

Cisco Catalyst IR8340 Rugged Series Router Software Configuration Guide, Cisco IOS XE Release 26.2.x

First Published: 2026-09-28

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 IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

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 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.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

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)

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

Full Cisco Trademarks with Software License ?

CHAPTER 1 **Overview** 1

- Introduction 1
- Accessing the CLI Using a Router Console 2
  - Access the router using the console interface 5
- Initial Bootup Security 6
  - Enforce Changing Default Password 6
  - Telnet and HTTP 7
- Accessing the CLI from a Remote Console 8
  - Preparing to Connect to the Router Console 8
  - Setting Up the IR8340 to Run SSH 9
  - Using Telnet to Access a Console Interface 10
- CLI Session Management 11
  - Information About CLI Session Management 11
  - Changing the CLI Session Timeout 11
  - Locking a CLI Session 12
- Communications, services, and additional information 12
  - Cisco Bug Search Tool 12
  - Documentation feedback 13

CHAPTER 2 **New Features** 15

- New Features for Cisco IOS XE 26.2.1 15

CHAPTER 3 **Using Cisco IOS XE Software** 17

- Understanding Command Modes 17

Keyboard Shortcuts 19
Using the no and default Forms of Commands 19
Using the History Buffer to Recall Commands 20
Managing Configuration Files 20
Saving Configuration Changes 20
Filtering Output from the show and more Commands 21
Finding Support Information for Platforms and Cisco Software Images 22
Using Cisco Feature Navigator 22
Getting Help 22
Finding Command Options: Example 23
Using Software Advisor 28
Using Software Release Notes 28

CHAPTER 4 **Basic Router Configuration** 29
IR8340 Interface Naming 29
Basic Configuration 30
Configuring Global Parameters 35
Configuring the Gigabit Ethernet Interface 36
Support for sub-interface on GigabitEthernet0/0/0 37
Configuring a Loopback Interface 37
Configuring CPU Allocation 38
Enabling Cisco Discovery Protocol 39
Configuring Command-Line Access 39
Configuring Static Routes 41
Configuring Dynamic Routes 43
Configuring Routing Information Protocol 43
Configuring Enhanced Interior Gateway Routing Protocol 44
Modular QoS (MQC) 44

CHAPTER 5 **Configuring Secure Shell** 45
Information About Secure Shell 45
Prerequisites for Configuring Secure Shell 45
Restrictions for Configuring Secure Shell 45
SSH And Router Access 46

SSH Servers, Integrated Clients, and Supported Versions 46
SSH Configuration Guidelines 47
How to Configure Secure Shell 47
Setting Up the IR8340 to Run SSH 47
Configuring the SSH Server 48
Monitoring the SSH Configuration and Status 50
Configuring the Router for Local Authentication and Authorization 50
Information about Secure Copy 51
Prerequisites for Secure Copy 52
Restrictions for Configuring Secure Copy 52
Configuring Secure Copy 52
Additional References 53

CHAPTER 6 **Installing the Software** 55
Installing the Software 55
Cisco Software Licensing 55
Consolidated Packages 55
Network-Essentials 56
Network-Advantage 56
Related Documentation 56
Installing the Cisco IOS XE Release 56
ROMMON Images 57
File Systems 57
Option to Enable or Disable USB Access 58
Autogenerated File Directories and Files 59
Flash Storage 60
Related Documentation 60

CHAPTER 7 **Software Maintenance Upgrade (SMU)** 61
Software Maintenance Upgrade (SMU) 61
SMU Workflow and Basic Requirements 62
SMU Example 62
Installing a Patch Image 62
Uninstalling the Patch Image 64

Uninstalling the Patch Image Using Rollback 64
Uninstalling the Patch Image Using Deactivate, Commit, and Remove 66

CHAPTER 8 **Smart Licensing Using Policy (SLP) 69**
Smart licensing using policy on Cisco routers 69

