)#end
switch#
```

- Displaying QKD configuration

```
switch#show running-config cryptopqc
!Command: show running-config cryptopqc
!Running configuration last done at: Mon Jan 29 22:19:16 2024
!Time: Mon Jan 29 22:19:35 2024
version 10.4(3) Bios:version 05.51
feature cryptopqc
crypto qkd profile ppk1
kme server 192.0.2.1 port 5000
transport tls authentication-type trustpoint tp1
switch#
```

Configuration examples of PPK profile

The following examples shows configuration of PPK profile on MACsec policy and display of the configured details:

- Configuring PPK profile on MACsec policy

```
switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)#macsec policy test
switch(config-macsec-policy) # ppk crypto-qkd-profile ppk1
switch(config-macsec-policy)#sak-expiry-time 1800
switch(config-macsec-policy)#exit
switch(config)#end
```

- Display of configured MACsec policy

```
switch#show macsec policy test
MACSec Cipher     Pri Window Offset Security     SAKRekey   timeICV     Policy Indicator
     Include-SCI
--------------------------------- ---------------- ---- ------------ --------
-------------- -------------- ------------- ------------
test   GCM-AES-XPN-256 16     148809600 0         should-secure 1800         FALSE
             TRUE
MACSec Policy     PPK Crypto-QKD-Profile Name
--------------------------------- --------------------------------
test                ppk1
switch#
```

Configuration examples of key chain and MACsec policy on an interface

The following example shows configuration of key chain, MACsec policy on an interface, and display of configured details:

- Configuring key chain

```
switch(config)#key chain KC1 macsec
switch(config-macseckeychain)#key 10100000
switch(config-macseckeychain-macseckey)#key-octet-string
F123456789ABCDEF0123456789ABCDEFF123456789ABCDEF0123456789ABCDEF cryptographic-algorithm
 AES_256_CMAC
switch(config-macseckeychain-macseckey)#exit
```

- Configuring MACsec policy to an interface

```
switch(config)#interface Ethernet 1/21
switch(config-if)#macsec keychain KC1 policy test
```

- Displaying MACsec session

```
switch#show macsec mka session
Interface     Local-TxSCI# Peers     Status     Key-Server Auth Mode
------------------ ------------------------------- ------------------ ------------------
 ------------------ ------------------
Ethernet1/21  6cb2.ae9f.e766/0001 1     Secured         No     PRIMARY-PPK
```

Configuration examples of point-to-point MACsec QKD profile

The following examples show configuring point-to-point MACsec QKD profile, binding QKD profile to MACsec policy and binding the MACsec policy to the interface:

Note Make sure that the KME1 and KME2 servers must be active for connection through management port.

Configuring Switch1

```
switch1#configure terminal
switch1(config)#crypto ca trustpoint tp1
switch1config-trustpoint)# exit

switch1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch1(config)#feature cryptopqc
switch1(config)#crypto qkd profile PPK1
switch1(config-crypto-qkd-profile)#kme server KME1 port 7010
switch1(config-crypto-qkd-profile)#transport tls authentication-type trustpoint tp1
switch1(config-crypto-qkd-profile)#exit

switch1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch1(config)#feature macsec
switch1(config)#macsec policy MP1
switch1(config-macsec-policy)#ppk crypto-qkd-profile PPK1
switch1(config-macsec-policy)#exit
switch1(config)#interface Ethernet1/21
switch1(config-if)#macsec keychain KC1 policy MP1
switch1(config-if)#exit
switch1(config)#interface Ethernet1/22
switch1(config-if)#macsec keychain KC1 policy MP1
switch1(config-if)#exit
switch1#
```

Configuring Switch2

```
switch2#configure terminal
switch2(config)#crypto ca trustpoint tp1
switch2config-trustpoint)# exit

switch2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch2(config)#feature cryptopqc
switch2(config)#crypto qkd profile PPK1
switch2(config-crypto-qkd-profile)#kme server KME2 port 7010
switch2(config-crypto-qkd-profile)#transport tls authentication-type trustpoint tp1
switch2(config-crypto-qkd-profile)#exit

switch2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch2(config)#feature macsec

switch2(config)#macsec policy MP1
switch2(config-macsec-policy)#ppk crypto-qkd-profile PPK1
switch2(config-macsec-policy)#exit
switch2(config)#interface Ethernet1/53
switch2(config-if)#macsec keychain KC1 policy MP1
switch2(config-if)#exit
switch2(config)#interface Ethernet1/54
switch2(config-if)#macsec keychain KC1 policy MP1
switch2(config-if)#exit
switch2#
```

The following shows the output of configuration on Switch 1 and Switch 2:

Displaying Switch1 configuration output:

```
switch1#show macsec mka session
Interface     Local-TxSCI# Peers     Status     Key-Server     Auth Mode
----------------- ---------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/22   3c8b.7ffe.0244/0001 1     Secured         Yes        PRIMARY-PPK
Ethernet1/21   3c8b.7ffe.0240/0001 1     Secured         Yes        PRIMARY-PPK
N9K3K STANDARD TEMPLATE FOR FEATURE REVIEWS
----------------- ---------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 2
Secured Sessions : 2
Pending Sessions : 0
switch1#
```

Displaying Switch2 configuration output:

```
switch2#show macsec mka session
Interface     Local-TxSCI# Peers     Status     Key-Server     Auth Mode
----------------- ---------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/53  5451.deb8.62b4/0001 1         Secured     No          PRIMARY-PPK
Ethernet1/54  5451.deb8.62b8/0001 1         Secured     No          PRIMARY-PPK
----------------- ---------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 2
Secured Sessions : 2
Pending Sessions : 0
switch2#
```

About Configurable EAPOL Destination and Ethernet Type

Beginning Cisco NX-OS Release 9.2(2), Cisco enables networks with WAN MACsec to change the Extensible Authentication Protocol (EAP) over LAN (EAPOL) protocol destination address, and the Ethernet type values to nonstandard values.

Configurable EAPOL MAC and Ethernet type provides you the ability to change the MAC address and the Ethernet type of the MKA packet, inorder to allow CE device to form MKA sessions over the ethernet networks that consume the standard MKA packets.

The EAPOL destination Ethernet type can be changed from the default Ethernet type of 0x888E to an alternate value or, the EAPOL destination MAC address can be changed from the default DMAC of 01:80:C2:00:00:03 to an alternate value, to avoid being consumed by a provider bridge.

This feature is available at the interface level and the alternate EAPOL configuration can be changed on any interface at any given time as follows:

- If the MACsec is already configured on an interface, the sessions will come up with a new alternate EAPOL configuration.
- When MACsec is not configured on an interface, the EAPOL configuration is applied to the interface and is effective when MACsec is configured on that inferface.

Enable EAPOL configuration

You can enable the EAPOL configuration on any available interface.

Before you begin

Make sure that MACsec is enabled.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface** *name*

Example:

```
switch(config)#interface ethernet 1/1
switch(config-if)#
```

Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port.

Step 3 **eapol mac-address** *mac_address* [**ethertype** *eth_type*]

Example:

```
switch(config-if)#eapol mac-address 0180.c200.0004 ethertype 0x888f
```

Enables the EAPOL configuration on the specified interface type and identity.

Note
If the ethernet type is not specified, the default ethernet type of MKA packets, which is 0x888e, is considered.

Step 4 **eapol mac-address broadcast-address** [**ethertype** *eth_type*]

Example:

```
switch(config-if)#eapol mac-address ffff.ffff.ffff ethertype 0x888f
```

Enables the broadcast address as the alternate mac address.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Step 6 **show macsec mka session detail**

Example:

```
switch#show macsec mka session detail
```

Displays the EAPOL settings.

Disable EAPOL configuration

You can disable the EAPOL configuration on any available interface.

Procedure

Step 1 **configure terminal**

Example:

```
switch#configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **interface** *name*

Example:

```
switch(config)#interface ethernet 1/1
switch(config-if)#
```

Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port.

Step 3 [**no**] **eapol mac-address** *mac_address* [**ethertype** *eth_type* **]**

Example:

```
switch(config-if)#no eapol mac-address 0180.c200.0004 ethertype 0x888f
```

Disables the EAPOL configuration on the specified interface type and identity.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch#copy running-config startup-config
```

Copies the running configuration to the startup configuration.

MACsec configuration verification commands

Use this reference to verify the MACsec configuration.

Table 142: MACsec configuration verification commands

Command	Purpose
show key chain *name*	Displays the keychain configuration.
show macsec mka session [**interface** *type slot*/*port*] [**detail**]	Displays information about the MACsec MKA session for a specific interface or for all interfaces.

Command	Purpose
show macsec mka session details	Displays information about the MAC address and the ethernet type that is currently used by the interfaces for all EAPOL packets.
show macsec mka summary	Displays the MACsec MKA configuration.
show macsec policy [*policy-name*]	Displays the configuration for a specific MACsec policy or for all MACsec policies.
show running-config macsec	Displays the running configuration information for MACsec.

The following example displays information about the MACsec MKA session for all interfaces. .

```
switch#show macsec mka session
Interface      Local-TxSCI#Peers    Status    Key-Server    Auth Mode
------------------ ---------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet3/20   006b.f1be.db38/0001    1         Secured    Yes          PRIMARY-PSK
Ethernet4/20   006b.f1be.e2f4/0001    1         Secured    No           PRIMARY-PSK
Ethernet1/1    00a3.8eff.e474/0001    0         Pending    Yes          PRIMARY-PSK
Ethernet1/2    00a3.8eff.e478/0001    1         Secured    Yes          PRIMARY-PSK
```

```
switch#show macsec mka session
Interface          Local-TxSCI#Peers            Status             Key-Server         Auth
 Mode
------------------ ---------------------------------- ------------------ -----------------
------------------ ------------------
Ethernet2/2        2c33.11b8.7d14/0001              1                  Secured
Yes                PRIMARY-PSK
Ethernet2/3        2c33.11b8.7d18/0001              1                  Secured
Yes                PRIMARY-PSK
------------------ ---------------------------------- ------------------ -----------------
------------------ ------------------
Total Number of Sessions : 2
Secured Sessions : 2
Pending Sessions : 0
```

The following example displays information about the MACsec MKA session for a specific interface. In addition to the common elements of the table as described in the previous example, the following also identifies the authentication mode which defines the current MACsec session type.

```
switch#show macsec mka session interface ethernet 1/1
Interface      Local-TxSCI# Peers   Status        Key-Server     Auth Mode
-------------- -------------------- --------- ------------- -------------- -------------
Ethernet1/1    70df.2fdc.baf4/0001      0     Pending       Yes            PRIMARY-PSK
Ethernet1/1    70df.2fdc.baf4/0001      1     Secured       No             FALLBACK-PSK
```

The following example displays detailed information about the MACsec MKA session for a specific Ethernet interface:

```
Interface Name          : Ethernet2/2
    Session Status                    : SECURED - Secured MKA Session with MACsec
    Local Tx-SCI                      : 2c33.11b8.7d14/0001
    Local Tx-SSCI                     : 2
    MKA Port Identifier               : 2
    CAK Name (CKN)                    : 12
    CA Authentication Mode            : PRIMARY-PSK
    Member Identifier (MI)            : B54263EF7949A561E25CE617
    Message Number (MN)               : 523
    MKA Policy Name                   : tests2
    Key Server Priority               : 16
    Key Server                        : Yes
```

```
    Include ICV                          : No
    SAK Cipher Suite                     : GCM-AES-XPN-256
    SAK Cipher Suite (Operational)       : GCM-AES-XPN-256
    Replay Window Size                   : 148809600
    Confidentiality Offset               : CONF-OFFSET-0
    Confidentiality Offset (Operational): CONF-OFFSET-0
    Latest SAK Status                    : Rx & TX
    Latest SAK AN                        : 0
    Latest SAK KI                        : B54263EF7949A561E25CE61700000001
    Latest SAK KN                        : 1
    Last SAK key time                    : 12:59:38 PST Tue Mar 19 2019
    CA Peer Count                        : 1
    Eapol dest mac                       : 0180.c200.0003
    Ether-type                           : 0x888e
Peer Status:
    Peer MI                              : 2C2C090E62A96F4D6E018210
    RxSCI                                : 2c33.11b8.8b88/0001
    Peer CAK                             : Match
    Latest Rx MKPDU                      : 13:16:54 PST Tue Mar 19 2019
```

The following example displays the MACsec MKA configuration:

```
switch#show macsec mka summary
Interface          MACSEC-policy                    Keychain
------------------ -------------------------------- ------------------------
Ethernet2/13       1                                1/10000000000000000
Ethernet2/14       1                                1/10000000000000000
```

The following example displays the configuration for all MACsec policies:

```
switch#show macsec policy
MACSec Policy        Cipher          Pri  Window     Offset   Security   SAK Rekey time ICV
 Indicator  Include-SCI
------------------ -------------     --- --------   --------  --------  --------------
------------ -------------
KC256-Po117b        GCM-AES-256      16   148809600    0    should-secure  pn-rollover
FALSE        True
pol1                GCM-AES-XPN-256  100  148809600    30   must-secure    60
FALSE        True
pol256-FanO         GCM-AES-XPN-256  16   148809600    0    must-secure    60
FALSE        True
pol256-MCT          GCM-AES-XPN-256  16   148809600    0    should-secure  60
FALSE        FALSE
system-default-
macsec-policy       GCM-AES-XPN-256  16   148809600    0    should-secure  pn-rollover
FALSE        FALSE
test1               GCM-AES-XPN-256  16   148809600    0    should-secure  pn-rollover
FALSE        True
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is not configured:

```
key chain KC256-1 macsec
  key 2000
    key-octet-string 7 075e701e1c5a4a5143475e5a527d7c7c706a6c724306170103555a5c57510b051e47080
a05000101005e0e50510f005c4b5f5d0b5b070e234e4d0a1d0112175b5e cryptographic-algorithm
AES_256_CMAC
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is configured:

```
key chain KC256-1 macsec
  key 2000
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
```

Display MACsec Statistics

Display MACsec statistics with these commands. Each description explains the command output.

Table 143: Show commands for Media Access Control Security (MACsec) statistics

Command	Description
show macsec mka statistics [**interface** *type slot*/*port*]	View statistics for MACsec Key Agreement (MKA).
show macsec secy statistics [**interface** *type slot*/*port*]	View MACsec security statistics.

This example shows Media Access Control Security (MACsec) Key Agreement (MKA) statistics for your selected Ethernet interface.

```
switch#show macsec mka statistics interface ethernet 2/2

Per-CA MKA Statistics for Session on interface (Ethernet2/2) with CKN 0x10
============================================================================
CA Statistics
   Pairwise CAK Rekeys..... 0

SA Statistics
   SAKs Generated.......... 0
   SAKs Rekeyed............ 0
   SAKs Received........... 0
   SAK Responses Received.. 0

MKPDU Statistics
   MKPDUs Transmitted...... 1096
      "Distributed SAK".. 0

MKPDUs Validated & Rx... 0
   "Distributed SAK".. 0

MKA Statistics for Session on interface (Ethernet2/2)
=====================================================
CA Statistics
   Pairwise CAK Rekeys..... 0

SA Statistics
   SAKs Generated.......... 0
   SAKs Rekeyed............ 0
   SAKs Received........... 0
   SAK Responses Received.. 0

MKPDU Statistics
   MKPDUs Transmitted...... 1096
      "Distributed SAK".. 0
   MKPDUs Validated & Rx... 0
      "Distributed SAK".. 0
   MKPDUs Tx Success.......... 1096
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUS No Tx on intf down.. 0
   MKPDUS No Rx on intf down.. 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 0
```

```
MKPDU Failures
   MKPDU Rx Validation ..................... 0
   MKPDU Rx Bad Peer MN..................... 0
   MKPDU Rx Non-recent Peerlist MN.......... 0
   MKPDU Rx Drop SAKUSE, KN mismatch........ 0
   MKPDU Rx Drop SAKUSE, Rx Not Set......... 0
   MKPDU Rx Drop SAKUSE, Key MI mismatch.... 0
   MKPDU Rx Drop SAKUSE, AN Not in Use...... 0
   MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set... 0
   MKPDU Rx Drop Packet, Ethertype Mismatch. 0

SA Failures
   SAK Generation................... 0
   Hash Key Generation.............. 0
   SAK Encryption/Wrap.............. 0
   SAK Decryption/Unwrap............ 0

CA Failures
   ICK Derivation................... 0
   KEK Derivation................... 0
   Invalid Peer MACsec Capability... 0

MACsec Failures
   Rx SA Installation............... 0
   Tx SA Installation............... 0
```

This example shows security statistics for Media Access Control Security (MACsec) on your selected Ethernet interface.

Note This section explains differences for uncontrolled and controlled packets in Rx and Tx statistics.

- Rx statistics
  - Uncontrolled = Encrypted and unencrypted
  - Controlled = Decrypted
- Tx statistics
  - Uncontrolled = Unencrypted
  - Controlled = Encrypted
  - Common = Encrypted and unencrypted

```
switch#show macsec secy statistics interface e2/28/1

Interface Ethernet2/28/1 MACSEC SecY Statistics:
-----------------------------------------
Interface Rx Statistics:
   Unicast Uncontrolled Pkts: 14987
   Multicast Uncontrolled Pkts: 1190444
   Broadcast Uncontrolled Pkts: 4
   Uncontrolled Pkts - Rx Drop: 0
   Uncontrolled Pkts - Rx Error: 0
   Unicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts: 247583
```

```
   Controlled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   In-Octets Uncontrolled: 169853963 bytes
   In-Octets Controlled: 55027017 bytes
   Input rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)

Interface Tx Statistics:
   Unicast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Uncontrolled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Uncontrolled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   Unicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts: 205429
   Controlled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   Out-Octets Uncontrolled: N/A (N9K-X9736C-FX not supported)
   Out-Octets Controlled: 20612648 bytes
   Out-Octets Common: 151787484 bytes
   Output rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Output rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Output rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Output rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)

SECY Rx Statistics:
   Transform Error Pkts: N/A (N9K-X9736C-FX not supported)
   Control Pkts: 952284
   Untagged Pkts: N/A (N9K-X9736C-FX not supported)
   No Tag Pkts: 0
   Bad Tag Pkts: 0
   No SCI Pkts: 0
   Unknown SCI Pkts: 0
   Tagged Control Pkts: N/A (N9K-X9736C-FX not supported)

SECY Tx Statistics:
   Transform Error Pkts: N/A (N9K-X9736C-FX not supported)
   Control Pkts: 967904
   Untagged Pkts: N/A (N9K-X9736C-FX not supported)

SAK Rx Statistics for AN [3]:
   Unchecked Pkts: 0
   Delayed Pkts: 0
   Late Pkts: 0
   OK Pkts: 1
   Invalid Pkts: 0
   Not Valid Pkts: 0
   Not-Using-SA Pkts: 0
   Unused-SA Pkts: 0
   Decrypted In-Octets: 235 bytes
   Validated In-Octets: 0 bytes

SAK Tx Statistics for AN [3]:
   Encrypted Protected Pkts: 2
   Too Long Pkts: N/A (N9K-X9736C-FX not supported)
   SA-not-in-use Pkts: N/A (N9K-X9736C-FX not supported)
   Encrypted Protected Out-Octets: 334 bytes
switch#
```

Configuration Example for MACsec

Use these examples to configure MACsec policies, set up keychains, enforce options, and monitor sessions.

Example 1: Configure a user-defined MACsec policy and apply the policy to interfaces

Create a custom MACsec policy, configure parameters, assign the policy to interface ranges, and verify your configuration with example command output.