CHAPTER 9 **Configuring Ethernet Switch Ports 71**
Configuring VLANs 71
Creating a VLAN 72
Configuring LAN Ports for Layer 2 Switching 73
Layer 2 LAN Port Modes 73
Default Layer 2 LAN Interface Configuration 74
Configuring LAN Interfaces for Layer 2 Switching 75
Configuring Private VLANs 75
Information About Private VLANs 75
Primary and Secondary VLANs in Private VLANs 76
Private VLAN Ports 76
Guidelines and Limitations for Private VLANs 76
Configuring a Private VLAN 77
Configuring a VLAN as a Private VLAN 77
Associating Secondary VLANs with a Primary Private VLAN 79
Configuring a Layer 2 Interface as a Private VLAN Host Port 80
Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port 81
Configuring Voice VLANs 83
Limitations and Restrictions 83
How to Configure Voice VLANs 83
Configuring Cisco IP Phone Voice Traffic 83
Removing Voice VLAN 84
Monitoring Voice VLAN 85
Configuring VXLAN Tunneling 85
IEEE 802.1x Protocol 86
Configuring IEEE 802.1X Port-Based Authentication 86
Enabling AAA Authorization for VLAN Assignment 87
Enabling IEEE 802.1X Authentication and Authorization 87

Spanning Tree Protocol Overview 90
Default STP Configuration 91
Enabling STP 91
Configuring Optional STP Features 93
Enabling PortFast 93
Configuring PortFast BPDU Filtering 94
Enabling BPDU Guard 96
Enabling UplinkFast 96
Enabling BackboneFast 97
MAC Table Manipulation 98
Creating a Static Entry in the MAC Address Table 98
MAC Address-Based Traffic Blocking 99
Configuring and Verifying the Aging Timer 100
MAC Learning on a Vlan 100
Assigning IP Addresses to Switch Virtual Interfaces 101
SVI Supported Features 102
IGMP Snooping for IPv4 104
IGMP Filtering and Throttling 104
Default IGMP Filtering and Throttling Configuration 105
Configuring IGMP Profiles 105
Applying IGMP Profiles 106
Setting the Maximum Number of IGMP Groups 107
Configuring the IGMP Throttling Action 108
MLD Snooping 108
MLD Snooping Configuration Guidelines 109
Default MLD Snooping Configuration 109
Enabling or Disabling MLD Snooping on a VLAN 110
Configuring UniDirectional Link Detection 111
Enabling UDLD Globally 111
Enabling UDLD on an Interface 112
Configuring the Switched Port Analyzer 112
SPAN and RSPAN 113
Creating a Local SPAN Session 113
Creating a Local SPAN with Incoming Traffic Allowed on Destination 115

Specifying VLANs to Filter 117
Verifying the SPAN Session 119
Removing a SPAN Session 119
Configuring a VLAN as an RSPAN VLAN 120
Creating an RSPAN Source Session 121
Specifying VLANs to Filter on RSPAN Source Session 123
Creating an RSPAN Destination Session and Configuring Incoming Traffic 124
EtherChannel Overview 126
Channel Groups and Port-Channel Interfaces 127
Port Aggregation Protocol 127
Link Aggregation Control Protocol 127
Configuring Layer 2 EtherChannels 127
Configuring EtherChannel Load-Balancing 129
Configuring the PAgP Learn Method and Priority 130
Configuring the LACP Port Channel Min-Links Feature 131
Configuring LACP Fast Rate Timer 132
Modular Quality of Service Command-Line Interface 133
Creating a Traffic Class 134
Creating a Traffic Policy 136
Configuring Class-Based Packet Marking 139
Attaching a Traffic Policy to an Interface 144
Classifying, Policing, and Marking Traffic on Physical Ports by Using Policy Maps 145

CHAPTER 10 **Configuring Switchport Blocking** 149
About Switchport Blocking 149
Configuring Switchport Blocking 149

CHAPTER 11 **Configuring Storm Control** 153
Information About Storm Control 153
Configuring Storm Control 154

CHAPTER 12 **Configuring MAC Address Notification** 157
MAC Address Notification 157
Configuring MAC Address Notification 157