```
switch(config)# macsec policy 1
switch(config-macsec-policy)# cipher-suite GCM-AES-256
switch(config-macsec-policy)# window-size 512
switch(config-macsec-policy)# key-server-priority 0
switch(config-macsec-policy)# conf-offset CONF-OFFSET-0
switch(config-macsec-policy)# security-policy should-secure
switch(config-macsec-policy)# exit

switch(config)# interface range Ethernet2/13 - 14
switch(config-if-range)# macsec keychain 1 policy 1
switch(config-if-range)# exit

switch# show macsec mka summary
Interface          MACSEC-policy                    Keychain
------------------ -------------------------------- ------------------------
Ethernet2/13       1                                1/10000000000000000
Ethernet2/14       1                                1/10000000000000000

switch# show macsec mka session
Interface      Local-TxSCI          # Peers    Status     Key-Server
-------------- -------------------- ---------- ---------- -------------
Ethernet2/13   006b.f1be.d31c/0001  1          Secured    Yes
Ethernet2/14   006b.f1be.d320/0001  1          Secured    No

switch# show running-config macsec
!Command: show running-config macsec
!Time: Mon Dec  5 04:53:40 2016
  version 7.0(3)I4(5)

  version 9.2(1)feature macsec
macsec policy 1
 cipher-suite GCM-AES-256
 key-server-priority 0
 window-size 512
 conf-offset CONF-OFFSET-0
 security-policy should-secure

interface Ethernet2/13
 macsec keychain 1 policy 1

interface Ethernet2/14
 macsec keychain 1 policy 1
```

Example 2: Configure a MACsec keychain and apply the system default policy to interfaces

Create a MACsec keychain, assign keys and cryptographic algorithms, apply the default policy to interfaces, and review example outputs.

```
switch(config)# key chain 1 macsec
switch(config-macseckeychain)# key 1000
switch(config-macseckeychain-macseckey)# key-octet-string
```

```
abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789 cryptographic-algorithm
aes_256_CMAC
switch(config-macseckeychain-macseckey)# exit

switch(config)# interface range Ethernet2/13 - 14
switch(config-if-range)# macsec keychain 1
switch(config-if-range)# exit
switch(config)#

switch# show running-config macsec
!Command: show running-config macsec
!Time: Mon Dec  5 04:50:16 2016
version 7.0(3)I4(5)
feature macsec
interface Ethernet2/13
  macsec keychain 1 policy system-default-macsec-policy
interface Ethernet2/14
  macsec keychain 1 policy system-default-macsec-policy

switch# show macsec mka session
Interface          Local-TxSCI                       # Peers            Status
Key-Server         Auth Mode
------------------ --------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet2/2        2c33.11b8.7d14/0001               1                  Secured
Yes                PRIMARY-PSK
Ethernet2/3        2c33.11b8.7d18/0001               1                  Secured
Yes                PRIMARY-PSK
------------------ --------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 2
Secured Sessions : 2
Pending Sessions : 0

switch# show macsec mka summary
Interface          Status   Cipher (Operational)   Key-Server   MACSEC-policy
  Keychain  Fallback-keychain
------------------ -------- ---------------------- ------------ --------------------------
 ---------- ---------------
Ethernet2/1        down     -                      -            tests1
 keych1    no keychain
Ethernet2/2        Secured  GCM-AES-XPN-256        Yes          tests2
 keych2    no keychain
Ethernet2/3        Secured  GCM-AES-256            Yes          tests3
 keyc3     no keychain
```

Example 3: Configuration and output for Peer Enforce Cipher feature

Configure and verify MACsec Peer Enforce Cipher. Review policies, interface mappings, command outputs, and feature updates for MACsec.

```
switch# show key chain
Key-Chain KC1 Macsec
  Key 10000000 -- text 7
"0729701e1d5d4c53404a522d26090f010e6364704053435556 0e007971772a263e30080a0407070303530227257b7321355655095852 5a771b1650382734362e2a"

  cryptographic-algorithm AES_256_CMAC
  send lifetime (always valid) [active]

Key-Chain KC2 Macsec
  Key 10100000 -- text 7
"0729701e1d5d4c53404a522d26090f010e6364704053435556 0e007971772a263e30080a0407070303530227257b7321355655095852 5a771b1650382734362e2a"

  cryptographic-algorithm AES_256_CMAC
```

```
  send lifetime (always valid) [active]

switch#

switch# show running-config macsec

!Command: show running-config macsec
!Running configuration last done at: Mon Apr 17 16:49:57 2023
!Time: Mon Apr 17 16:50:09 2023

version 10.3(3) Bios:version 05.47
feature macsec

macsec policy MP1
  no protocol lldp encrypted
  cipher-suite enforce-peer GCM-AES-XPN-256 GCM-AES-XPN-128

macsec policy MP2
  no protocol lldp encrypted
  cipher-suite enforce-peer GCM-AES-256

macsec policy Test-MP1
  cipher-suite enforce-peer GCM-AES-XPN-256 GCM-AES-XPN-128

macsec policy Test-MP2
  cipher-suite enforce-peer GCM-AES-256
  removed based on input from puajith

interface Ethernet1/97/1
  macsec keychain KC1 policy MP1

interface Ethernet1/97/2
  macsec keychain KC2 policy MP2

switch#

switch# show macsec policy
MACSec Policy                 Cipher               Pri  Window     Offset  Security
SAK Rekey time  ICV Indicator  Include-SCI
----------------------------- -------------------- ---- ---------- ------- -------------
------------- ------------- -------------
MP1                           Enforce-Peer         16   148809600  0       should-secure
pn-rollover    FALSE         TRUE
MP2                           Enforce-Peer         16   148809600  0       should-secure
pn-rollover    FALSE         TRUE
system-default-macsec-policy  GCM-AES-XPN-256      16   148809600  0       should-secure
pn-rollover    FALSE         TRUE

MACSec Policy                 Lldp-bypass
----------------------------- -------------
MP1                           True
MP2                           True
system-default-macsec-policy FALSE

MACSec Policy                 PPK Crypto-QKD-Profile Name
----------------------------- --------------------------------

MACSec Policy                 Cipher-Suite Enforce-Peer
----------------------------- ----------------------------------------------------
MP1                           GCM-AES-XPN-256 GCM-AES-XPN-128
MP2                           GCM-AES-256

switch#
```

Example 4: Sample output of the show macsec mka session detail command

View detailed session status for MACsec. The information includes interface details, session status, cipher suite, session lifetime, and peer data.

```
switch# show macsec mka session details
MKA Detailed Status for MKA Session
-----------------------------------
Interface Name           : Ethernet1/97/1
Session Status           : SECURED - Secured MKA Session with MACsec
Local Tx-SCI             : c4f7.d530.1484/0001
Local Tx-SSCI            : 1
MKA Port Identifier      : 1
CAK Name (CKN)           : 10000000
CA Authentication Mode   : PRIMARY-PSK
Member Identifier (MI)   : D94B90E3FDB111CE583E7158
Message Number (MN)      : 111
MKA Policy Name          : MP1
Key Server Priority      : 16
Key Server               : Yes
Include ICV              : No
SAK Cipher Suite         : GCM-AES-XPN-128
SAK Cipher Suite (Operational) : GCM-AES-XPN-128
Replay Window Size       : 148809600
Confidentiality Offset   : CONF-OFFSET-0
Confidentiality Offset (Operational) : CONF-OFFSET-0
Latest SAK Status        : Rx & TX
Latest SAK AN            : 1
Latest SAK KI            : D94B90E3FDB111CE583E715800000001
Latest SAK KN            : 1
Last SAK key time        : 16:48:41 PST Mon Apr 17 2023
CA Peer Count            : 1
Eapol dest mac           : 0180.c200.0003
Ether-type               : 0x888e
Peer Status:
  Peer MI                : 00110000000100010000001
  RxSCI                  : 0011.0000.0001/0001
  Peer CAK               : Match
  Latest Rx MKPDU        : 16:52:07 PST Mon Apr 17 2023

Interface Name           : Ethernet1/97/2
Session Status           : SECURED - Secured MKA Session with MACsec
Local Tx-SCI             : c4f7.d530.1485/0001
Local Tx-SSCI            : 1
MKA Port Identifier      : 1
CAK Name (CKN)           : 10100000
CA Authentication Mode   : PRIMARY-PSK
Member Identifier (MI)   : 43AE54C19982238C298E0241
Message Number (MN)      : 107
MKA Policy Name          : MP2
Key Server Priority      : 16
Key Server               : Yes
Include ICV              : No
SAK Cipher Suite         : GCM-AES-256
SAK Cipher Suite (Operational) : GCM-AES-256
Replay Window Size       : 148809600
Confidentiality Offset   : CONF-OFFSET-0
Confidentiality Offset (Operational) : CONF-OFFSET-0
Latest SAK Status        : Rx & TX
Latest SAK AN            : 0
Latest SAK KI            : 43AE54C19982238C298E024100000001
Latest SAK KN            : 1
Last SAK key time        : 16:48:42 PST Mon Apr 17 2023
CA Peer Count            : 1
```

```
Eapol dest mac          : 0180.c200.0003
Ether-type              : 0x888e
Peer Status:
  Peer MI               : 00270000000010001000000001
  RxSCI                 : 0027.0000.0001/0001
  Peer CAK              : Match
  Latest Rx MKPDU       : 16:52:06 PST Mon Apr 17 2023

switch#
```

XML output for MACsec show commands

Supported MACsec show commands with XML output:

You can use these MACsec **show** with **| xml** for scripting.

- **show key chain** *name* **| xml**
- **show macsec mka session interface** *interface slot/port* **details | xml**
- **show macsec mka statistics interface** *interface slot/port* **| xml**
- **show macsec mka summary | xml**
- **show macsec policy** *name* **| xml**
- **show macsec secy statistics interface** *interface slot/port* **| xml**
- **show running-config macsec | xml**

The examples help you understand the output of each **show** commands:

Example 1: Displays the keychain configuration.

```
switch#show key chain "Kc2" | xml <?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0:rpm">
 <nf:data>
  <show>
   <key>
    <chain>
     <__XML__OPT_Cmd_rpm_show_keychain_cmd_keychain>
      <keychain>Kc2</keychain>
     </__XML__OPT_Cmd_rpm_show_keychain_cmd_keychain>
    </chain>
   </key>
  </show>
 </nf:data>
</nf:rpc-reply>]]>]]>
```

Example 2: Displays information about the MACsec MKA session for a specific interface.

```
switch#show macsec mka session interface ethernet 4/31 details | xml <?xml version="1.0"
encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <mka>
     <session>
      <__XML__OPT_Cmd_show_macsec_mka_session_interface>
```

```
       <interface>
<__XML__INTF_ifname>
<__XML__PARAM_value>
<__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
</__XML__PARAM_value>
</__XML__INTF_ifname>
       </interface>
       <__XML__OPT_Cmd_show_macsec_mka_session_details>
<details/>
<__XML__OPT_Cmd_show_macsec_mka_session___readonly__>
<__readonly__>
<TABLE_mka_session_details>
<ROW_mka_session_details>
<ifname>Ethernet4/31</ifname>
<status>Secured</status>
<sci>0c75.bd03.5360/0001</sci>
<ssci>1</ssci>
<port_id>1</port_id>
<mi>02000000000000000000000000000000000000000000000000000000000000000000
</mi>
<mi>F511280A765CE41C79458753</mi>
<mn>2770</mn>
<policy>am2</policy>
<ks_prio>0</ks_prio>
<keyserver>No</keyserver>
<cipher>GCM-AES-XPN-256</cipher>
<window>512</window>
<conf_offset>CONF-OFFSET-0</conf_offset>
<sak_status>Rx &amp;amp; TX</sak_status>
<sak_an>1</sak_an>
<sak_ki>516486241</sak_ki>
<sak_kn>90</sak_kn>
<last_sak_rekey_time>07:12:02 UTC Fri Jan 20 2017</last_sak_rekey_ti
me></ROW_mka_session_details> </TABLE_mka_session_details> </__readonly__>
</__XML__OPT_Cmd_show_macsec_mka_session___readonly__>
</__XML__OPT_Cmd_show_macsec_mka_session_details>
</__XML__OPT_Cmd_show_macsec_mka_session_interface> </session> </mka> </macsec> </show>
</nf:data> </nf:rpc-reply>]]>]]>
```

Example 3: Displays MACsec MKA statistics.

```
switch#show macsec mka statistics interface ethernet 4/31 | xml <?xml version="1.0"
encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
<macsec>
    <mka>
     <statistics>
      <__XML__OPT_Cmd_some_macsec_mka_statistics_interface>
       <interface>
<__XML__INTF_ifname>
<__XML__PARAM_value>
<__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
<__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
</__XML__PARAM_value>
</__XML__INTF_ifname>
       </interface>
       <__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
<__readonly__>
<TABLE_mka_intf_stats>
<ROW_mka_intf_stats>
<TABLE_ca_stats>
<ROW_ca_stats>
```

```
<ca_stat_ckn>0x2</ca_stat_ckn>
<ca_stat_pairwise_cak_rekey>0</ca_stat_pairwise_cak_rekey>
<sa_stat_sak_generated>0</sa_stat_sak_generated>
<sa_stat_sak_rekey>0</sa_stat_sak_rekey>
<sa_stat_sak_received>91</sa_stat_sak_received>
<sa_stat_sak_response_rx>0</sa_stat_sak_response_rx>
<mkpdu_stat_mkpdu_tx>2808</mkpdu_stat_mkpdu_tx>
<mkpdu_stat_mkpdu_tx_distsak>0</mkpdu_stat_mkpdu_tx_distsak>
<mkpdu_stat_mkpdu_rx>2714</mkpdu_stat_mkpdu_rx>
<mkpdu_stat_mkpdu_rx_distsak>91</mkpdu_stat_mkpdu_rx_distsak>
</ROW_ca_stats>
</TABLE_ca_stats>
</ROW_mka_intf_stats>
</TABLE_mka_intf_stats>
</__readonly__>
       </__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
       <interface>
<__XML__INTF_ifname>
<__XML__PARAM_value>
<__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
</__XML__PARAM_value>
</__XML__INTF_ifname>
       </interface>
       <__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
<__readonly__>
<TABLE_mka_intf_stats>
<ROW_mka_intf_stats>
<TABLE_idb_stats>
<ROW_idb_stats>
<ca_stat_pairwise_cak_rekey>0</ca_stat_pairwise_cak_rekey>
<sa_stat_sak_generated>0</sa_stat_sak_generated>
<sa_stat_sak_rekey>0</sa_stat_sak_rekey>
<sa_stat_sak_received>91</sa_stat_sak_received>
<sa_stat_sak_response_rx>0</sa_stat_sak_response_rx>
<mkpdu_stat_mkpdu_tx>2808</mkpdu_stat_mkpdu_tx>
<mkpdu_stat_mkpdu_tx_distsak>0</mkpdu_stat_mkpdu_tx_distsak>
<mkpdu_stat_mkpdu_rx>2714</mkpdu_stat_mkpdu_rx>
<mkpdu_stat_mkpdu_rx_distsak>91</mkpdu_stat_mkpdu_rx_distsak>
<idb_stat_mkpdu_tx_success>2808</idb_stat_mkpdu_tx_success>
<idb_stat_mkpdu_tx_fail>0</idb_stat_mkpdu_tx_fail>
<idb_stat_mkpdu_tx_pkt_build_fail>0</idb_stat_mkpdu_tx_pkt_build_fail>
<idb_stat_mkpdu_no_tx_on_intf_down>0</idb_stat_mkpdu_no_tx_on_intf_down>
<idb_stat_mkpdu_no_rx_on_intf_down>0</idb_stat_mkpdu_no_rx_on_intf_down>
<idb_stat_mkpdu_rx_ca_notfound>0</idb_stat_mkpdu_rx_ca_notfound>
<idb_stat_mkpdu_rx_error>0</idb_stat_mkpdu_rx_error>
<idb_stat_mkpdu_rx_success>2714</idb_stat_mkpdu_rx_success>
<idb_stat_mkpdu_failure_rx_integrity_check_error>0</idb_stat_mkpdu_
failure_rx_integrity_check_error><idb_stat_mkpdu_failure_invalid_peer_mn_error>0</idb_stat_mkpdu_fai
lure_invalid_peer_mn_error><idb_stat_mkpdu_failure_nonrecent_peerlist_mn_error>1</idb_stat_mkp
du_failure_nonrecent_peerlist_mn_error><idb_stat_mkpdu_failure_sakuse_kn_mismatch_error>0</idb_stat_mkpdu_
failure_sakuse_kn_mismatch_error><idb_stat_mkpdu_failure_sakuse_rx_not_set_error>0</idb_stat_mkpdu_f
ailure_sakuse_rx_not_set_error><idb_stat_mkpdu_failure_sakuse_key_mi_mismatch_error>0</idb_stat_mk
pdu_failure_sakuse_key_mi_mismatch_error><idb_stat_mkpdu_failure_sakuse_an_not_in_use_error>0</idb_stat_mkpd
u_failure_sakuse_an_not_in_use_error><idb_stat_mkpdu_failure_sakuse_ks_rx_tx_not_set_error>0</idb_stat_m
kpdu_failure_sakuse_ks_rx_tx_not_set_error><idb_stat_mkpdu_failure_sakuse_eapol_ethertype_mismatch_error>0</id
b_stat_mkpdu_failure_sakuse_eapol_ethertype_mismatch_error><idb_stat_sak_failure_sak_generate_error>0</idb_stat_sak_failure_sa
k_generate_error><idb_stat_sak_failure_hash_generate_error>0</idb_stat_sak_failure_h
ash_generate_error><idb_stat_sak_failure_sak_encryption_error>0</idb_stat_sak_failure_
sak_encryption_error><idb_stat_sak_failure_sak_decryption_error>0</idb_stat_sak_failure_
sak_decryption_error><idb_stat_sak_failure_ick_derivation_error>0</idb_stat_sak_failure_
ick_derivation_error><idb_stat_sak_failure_kek_derivation_error>0</idb_stat_sak_failure_
kek_derivation_error><idb_stat_sak_failure_invalid_macsec_capability_error>0</idb_stat_s
ak_failure_invalid_macsec_capability_error><idb_stat_macsec_failure_rx_sa_create_error>0</idb_stat_macsec_fail
ure_rx_sa_create_error><idb_stat_macsec_failure_tx_sa_create_error>0</idb_stat_macsec_fail
```

```
ure_tx_sa_create_error></ROW_idb_stats> </TABLE_idb_stats> </ROW_mka_intf_stats>
</TABLE_mka_intf_stats> </__readonly__>
</__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
</__XML__OPT_Cmd_some_macsec_mka_statistics_interface> </statistics> </mka> </macsec> </show>
 </nf:data> </nf:rpc-reply>]]>]]>
```

Example 4: Displays the MACsec MKA configuration.

```
switch#show macsec mka summary | xml <?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <mka>
     <__XML__OPT_Cmd_some_macsec_summary>
      <__XML__OPT_Cmd_some_macsec___readonly__>
       <__readonly__>
<TABLE_mka_summary>
<ROW_mka_summary>
<ifname>Ethernet2/1</ifname>
<policy>am2</policy>
<keychain>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
</ROW_mka_summary>
<ROW_mka_summary>
<ifname>Ethernet3/1</ifname>
<policy>am2</policy>
<keychain>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
</ROW_mka_summary>

[TRUNCATED FOR READABILITY]

<ROW_mka_summary>
<ifname>Ethernet3/32</ifname>
<policy>am2</policy>
<keychain>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
</ROW_mka_summary>
</TABLE_mka_summary>
       </__readonly__>
      </__XML__OPT_Cmd_some_macsec___readonly__>
     </__XML__OPT_Cmd_some_macsec_summary>
    </mka>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>]]>]]>
```

Example 5: Displays the configuration for a specific MACsec policy.

```
switch#show macsec policy am2 | xml <?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <policy>
     <__XML__OPT_Cmd_some_macsec_policy_name>
      <policy_name>am2</policy_name>
      <__XML__OPT_Cmd_some_macsec___readonly__>
       <__readonly__>
<TABLE_macsec_policy>
<ROW_macsec_policy>
```

```
<name>am2</name>
<cipher_suite>GCM-AES-XPN-256</cipher_suite>
<keyserver_priority>0</keyserver_priority>
<window_size>512</window_size>
<conf_offset>0</conf_offset>
<security_policy>must-secure</security_policy>
<sak-expiry-time>60</sak-expiry-time>
</ROW_macsec_policy>
</TABLE_macsec_policy>
       </__readonly__>
      </__XML__OPT_Cmd_some_macsec___readonly__>
     </__XML__OPT_Cmd_some_macsec_policy_name>
    </policy>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>]]>]]>
```

Example 6: Displays MACsec security statistics.