CHAPTER 13 **Configuring Q-in-Q and Layer 2 Protocol Tunneling** 159

Information About Q-in-Q Tunnels 159

Information About Layer 2 Protocol Tunneling 160

Configuring VLAN Mapping for Selective Q-in-Q on a 802.1Q Tunnel Port 160

Enabling the Layer 2 Protocol Tunnel 162

Configuring Thresholds for Layer 2 Protocol Tunnel Ports 163

Verifying the Q-in-Q Configuration 164

VLAN Translation One-to-One Mapping 164

CHAPTER 14 **Layer 2 Tunneling Protocol Version 3** 165

Layer 2 Tunneling Protocol Version 3 165

Prerequisites for Layer 2 Tunneling Protocol Version 3 165

Restrictions for Layer 2 Tunneling Protocol Version 3 166

General L2TPv3 Restrictions 166

VLAN-Specific Restrictions 167

IPv6 Protocol Demultiplexing for L2TPv3 Restrictions 167

L2TPv3 Control Message Hashing Restrictions 167

L2TPv3 Digest Secret Graceful Switchover Restrictions 167

Quality of Service Restrictions in L2TPv3 Tunneling 168

Information About Layer 2 Tunneling Protocol Version 3 168

L2TPv3 Header Description 168

L2TPv3 Operation 169

L2TPv3 Features 171

Ethernet over L2TPv3 173

GEC over L2TPv3 174

Sequencing 174

L2TPv3 Type of Service Marking 175

Keepalive 175

MTU Handling 175

L2TPv3 Control Message Hashing 176

L2TPv3 Control Message Rate Limiting 177

L2TPv3 Digest Secret Graceful Switchover 177

L2TPv3 Pseudowire 177

Manual Clearing of L2TPv3 Tunnels 178
L2TPv3 Tunnel Management 178
L2TPv3 Protocol Demultiplexing 178
L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations 178
HDLC over L2TPv3 179
L2TPv3 Benefits 179
Supported L2TPv3 Payloads 179
Ethernet 179
VLAN 180
IPv6 Protocol Demultiplexing 180
Performance Impact of L2TPv3 on Cisco ASR 1000 Series Routers 181
Layer 2 Protocol Tunneling and Forwarding 182
How to Configure Layer 2 Tunneling Protocol Version 3 182
Configuring L2TP Control Channel Parameters 182
Configuring L2TP Control Channel Timing Parameters 182
Configuring L2TPv3 Control Channel Authentication Parameters 184
Configuring L2TP Control Channel Maintenance Parameters 189
Configuring the L2TPv3 Pseudowire 190
Configuring the Xconnect Attachment Circuit 193
Configure L2TPv3 on a Switched Virtual Interface 195
Manually Configuring L2TPv3 Session Parameters 197
Configuring Protocol Demultiplexing for L2TPv3 199
Configuring an L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations 200
Configuring GEC over L2TPv3 201
Configuring GEC with Dot1Q 203
Configuring GEC with QinQ 204
Manually Clearing L2TPv3 Tunnels 205
Configuration Examples for Layer 2 Tunneling Protocol Version 3 206
Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport 211
Example: Configuring GEC over L2TPv3 212
Example: Configuring GEC with Dot1q over L2TPv3 212
Example: Configuring GEC with QinQ over L2TPv3 212
Additional References 213
Glossary 214