```
switch#show macsec secy statistics interface ethernet 4/31 | xml <?xml version="1.0"
encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <secy>
     <statistics>
      <interface>
       <__XML__INTF_ifname>
<__XML__PARAM_value>
<__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
</__XML__PARAM_value>
<__XML__OPT_Cmd_some_macsec_secy_statistics___readonly__>
<__readonly__>
<TABLE_statistics>
<ROW_statistics>
<in_pkts_unicast_uncontrolled>0</in_pkts_unicast_uncontrolled>
<in_pkts_multicast_uncontrolled>42</in_pkts_multicast_uncontrolled>
<in_pkts_broadcast_uncontrolled>0</in_pkts_broadcast_uncontrolled>
<in_rx_drop_pkts_uncontrolled>0</in_rx_drop_pkts_uncontrolled>
<in_rx_err_pkts_uncontrolled>0</in_rx_err_pkts_uncontrolled>
<in_pkts_unicast_controlled>0</in_pkts_unicast_controlled>
<in_pkts_multicast_controlled>2</in_pkts_multicast_controlled>
<in_pkts_broadcast_controlled>0</in_pkts_broadcast_controlled>
<in_rx_drop_pkts_controlled>0</in_rx_drop_pkts_controlled>
<in_rx_err_pkts_controlled>0</in_rx_err_pkts_controlled>
<in_octets_uncontrolled>7230</in_octets_uncontrolled>
<in_octets_controlled>470</in_octets_controlled>
<input_rate_uncontrolled_pps>0</input_rate_uncontrolled_pps>
<input_rate_uncontrolled_bps>9</input_rate_uncontrolled_bps>
<input_rate_controlled_pps>0</input_rate_controlled_pps>
<input_rate_controlled_bps>23</input_rate_controlled_bps>
<out_pkts_unicast_uncontrolled>0</out_pkts_unicast_uncontrolled>
<out_pkts_multicast_uncontrolled>41</out_pkts_multicast_uncontrolled>
<out_pkts_broadcast_uncontrolled>0</out_pkts_broadcast_uncontrolled>
<out_rx_drop_pkts_uncontrolled>0</out_rx_drop_pkts_uncontrolled>
<out_rx_err_pkts_uncontrolled>0</out_rx_err_pkts_uncontrolled>
<out_pkts_unicast_controlled>0</out_pkts_unicast_controlled>
<out_pkts_multicast_controlled>2</out_pkts_multicast_controlled>
<out_pkts_broadcast_controlled>0</out_pkts_broadcast_controlled>
<out_rx_drop_pkts_controlled>0</out_rx_drop_pkts_controlled>
<out_rx_err_pkts_controlled>0</out_rx_err_pkts_controlled>
<out_octets_uncontrolled>6806</out_octets_uncontrolled>
```

```
<out_octets_controlled>470</out_octets_controlled>
<out_octets_common>7340</out_octets_common>
<output_rate_uncontrolled_pps>2598190092</output_rate_uncontrolled_pps>
<output_rate_uncontrolled_bps>2598190076</output_rate_uncontrolled_bps>
<output_rate_controlled_pps>0</output_rate_controlled_pps>
<output_rate_controlled_bps>23</output_rate_controlled_bps>
<in_pkts_transform_error>0</in_pkts_transform_error>
<in_pkts_control>40</in_pkts_control>
<in_pkts_untagged>0</in_pkts_untagged>
<in_pkts_no_tag>0</in_pkts_no_tag>
<in_pkts_badtag>0</in_pkts_badtag>
<in_pkts_no_sci>0</in_pkts_no_sci>
<in_pkts_unknown_sci>0</in_pkts_unknown_sci>
<in_pkts_tagged_ctrl>0</in_pkts_tagged_ctrl>
<out_pkts_transform_error>0</out_pkts_transform_error>
<out_pkts_control>41</out_pkts_control>
<out_pkts_untagged>0</out_pkts_untagged>
<rx_sa_an>1</rx_sa_an>
<in_pkts_unchecked>0</in_pkts_unchecked>
<in_pkts_delayed>0</in_pkts_delayed>
<in_pkts_late>0</in_pkts_late>
<in_pkts_ok>1</in_pkts_ok>
<in_pkts_invalid>0</in_pkts_invalid>
<in_pkts_not_valid>0</in_pkts_not_valid>
<in_pkts_not_using_sa>0</in_pkts_not_using_sa>
<in_pkts_unused_sa>0</in_pkts_unused_sa>
<in_octets_decrypted>223</in_octets_decrypted>
<in_octets_validated>0</in_octets_validated>
<tx_sa_an>1</tx_sa_an>
<out_pkts_encrypted_protected>1</out_pkts_encrypted_protected>
<out_pkts_too_long>0</out_pkts_too_long>
<out_pkts_sa_not_inuse>0</out_pkts_sa_not_inuse>
<out_octets_encrypted_protected>223</out_octets_encrypted_protected>
</ROW_statistics>
</TABLE_statistics>
</__readonly__>
</__XML__OPT_Cmd_some_macsec_secy_statistics___readonly__>
       </__XML__INTF_ifname>
      </interface>
     </statistics>
    </secy>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>]]>]]>
```

Example 7: Displays the running configuration information for MACsec.

```
switch#show running-config macsec | xml
!Command: show running-config macsec
!Time: Fri Jan 20 07:12:34 2017

version 7.0(3)I4(6)
****************************************
This may take time. Please be patient.
****************************************
<?xml version="1.0"?>
<nf:rpc xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://www.cis
example.com/nxos:7.0.3.I4.6.:configure_" xmlns:m="http://www.cisco.com/nxos:7.0.3.I4.
6.:_exec" xmlns:m1="http://www.cisco.com/nxos:7.0.3.I4.6.:configure__macsec-poli
cy" xmlns:m2="http://www.cisco.com/nxos:7.0.3.I4.6.:configure__if-eth-non-member
" message-id="1">
  <nf:get-config>
    <nf:source>
      <nf:running/>
```

```
    </nf:source>
    <nf:filter>
      <m:configure>
<m:terminal>
<feature>
<macsec/>
</feature>
<macsec>
<policy>
<__XML__PARAM__policy_name>
<__XML__value>am2</__XML__value>
<m1:cipher-suite>
<m1:__XML__PARAM__suite>
<m1:__XML__value>GCM-AES-XPN-256</m1:__XML__value>
</m1:__XML__PARAM__suite>
</m1:cipher-suite>
<m1:key-server-priority>
<m1:__XML__PARAM__pri>
<m1:__XML__value>0</m1:__XML__value>
</m1:__XML__PARAM__pri>
</m1:key-server-priority>
<m1:window-size>
<m1:__XML__PARAM__size>
<m1:__XML__value>512</m1:__XML__value>
</m1:__XML__PARAM__size>
</m1:window-size>
<m1:conf-offset>
<m1:__XML__PARAM__offset>
<m1:__XML__value>CONF-OFFSET-0</m1:__XML__value>
</m1:__XML__PARAM__offset>
</m1:conf-offset>
<m1:security-policy>
<m1:__XML__PARAM__policy>
<m1:__XML__value>must-secure</m1:__XML__value>
</m1:__XML__PARAM__policy>
</m1:security-policy>
<m1:sak-expiry-time>
<m1:__XML__PARAM__ts>
<m1:__XML__value>60</m1:__XML__value>
</m1:__XML__PARAM__ts>
</m1:sak-expiry-time>
</__XML__PARAM__policy_name>
</policy>
</macsec>
<interface>
<__XML__PARAM__interface>
<__XML__value>Ethernet2/1</__XML__value>
<m2:macsec>
<m2:keychain>
<m2:__XML__PARAM__keychain_name>
<m2:__XML__value>kc2</m2:__XML__value>
<m2:policy>
<m2:__XML__PARAM__policy_name>
<m2:__XML__value>am2</m2:__XML__value>
</m2:__XML__PARAM__policy_name>
</m2:policy>
</m2:__XML__PARAM__keychain_name>
</m2:keychain>
</m2:macsec>
</__XML__PARAM__interface>
</interface>

[TRUNCATED FOR READABILITY]
```

```
<interface>
<__XML__PARAM__interface>
<__XML__value>Ethernet4/31</__XML__value>
<m2:macsec>
<m2:keychain>
<m2:__XML__PARAM__keychain_name>
<m2:__XML__value>kc2</m2:__XML__value>
<m2:policy>
<m2:__XML__PARAM__policy_name>
<m2:__XML__value>am2</m2:__XML__value>
</m2:__XML__PARAM__policy_name>
</m2:policy>
</m2:__XML__PARAM__keychain_name>
</m2:keychain>
</m2:macsec>
</__XML__PARAM__interface>
</interface>
</m:terminal>
      </m:configure>
    </nf:filter>
  </nf:get-config>
</nf:rpc>]]>]]>
```

Macsec supported MIBs and download locations

MACsec supports these MIBs.

- IEEE8021-SECY-MIB
- CISCO-SECY-EXT-MIB

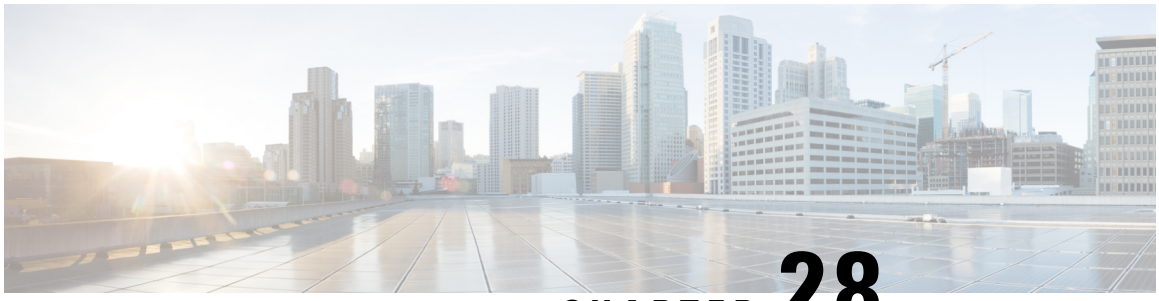
Go to the MIB support list web page to find and download supported MIBs: https://example.com/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html.

Related documentation

Use this table to find related documentation for keychain management and system messages.

Table 144: Find related documentation

Related Topic	Document Title
Keychain management	Nexus 9000 Series NX-OS Security Configuration Guide
System messages	Nexus 9000 Series Nexus Operating System (NX-OS) System Messages References



CHAPTER 28

Configure TCP authentication option

This document describes how to configure TCP authentication option on Cisco NX-OS devices.

- About TCP authentication option, on page 691
- TCP-AO key chain, on page 691
- TCP-AO key rollover, on page 693
- Guidelines and limitations, on page 694
- Configure TCP key chain and keys, on page 694
- TCP keychain verification commands, on page 697
- Configuration example for a TCP keychain, on page 698

About TCP authentication option

With TCP Authentication Option (TCP-AO), defined in RFC 5925, you can protect long-lived TCP connections against replays using stronger Message Authentication Codes (MACs).

TCP-AO is the proposed replacement for TCP MD5, defined in RFC 2385. Unlike TCP MD5, TCP-AO is resistant to collision attacks, provides algorithmic agility and support for key management.

TCP-AO has the following distinct features:

- TCP-AO supports the use of stronger Message Authentication Codes (MACs) to enhance the security of long-lived TCP connections.
- TCP-AO protects long-lived TCP connections against replays and coordinates key changes between endpoints by providing a more explicit key management.

he TCP-AO feature deprecates TCP MD5. Cisco NX-OS devices will continue to support the TCP-MD5 option for legacy BGP peers. However, a configuration in which one end of the peering is configured with the TCP MD5 option, and the other with the TCP-AO option is not supported.

TCP-AO key chain

TCP-AO is based on traffic keys and Message Authentication Codes (MACs) generated using the keys and a MAC algorithm. The traffic keys are derived from master keys that you can configure in a TCP-AO key chain. Use the **key chain** *key-chain-name* **tcp** command in the global configuration mode to create a TCP-AO

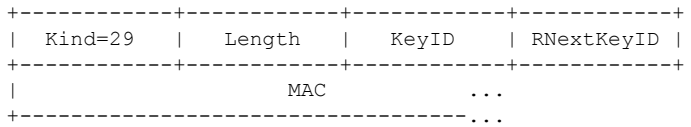
key chain and configure keys in the chain. The TCP-AO key chain must be configured on both the peers communicating via a TCP connection.

Keys in a TCP-AO key chain have the following configurable properties:

Table 145: TCP-AO key chain

Configurable Property	Description
send-id	Key identifier of the TCP-AO option of the outgoing segment. The send identifier configured on a router must match the receive identifier configured on the peer.
recv-id	Key identifier compared with the TCP-AO key identifier of the incoming segment during authentication. The receive identifier configured on a router must match the send identifier configured on the peer.
cryptographic-algorithm	The MAC algorithm to be used to create MACs for outgoing segments. The algorithm can be one of the following: • AES-128-CMAC authentication algorithm • HMAC-SHA-1 authentication algorithm • HMAC-SHA-256 authentication algorithm
include-tcp-options	This flag indicates whether TCP options other than TCP-AO will be used to calculate MACs. With this flag enabled, the contents of all options along with a zero-filled authentication option, is used to calculate the MAC. When the flag is disabled, all options other than TCP-AO are excluded from MAC calculations. This flag is disabled by default. **Note** The configuration of this flag is overridden by the application configuration when the application configuration is available.
send-lifetime	This configuration determines the time for which a key is valid and can be used for TCP-AO-based authentication of TCP segments. When the lifetime of key elapses and the key expires, the next key with the youngest lifetime is selected.
key-string	The key string is a pre-shared master key configured on both peers and is used to derive the traffic keys.

TCP-AO Format



```
...-----------------+
...  MAC (con't)    |
...-----------------+
```

The fields of the TLV format are as follows:

- Kind: Indicates TCP-AO with a value of 29.
- Length: Indicates the length of the TCP-AO sequence.
- KeyID: The send identifier of the Master Key Tuple (MKT) that was used to generate the traffic keys.
- RNextKeyID: The receive identifier of the MKT that is ready to be used to authenticate received segments.
- MAC: The MAC computed for the TCP segment data and the prefixed pseudo header.

Master key tuples

Traffic keys are the keying material used to compute the message authentication codes of individual TCP segments.

Master Key Tuples (MKTs) enable you to derive unique traffic keys, and to include the keying material required to generate those traffic keys. MKTs indicate the parameters under which the traffic keys are configured. The parameters include whether TCP options are authenticated, and indicators of the algorithms used for traffic key derivation and MAC calculation.

Each MKT has the following two identifiers:

- *SendID* : The SendID identifier is inserted as the KeyID identifier of the TCP AO option of the outgoing segments.
- *RecvID* : The RecvID is matched against the TCP AO KeyID of the incoming segments.

TCP-AO key rollover

TCP-AO keys are valid for a defined duration configured using the send-lifetime. If send-lifetime is not configured the key is considered inactive. Key rollover is initiated based on the send lifetimes of keys.

TCP-AO coordinates the use of new MKTs using the RNextKeyID and KeyID field on the TCP-AO option field. For hitless key rollovers, new and old keys in keychain configurations need to have at least 15 minutes of overlap. This is required so that the TCP-AO has enough time to coordinate use of the new MKT.

When key rollover is initiated, one of the peer routers, say Router A, indicates that the rollover is necessary. To indicate that the rollover is necessary, Router A sets the RNextKeyID to the receive identifier (recv-id) of the new MKT to be used. On receiving the TCP segment, the peer router, say Router B, looks up the send identifier (send-id) in its database to find the MKT indicated by the RNextKeyID in the TCP-AO payload. If the key is available and valid, Router B sets the current key to the new MKT. After Router B has rolled over, Router A also sets the current key to the new Primary Key Tuples.

Key rollover is initiated with overlapping send-lifetimes and send-lifetime expiry

If you do not configure a new key that can be activated before the expiry of the current key, the key may time out and expire. Such an expiry can cause retransmissions with the peer router rejecting segments authenticated with the expired key. The connection may fail due to Retransmission Time Out (RTO). When new valid keys are configured and usable, the connection can be re-established.

Guidelines and limitations

Use this reference to review the guidelines and limitations.

Follow the guidelines to configure TCP authentication option:

- The send-id and recv-id of each key in the key chain must be unique. Because send-id and recv-id must be chosen from the range 0 to 255, the TCP-AO key chain can have a maximum of 256 keys.
- Only one keychain can be associated with an application connection. Rollover is always performed within the keys in this keychain.
- If the key in use expires, expect segment loss until a new key that has a valid lifetime is configured on each side and keys rollover.

All the following configurations must be done for a TCP-AO keychain key to be considered active: send-id, recv-id, key-string, send-lifetime and cryptographic-algorithm:

- The key chain software process uses the newest key (the youngest key), as determined by the send-lifetime configuration. If two different keys in the same key chain have the same send-lifetime, the key configured most recently is used instead. Configuring two keys with identical send-lifetimes is not recommended.
- he user must configure at least 15 minutes of overlapping time between two overlapping keys.
- Modifying the configuration of a key in use such as key-string, send-id, recv-id, cryptographic-algorithm or send-lifetime will result in TCP connection flap.
- A keychain's configuration type must match the type it has been linked to within the client protocol. If an attempt is made to mismatch these types, a syslog message is generated to notify the user. For example: It is not supported if a keychain named keychain_abc is configured as a Macsec keychain but is associated as a TCP keychain with BGP. Similarly, the case where the keychain is first associated with the client (a process known as forward-referencing) and then configured as a different keychain type, is also not supported.

Configure TCP key chain and keys

Before you begin

- Ensure that the key-string, send-lifetimes, cryptographic-algorithm, and ids of keys match on both peers.
- Ensure that the send-id on a router matches the recv-id on the peer router. We recommend using the same id for both the parameters unless there is a need to use separate key spaces.
- The send-id and recv-id of a key cannot be reused for another key in the same key chain.
- The key-string is encrypted and stored in the Type-6 format if the AES password encryption feature is enabled and a primary key is configured. Otherwise, the password will be stored in the Type-7 encrypted format.
- For more details, see Configuring a Primary Key and Enabling the AES Password Encryption Feature.

Procedure

Step 1 Use the **key chain** *name* **tcp** command to enter keychain configuration mode for the keychain that you specified.

Example:

```
switch#configure terminal
switch(config)#key chain bgp-keys tcp
```

Step 2 Use the **key** *key-ID* command to enter key configuration mode for the key that you specified.

Example:

```
switch(config-tcpkeychain)# key 13
```

The *key-ID* argument must be a whole number between 0 and 65535.

Step 3 Use the **send-id** *send-ID* command to specify the send identifier for the key.

Example:

```
switch(config-tcpkeychain-tcpkey)#send-id 2
```

The send-ID must be in the range from 0 to 255 and unique value per key chain.

Step 4 Use the **recv-id** *recv-ID* command to specify the receive identifier for the key.

Example:

```
switch(config-tcpkeychain-tcpkey)#recv-id 2
```

The *recv-ID* must be in the range from 0 to 255 and unique value per key chain.

Step 5 Use the **key-string** *[encryption-type] text-string* command to configure the text string for the key. The text-string argument is alphanumeric, case-sensitive, and supports special characters.

Example:

```
switch(config-tcpkeychain-tcpkey)#key-string 0 AS3cureStr1ng
```

The encryption-type argument can be one of the following values:

- 0—The text-string argument that you enter is unencrypted text. This is the default.
- 6—Beginning with Cisco NX-OS Release 10.3(3)F, the Cisco proprietary (Type-6 encrypted) method is supported on Cisco Nexus 9000 Series platform switches.
- 7—The text-string argument that you enter is encrypted. The encryption method is a Cisco proprietary method. This option is useful when you are entering a text string based on the encrypted output of a **show key chain** command that you ran on another Cisco NX-OS device.

The **key-string** command has limitations on using the following special characters in the *text-string*:

Table 146: Configure TCP key chain and keys

Special Character	Description	Comments
|	Vertical bar or pipe	Unsupported at start of key-string

Special Character	Description	Comments
>	Greater than	Unsupported at start of key-string
\	Backslash	Unsupported start or end of a key-string
(	Left parenthesis	Unsupported at start of key-string
'	Apostrophe	Unsupported at start of key-string
"	Quotation mark	Unsupported at start of key-string
?	Question mark	Supported. However, press **Ctrl-V** before entering a question mark (?).

For more information on the special characters usage in commands, see Understanding the Command-Line Interface section.

Step 6 Use the **[no] cryptographic-algorithm {HMAC-SHA-1 | HMAC-SHA-256 | AES-128-CMAC}** command to specify the the algorithm to be used to compute MACs for TCP segments. You can configure only one cryptographic algorithm for each key.

Example:

```
switch(config-tcpkeychain-tcpkey)#cryptographic-algorithm HMAC-SHA-1
```

Step 7 Use the **send-lifetime [local]** *start-time* **duration** [duration-value | infinite | end-time] command to configure a send lifetime for the key.

Example:

```
switch(config-tcpkeychain-tcpkey)#send-lifetime local 01:01:01 Jan 01 2023 01:01:01 Jan 10
 2023
```

By default, the device treats the start-time and end-time arguments as UTC. If you specify the *local* keyword, the device treats these times as local times.

The start-time argument is the time of day and date that the key becomes active.

You can specify the end of the send lifetime with one of the following options:

- *duration* duration-value —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
- *infinite*—The send lifetime of the key never expires.
- end-time —The end-time argument is the time of day and date that the key becomes inactive.

Step 8 Use the **include-tcp-options** command to include the TCP Options of the TCP header.

Example:

```
switch(config-tcpkeychain-tcpkey)#include-tcp-options
```

An optional configuration to specify if the full TCP Options part of the TCP header (other than TCP AO option) needs to be included while computing the 'MAC' digest of the packets.

TCP keychain verification commands

Use this reference to verify the TCP keychain.

Use the following steps to verify the TCP keychain.

Table 147: TCP keychain verification commands

Command	Purpose
show key chain *[name] [detail]*	Displays the keychains configured on the device.