CHAPTER 15 **Power over Ethernet** 217

Power over Ethernet 217

Device Detection and Power Allocation 218

Configuring Power Over Ethernet 218

Configuring Universal PoE 219

PoE Debug Commands 220

CHAPTER 16 **Configuring the T1/E1 Network Interface Module** 221

Information About T1/E1 Network Interface Module 221

Configuring T1/E1 Network Interface Module 221

Configuring the Card Type 221

Changing the Card Type 222

Configuring the T1/E1 Network Interface Module for Data Support 223

Example of T1/E1 Network Interface Module Configuration 226

CHAPTER 17 **Configuring the Serial Network Interface Module** 229

About the Asynchronous/Synchronous Serial Network Interface Module 229

Configuring the Serial Interface to Sync or Async Mode 229

Configuring Synchronous Serial Ports 230

Checking DCE and DTE Cable Type 230

Specifying Synchronous Serial Encapsulation 230

DCE and DTE Configuration for HDLC Encapsulation 231

DCE and DTE Configuration for PPP (CHAP) Encapsulation 232

DCE and DTE Configuration for PPP (PAP) Encapsulation 233

DCE and DTE Configuration for Frame Relay Encapsulation 234

Serial Synchronous Show and Debug Commands 236

Configuring Asynchronous Serial Ports 237

Specifying Asynchronous Serial Encapsulation 237

CHAPTER 18 **Cellular pluggable interface modules** 239

CHAPTER 19 **Information About SCADA** 247

SCADA Overview 247
Role of the IR8340 247
Key Terms 248
Protocol Translation Application 248
Prerequisites 249
Guidelines and Limitations 250
Default Settings 250
Configuring Protocol Translation 250
Enabling the IR8340 Serial Port and SCADA Encapsulation 250
EXAMPLE 251
Configuring T101 and T104 Protocol Stacks 251
Prerequisites 251
Configuring the T101 Protocol Stack 252
EXAMPLE 254
Configuring the T104 Protocol Stack 254
EXAMPLE 257
Configuration Example 258
Configuring the DNP3 Protocol Stacks 259
Configuring DNP3 Serial 259
EXAMPLE 261
Configuring DNP3 IP 261
EXAMPLE 263
Starting and Stopping the Protocol Translation Engine 263
EXAMPLE 264
Verifying Configuration 264
Debug Commands 265

CHAPTER 20 **Raw Socket Transport** 267
Information About Raw Socket Transport 267
TCP Transport 268
UDP Transport 268
Serial Data Processing 269
VRF-Aware Raw Socket 269
Prerequisites 270

Guidelines and Limitations 270
Default Settings 270
Configuring Raw Socket Transport 270
Enabling Raw Socket Transport on the Serial Interface 270
Configuring Common Raw Socket Line Options 271
Configuring Raw Socket TCP 272
Configuring the Raw Socket TCP Server 273
Configuring the Raw Socket TCP Client 274
Configuring a Raw Socket UDP Peer-to-Peer Connection 275
Verifying Configuration 276
Configuration Example 277
Raw Socket TCP 277
Raw Socket UDP 278
Raw Socket VRF 279
Show Line Details for Configuring Raw-TCP/UDP 280
Raw-Socket Show and Debug Commands 281

CHAPTER 21 **Serial MPLS Pseudowire** 283
Serial MPLS pseudowire on IR8340 283
Data exchange between substation RTU and SCADA controller using MPLS pseudowire 284
Serial MPLS pseudowire deployment scenarios 285
Serial to serial MPLS pseudowire deployment configuration using CLI 286
Serial to raw socket (IP) MPLS pseudowire deployment configuration using CLI 288
Raw socket configuration using CLI 289
Verify serial to serial configurations using CLI 290
Verify Serial to raw socket (IP) configurations using CLI 292
Limitations and restrictions of current serial MPLS pseudowire release 294

CHAPTER 22 **Configuring MODBUS TCP** 295
Understanding MODBUS TCP 295
MODBUS and Security 295
Multiple Request Messages 296
Configuring the Router as the MODBUS TCP Server 297
Defaults 297

Enabling MODBUS TCP on the Switch 297
MODBUS TCP Registers 298
System Information Registers 298
Port Information Registers 299
Interpreting the Port State 309
NERC CIP compliance 309

CHAPTER 23 vCPU and RAM Distribution 313
Introduction 313
Distribution of vCPU and RAM Resources for Cisco IOx Applications 313
Higher CPU and RAM Allocation for IOx Applications 314
Configure Data Plane Heavy Template 314
Verify the Active vCPU and RAM Distribution 314
Configure Service Plane Heavy Template 315
Verify Service Plane Heavy 316

CHAPTER 24 VLAN Access Control Lists 317
Information About VLAN Access Control Lists 317
VLAN Maps 317
Configuring VACLs 317
Defining a VLAN Access Map 318
Configuring a Match Clause in a VLAN Access Map Sequence 318
Configuring an Action Clause in a VLAN Access Map Sequence 319
Applying a VLAN Access Map 319
Verifying VLAN Access Map Configuration 319
Debugging VACLs 319