```
switch#show key chain
Key-Chain bgp_keys tcp
Key 2 -- text 7 "070e234f"
send-id 2
recv-id 2
cryptographic-algorithm AES_128_CMAC
send lifetime UTC (08:17:00 May 29 2023)-(08:21:00 May 29 2023)
include-tcp-options
Key 3 -- text 7 "070c2058"
send-id 3
recv-id 4
cryptographic-algorithm HMAC-SHA-1
send lifetime UTC (08:20:00 May 29 2023)-(always valid) [active]
include-tcp-options
Key 12 -- text ""
send lifetime UTC (08:20:00 May 29 2023)-(always valid)
```

Note [active] indicates that the key is valid and active otherwise the key is inactive. In the above example only key 3 is active and usable.

The **show key chain detail** command will explicitly display any active and inactive key(s). In the case of Type 6 encryption, the show key chain detail command will display if the type 6 key-string is decryptable or not. It will also display the newest (youngest) active send key that the client is currently using to authenticate packets.

```
switch#show key chain detail
Key-Chain bgp_keys tcp
Key 1 -- text 6 "JDYk9k4kmaciqaH6Eu2+9C0tmCRl9k7JAMYs/fXGbW1lmHP88PAA=="
Type6 Decryptable: yes
send-id 1
recv-id 1
cryptographic-algorithm HMAC-SHA-1
send lifetime local (18:15:42 May 15 2023)-(always valid) [active]
include-tcp-options
accept-ao-mismatch
Key 2 -- text 6 "JDYkB+Fs8u3ujRDpFSu4tH6H7iTS45JJA6sKeGsBD0L3HjGDeg9AA=="
Type6 Decryptable: yes
send-id 2
recv-id 2
cryptographic-algorithm AES_128_CMAC
send lifetime local (17:10:47 May 15 2023)-(18:15:42 May 15 2023) [inactive]
youngest active send key: 1
```

Configuration example for a TCP keychain

This example shows how to configure a TCP keychain named bgp_keys. Each key text string is encrypted. The keys have overlapping lifetime configurations:

```
key chain bgp_keys tcp
key 1
send-id 1
recv-id 1
key-string 7 070e234f
send-lifetime 01:00:00 Oct 10 2023 01:00:00 Oct 11 2023
cryptographic-algorithm AES-128-CMAC
key 2
send-id 2
recv-id 2
key-string 7 075e731f
send-lifetime 00:45:00 Oct 11 2023 01:00:00 Oct 12 2023
cryptographic-algorithm HMAC-SHA-256
include-tcp-options
```



INDEX

802.1X 265, 271
  configuring 271
  prerequisites 265

A

aaa accounting default 25
aaa accounting default group 48
aaa accounting default local 48
aaa accounting dot1x default group 295
aaa authentication dot1x default group 273
aaa authentication login {mschap | mschapv2} enable 44
aaa authentication login ascii-authentication 110
aaa authentication login chap enable 43
aaa authentication login console 25, 32, 34
aaa authentication login console group 32, 34
aaa authentication login console local 32, 34
aaa authentication login console none 32, 34
aaa authentication login default 25
aaa authentication login error-enable 37
aaa authorization {commands | config-commands} {console | default} {group} 111
aaa authorization {group | local} 137
aaa authorization {ssh-certificate | ssh-publickey} 137
aaa authorization default 137
aaa authorization ssh-certificate default 45–46
aaa group server radius 74
aaa group server tacacs+ 101
aaa user default-role 36
absolute end 388
absolute start 388
accept-lifetime 565
action {drop | forward | redirect} 411

B

BGP 584
  using with Unicast RPF 584

C

chgrp 148
chown 148
class-map 616
clear access-list ipsg stats 544
clear accounting log 55
clear hardware rate-limiter {all | access-list-log | bfd | exception | fex | layer-3 glean | layer-3 multicast local-groups | span-egress} 643
clear hardware rate-limiter module 643
clear ip access-list counters 375
clear ip arp inspection log 532
clear ip arp inspection statistics 532
clear ipv6 access-list counters 376
clear ldap-server statistics 141
clear line 169, 171
clear mac access-list counters 405
clear port-security dynamic 430
clear port-security dynamic address 429
clear radius-server statistics 88
clear ssh hosts 167
clear tacacs-server statistics 117
conf-offset 662
control-plane 616
copy scp 175
copy scp: 155
copy sftp 175
crypto ca authenticate 159
crypto ca crl request 159
crypto ca trustpoint 159
cryptographic-algorithm {HMAC-SHA-1 | HMAC-SHA-256 | HMAC-SHA-384 | HMAC-SHA-512 | MD5} 566

D

deadtime 74
deafult settings 421
  port security 421
denial-of-service attacks 585
  IP address spoofing, mitigating 585
deny 338, 340–341
DoS attacks 585
  Unicast RPF, deploying 585
dot1x default 292
dot1x host-mode {multi-host | single-host} 288
dot1x max-req 293
dot1x port-control {auto | force-authorized | forced-unauthorized} 275
dot1x re-authentication 280
dot1x timeout quiet-period 282

dot1x timeout ratelimit-period 282
dot1x timeout re-authperiod 280
dot1x timeout server-timeout 283
dot1x timeout supp-timeout 283
dot1x timeout tx-period 283

E

encryption decrypt type6 552
encryption delete type6 554
encryption re-encrypt obfuscated 551–552, 562

F

feature dhcp 457
feature dot1x 272
feature macsec 655–656
feature password encryption aes 551, 561
feature port-security 424
feature scp-server 157
feature sftp-server 157
feature ssh 150, 166
feature tacacs+ 97
feature telnet 170
FIPS 17–18
  disabling 18
  enabling 17
fragments {permit-all | deny-all} 338, 340

G

generate type7_encrypted_secret 54, 68–69, 99, 101
guidelines 422
  port security 422

H

hardware access-list tcam region 344
hardware access-list tcam region ing-ifacl qualify udf 361, 397
hardware profile tcam resource service-template 352
hardware profile tcam resource template 351
hardware rate-limiter access-list-log 642
hardware rate-limiter bfd 642
hardware rate-limiter exception 642
hardware rate-limiter fex 642
hardware rate-limiter layer-3 glean 642
hardware rate-limiter layer-3 multicast local-groups 642
hardware rate-limiter span-egress 643
host 382–383

I

ip access-class 342
ip access-group 363
ip access-list 337, 340–341, 362, 371
ip arp inspection log-buffer entries 530
ip arp inspection trust 528
ip arp inspection validate 529
ip arp inspection validate dst-mac 529
ip arp inspection validate ip 529
ip arp inspection validate src-mac 529
ip arp inspection vlan 527, 530
ip dhcp packet strict-validation 444, 463
ip dhcp relay 468, 471
ip dhcp relay address 473
ip dhcp relay address use-vrf 473
ip dhcp relay information option 469
ip dhcp relay information option server-id-override-disable 471
ip dhcp relay information option trust 465
ip dhcp relay information option vpn 470
ip dhcp relay information trust-all 467
ip dhcp relay information trusted 466
ip dhcp relay source-interface 474
ip dhcp relay sub-option circuit-id customized 469
ip dhcp relay sub-option circuit-id format-type string 469
ip dhcp relay sub-option type cisco 470
ip dhcp smart-relay 476
ip dhcp smart-relay global 475
ip dhcp snooping information option 461
ip dhcp snooping ipsg-excluded vlan 543
ip dhcp snooping trust 464
ip dhcp snooping verify mac-address 460
ip dhcp snooping vlan 459
ip port access group 365
ip radius source-interface 75
ip source binding 543
ip tacacs source-interface 102
ip verify source dhcp-snooping-vlan 542
ip verify unicast source reachable-via 590
ip verify unicast source reachable-via any 588
ipv6 access-class 342
ipv6 access-list 337, 340–341
ipv6 address use-link-local-only 487
ipv6 dhcp relay 478
ipv6 dhcp relay address 483
ipv6 dhcp relay option type cisco 479
ipv6 dhcp relay option vpn 479
ipv6 dhcp relay source-interface 485
ipv6 dhcp smart-relay 481
ipv6 dhcp smart-relay global 480
ipv6 port traffic-filter 365
ipv6 traffic-filter 363
ipv6 verify unicast source reachable-via 590
ipv6 verify unicast source reachable-via any 588

K

key 553, 563–564, 566, 657
key chain 559, 563–564, 566, 657
key-chain macsec-psk no-show 657

key-octet-string 658
key-server-priority 661
key-string 563

L

ldap search-map 134
ldap-server deadtime 135–136
ldap-server host 128, 132–133, 135
ldap-server host idle-time 135
ldap-server host password 129, 135
ldap-server host port 129, 133
ldap-server host rootDN 129
ldap-server host test rootDN 135
ldap-server host timeout 129, 133
ldap-server host username 135
ldap-server timeout 131
limitations 422
    port security 422
line vty 342
login block-for 50
login block-for attempts 50
login on-failure log 38
login on-success log 38
login quiet-mode access-class 51

M

mac access-list 395, 397, 399
MAC addresses 416
    learning 416
mac packet-classify 403
mac port access-group 398, 402
macsec policy 660
match {ip | ipv6} address 410
match access-group name 631
match mac address 410

N

no {periodic | absolute} 389
no aaa authentication login {console | default | fallback error local} 26, 35
no aaa authentication login ascii-authentication 43–44
no dot1x system-auth-control 290
no feature dot1x 291
no feature ssh 150, 166, 168
no feature tacacs+ 116
no host 382–383
no ip access-list 343
no ipv6 access-list 343
no key chain 560
no mac access-list 401
no object-group {ip address | ipv6 address | ip port} 385
no ssh key dsa 169
no ssh key rsa 169
no time-range 390
no vlan access-map 412

O

object-group ip address 382
object-group ip port 384
object-group ipv6 address 383

P

password prompt username 53
periodic 388
permit 338, 340–341
permit | deny 396
permit http-method 371
permit ip 362
permit mac 398
permit udf 362
permit vrf 247
PKI 187, 191
    certificate revocation checking 187
    generating RSA key pairs 191
police 624, 630–631
police cir 624, 630–631
policy-map 616
port security 415–416, 418, 421–422
    default settings 421
    description 415
    guidelines 422
    limitations 422
    MAC address learning 416
    MAC move 418
    violations 418

R

radius-server deadtime 81–83
radius-server directed-request 76
radius-server host 54, 67, 69, 74, 78–79, 82
radius-server host accounting 79
radius-server host acct-port 79
radius-server host auth-port 79
radius-server host authentication 79
radius-server host idle-time 82
radius-server host password 82
radius-server host retransmit 78
radius-server host test 82
radius-server host timeout 78
radius-server host username 82
radius-server key 54, 68
radius-server retransmit 77
radius-server test {idle-time} 80
radius-server test {password} 80

radius-server test {username} 80
radius-server timeout 77
reload 348, 352, 362, 397
resequence {ip | ipv6} access-list 342
resequence mac access-list 400
resequence time-range 390
role name 247
role name priv 115
RSA key pairs 191
  generating for PKI 191
rule {deny | permit} command 115

S

sak-expiry-time 661
secure MAC addresses 416
  learning 416
security 416
  port 416
    MAC address learning 416
security-policy 661
send-lifetime 565, 658
server 74–75, 101
service-policy 616
show {ip | ipv6 | access-lists} 386
show aa accounting 55
show aaa accounting 48, 295
show aaa authentication 33–34, 36–37, 55
show aaa authentication login {ascii-authentication | chap | error-enable | mschap | mschapv2} 55
show aaa authentication login {mschap | mschapv2} 45
show aaa authentication login chap 43
show aaa authorization 46, 112, 138
show aaa authorization all 46
show aaa groups 55
show aaa user default-role 37
show accounting log 54
show class-map type control-plane 633
show cli syntax roles network-admin 251
show cli syntax roles network-operator 251
show copp profile 633
show crypto ca certificates 160, 172
show crypto ca crl 160, 172
show dot1x 273, 290
show dot1x {all | interface ethernet} 303
show dot1x all 275, 280, 283, 289, 292–293
show dot1x interface ethernet 275
show encryption service stat 551, 562
show hardware access-list interface input entries detail 375
show hardware access-list tcam region 348, 373
show hardware access-list tcam template 352, 373
show hardware rate-limiter 643–644
show hardware rate-limiter access-list-log 643–644
show hardware rate-limiter bfd 643–644
show hardware rate-limiter exception 643–644
show hardware rate-limiter fex 643–644
show hardware rate-limiter layer-3 glean 643–644
show hardware rate-limiter layer-3 multicast local-groups 643–644
show hardware rate-limiter module 643–644
show hardware rate-limiter span-egress 643–644
show incompatibility nxos bootflash: 620
show interface counters storm-control 572, 579–580
show interface ethernet counters storm-control 579–580
show interface port-channel counters storm-control 579
show interface port-channel counters storm-control multi-threshold 579–580
show interface port-channel counters storm-control multi-threshold broadcast 580
show interface port-channel counters storm-control multi-threshold multicast 580
show interface port-channel counters storm-control multi-threshold unicast 580
show interface switchport 597
show ip access-lists 339–340, 342–343, 371, 373, 375
show ip access-lists summary 344
show ip arp inspection 532
show ip arp inspection interface 528
show ip arp inspection interfaces 532
show ip arp inspection log 532
show ip arp inspection statistics 532
show ip arp inspection vlan 527, 532
show ip dhcp relay 465, 468, 470, 474–476
show ip dhcp relay information trusted-sources 465–467
show ip dhcp snooping binding 543
show ip interface 588
show ip ver source 544
show ip ver source ethernet 544
show ip ver source port-channel 544
show ipv6 access-lists 339–340, 342, 374–375
show ipv6 access-lists summary 344
show ipv6 dhcp relay 478, 480, 482, 485
show ipv6 dhcp relay interface 480
show key chain 559–560, 564–567, 658
show key chain mode decrypt 564–565
show ldap-search-map 134, 142
show ldap-server 128–129, 131–133, 136–137, 142
show ldap-server groups 142
show ldap-server statistics 141–142
show logging ip access-list cache 369, 374
show logging ip access-list status 374
show login 51, 55
show login failures 51
show login on-failure log 38
show login on-successful log 39
show mac access-lists 396, 399–401, 405
show macsec mka session 673
show macsec mka statistics 676
show macsec mka summary 674
show macsec policy 662, 674
show macsec secy statistics 676
show object-group 382–383, 385–386
show policy-map interface control-plane 632

show policy-map type control-plane 632
show port-security 424, 434
show port-security address 430, 435
show port-security address interface 430
show port-security interface 434
show radius-server 67–69, 76–79, 81–83, 274
show radius-server directed-request 76
show radius-server group 274
show radius-server groups 75
show radius-server statistics 87–88
show role 248, 251
show role feature 251
show role feature-group 251
show running-config aaa 56
show running-config acllog 374
show running-config aclmgr 364–365, 374, 386, 402, 405, 411–413, 633
show running-config aclmgr all 374, 405
show running-config all | include max-login 52, 56
show running-config copp 633
show running-config dhcp 457, 459–460, 462–468, 470–471, 473–476, 479–483, 529–532, 542
show running-config interface 372, 579, 597
show running-config interface {ethernet | port-channel} 576, 579
show running-config interface ethernet 403, 487, 592, 597
show running-config interface mgmt 0 487
show running-config interface port-channel 403, 597
show running-config interface vlan 487
show running-config ip 592
show running-config ipv6 592
show running-config ldap 142
show running-config macsec 674
show running-config port-security 425–426, 428, 432–434
show running-config security 158, 172, 251
show running-config security all 154, 172, 251
show running-config tacacs 118
show running-config tacacs all 118
show ssh key 150, 169, 172
show ssh key dsa 172
show ssh key md5 172
show ssh key rsa 172
show ssh server 168, 172
show startup-config aaa 56
show startup-config acllog 374
show startup-config aclmgr 375, 405, 413, 633
show startup-config aclmgr all 375, 405, 413
show startup-config interface ethernet 592
show startup-config ip 592
show startup-config ldap 142
show startup-config security 251
show startup-config tacacs 118
show system login 51
show system login failures 51
show tacacs-server 99–102, 104–105, 107, 109–110, 118
show tacacs-server directed-request 104, 118
show tacacs-server groups 102, 118
show tacacs-server sorted 118
show tacacs-server statistics 117–118
show tacacs+ {pending | pending-diff} 98, 103–105, 109–110, 112
show tacacs+ {status | pending | pending-diff} 118
show telnet server 170, 172
show time-range 389–390
show user-account 152–153, 160, 172, 250–251
show username 156
show username keypair 172
show userpassphrase {length | max-length | min-length} 53, 56
show users 160, 169, 171–172
show vlan access-map 413
show vlan filter 413
ssh 154–155
ssh key 150
ssh key force 150
ssh key rsa 150
ssh login-attempts 153
ssh vrf 154
ssh6 154
ssh6 vrf 154
statistics per-entry 338, 340, 396, 399, 411
storm-control {broadcast | multicast | unicast} 576
storm-control action trap 576, 578
storm-control multi unicast 578
switchport 425–426
switchport block ethernet switchport 597
switchport block port-channel switchport 597
switchport port-security 425
switchport port-security aging time 433
switchport port-security aging type 433
switchport port-security mac-address 427–428
switchport port-security mac-address sticky 426, 429–430
switchport port-security maximum 431
switchport port-security violation 434
system login block-for 50
system login block-for attempts 50
system login block-for within 50
system login quiet-mode access-class 51

T

tacacs-server dead-time 107–108
tacacs-server deadtime 109
tacacs-server directed-request 103
tacacs-server host 54, 98, 100, 102, 104–105, 108
tacacs-server host port 105
tacacs-server host timeout 104
tacacs-server key 54, 99
tacacs-server test 107
tacacs-server test idle-time 107
tacacs-server test username 107
tacacs+ commit 98, 103–105, 109–110, 112
telnet 171
telnet vrf 171
telnet6 171

telnet6 vrf 171
terminal no verify-only 113
terminal no verify-only username 113
terminal verify-only 113
terminal verify-only username 113
test aaa authorization command-type {commands | config-commands}
user command 113
test aaa group 84, 116
test aaa server radius 84
test aaa server radius vrf 84
test aaa server tacacs+ 115
time-range 388

U

udf 361, 397
Unicast RPF 583–585, 587, 592
BGP attributes 584
BOOTP and 585
default settings 587
deploying 585
description 583
DHCP and 585
FIB 583
guidelines 585
implementation 584
limitations 585
Unicast RPF *(continued)*
tunneling and 585
verifying configuration 592
use-vrf 75
user max-logins 51
username 151
username keypair export 156
username keypair export {rsa | dsa} 156
username keypair generate 155
username keypair import 156
username keypair import {rsa | dsa} 156
username password 158
username sshkey 152
username sshkey file bootflash 151
userpassphrase max-length 52
userpassphrase min-length 52

V

vlan access-map 410
vlan filter 413
vrf policy deny 247

W

window-size 661