CHAPTER 25 Configuring MACsec 321
MACsec Encryption Overview 321
Limitations and Restrictions 321
Media Access Control Security and MACsec Key Agreement 322
Configuring MACsec Encryption 323
Configuring MKA and MACsec 323
Configuring an MKA Policy 323

Configuring MACsec MKA using PSK 324
Configuring MACsec MKA on an Interface using PSK 325
Example: Sample Configuration of Switch-to-Switch MACsec 326
Example: Sample Configuration of Switch-to-Host MACsec 327
Verifying the Configuration 329

CHAPTER 26 **WAN MACSEC and MKA Support Enhancements** 333
MACsec and MKA Overview 333
Benefits of WAN MACsec and MKA Support Enhancements 334
Best Practices for Implementing WAN MACsec and MKA Support Enhancements 334
Limitations in supported configurations 335
MKA Policy Inheritance 335
Key Lifetime and Hitless Key Rollover 335
Encryption Algorithms for Protocol Packets 335
Access Control Option for Smoother Migration 336
Extensible Authentication Protocol over LAN Destination Address 337
Replay Protection Window Size 337
How to Configure WAN MACsec and MKA Support Enhancements 338
Configuring MKA 338
Configuring MKA Pre-Shared Key 339
Configuring an Option to Change the EAPoL Ethernet Type 340
Sample Configuration for Point-to-Point WAN MACsec 341
Example: Port Based WAN MACsec 341
Example: VLAN Based WAN MACsec 341
Sample Show Command Output for Port Based WAN MACsec 342
Sample Show Command Output for VLAN Based WAN MACsec 344

CHAPTER 27 **Configuring IPv6 First Hop Security** 347
IPv6 First Hop Security Overview 347
Overview of DHCPv6 Guard 347
Restrictions of DHCPv6 Guard 347
Overview of IPv6 RA Guard 347
Limitations and Restrictions of IPv6 RA Guard 348
Configuring an IPv6 DHCP Guard Policy 348

Attaching an IPv6 DHCP Guard Policy to an Interface or a VLAN 349
Configuring an IPv6 Router Advertisement Guard Policy 350
Attaching an IPv6 Router Advertisement Guard Policy to an Interface or a VLAN 352

CHAPTER 28 **Configuring IP Device Tracking** 355
Information About IP Device Tracking 355
Overview of SISF-Based Device Tracking 356
Options to Enable SISF-Based Device Tracking 356
Manually Enabling SISF-Based Device Tracking 356
Programmatically Enabling SISF-Based Device Tracking 357
How to Configure SISF-Based Device Tracking 357
Manually Enabling SISF-Based Device Tracking 357
Applying the Default Device Tracking Policy to a Target 357
Creating a Custom Device Tracking Policy with Custom Settings 358
Attaching a Device Tracking Policy to an Interface 362
Attaching a Device Tracking Policy to a VLAN 363
Configuring a Multi-Switch Network to Stop Creating Binding Entries from a Trunk Port 364
Enabling SISF Syslogs 365
Example: DHCP Snooping Auto Enabling DT PROGRAMMATIC Policy 365

CHAPTER 29 **Configuring Security for VPNs with IPsec** 367
Information About Configuring Security for VPNs with IPsec 367
Supported Standards 367
Supported Encapsulation 369
IPsec Functionality Overview 369
IKEv1 Transform Sets 370
IKEv2 Transform Sets 370
How to Configure IPsec VPNs 370
Creating Crypto Access Lists 370
Configuring Transform Sets for IKEv1 and IKEv2 Proposals 371
Configuring Transform Sets for IKEv1 372
Configuring Transform Sets for IKEv2 373
Creating Crypto Map Sets 375
Creating Static Crypto Maps 375

Creating Dynamic Crypto Maps 378
Creating Crypto Map Entries to Establish Manual SAs 381
Applying Crypto Map Sets to Interfaces 384

CHAPTER 30 Configuring High-availability Seamless Redundancy (HSR) 385
Information About HSR 385
Loop Avoidance 386
HSR RedBox Modes of Operation 387
HSR-SAN Mode 387
HSR-SAN Interfaces 388
Configuring an HSR Ring 388
Configuring Interface Sub-Mode 388
Clearing All Node Table and VDAN Table Dynamic Entries 389
Verifying Configuration 389

CHAPTER 31 Configuring Parallel Redundancy Protocol (PRP) 391
Information About PRP 391
PRP Channels 392
Creating a PRP Channel and Group 392
Clearing All Node Table and VDAN Table Dynamic Entries 394
Disabling the PRP Channel and Group 394
PRP Mode LED 395
Verifying Configuration 395

CHAPTER 32 Configuring Resilient Ethernet Protocol (REP) 397

CHAPTER 33 System Messages 399
Process management systems 399
Error message details 399

CHAPTER 34 Environmental Monitoring 405
Environmental monitoring 405
Environmental Monitoring and Reporting Functions 405

Environmental Monitoring Functions 406
Environmental Reporting Functions 407
Additional References 407
Technical assistance 407

CHAPTER 35 **IOx Application Hosting 409**
Application Hosting 409
Application hosting 409
Application hosting 409
IOx 410
Cisco application hosting 410
IOXMAN 411
Application hosting on the IR8340 router 411
Application hosting on Layer 2 and Layer 3 interfaces 412
VirtualPortGroup 413
vNICs 413
Application hosting configuration 414
Enable IOx 414
Configure application hosting to Layer 2 interfaces 415
Configure a VirtualPortGroup to a Layer 3 data port 416
Configure Docker run time options 418
Install and uninstall apps 419
Override the app resource configuration 420
Override the app resource configuration 421
Configure IOx with ERSPAN 422
Configuration examples for application hosting 423
Example: Enable IOx 423
VirtualPortGroup to Layer 3 data port configuration example 423
App installation and uninstallation examples 424
App resource configuration override example 424
Signed application support 424
Cisco Cyber Vision 425
Cisco ThousandEyes Enterprise Agent 425

CHAPTER 36 **ROM Monitor Overview** 427

ROM Monitor 427

Access ROM Monitor mode 428

Check the Current ROMMON Version 428

ROM Monitor commands 430

Examples 430

Change the ROM Monitor prompt 431

Display the Configuration Register Setting 431

Environment variable settings 432

Frequently used environmental variables 432

Display environment variable settings 432

Environment variable settings 433

Save Environment Variable settings 433

Exit ROM Monitor Mode 434

Configuration example 434

Upgrade the ROMmon for a Router 435

CHAPTER 37 **Process Health Monitoring** 437

Monitoring Control Plane Resources 437

Avoiding Problems Through Regular Monitoring 437

Cisco IOS Process Resources 437

Overall Control Plane Resources 447

Monitoring Hardware Using Alarms 449

Router Design and Monitoring Hardware 449

BootFlash Disk Monitoring 449

Approaches for Monitoring Hardware Alarms 449

Viewing the Console or Syslog for Alarm Messages 449

Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP 450

CHAPTER 38 **Device Sensors** 451

Restrictions for device sensor configuration 452

Information about device sensors 452

Device sensors 452
Device sensor configuration 454
Enable accounting augmentation 454
Configure protocol attributes in access and accounting requests 455
Create a protocol filter 456
Apply a protocol filter to the sensor output 457
Track TLV changes 458
Device sensor configuration verification commands 459
Configuration examples of device sensor 461
Device sensor configuration examples 461

CHAPTER 39 DHCP Option 82 463
DHCP Option 82 463
Prerequisites 464
Key considerations 465
Limitations 465
DHCP Option 82 packet flow 465
Hexadecimal pattern and suboption decoding 466
Configure DHCP Option 82 on server 467
Configure DHCP address pool 469
Configure DHCP relay agent 470
Configure relay chaining on relay agents 472
Verify DHCP Option 82 473
Debug DHCP Option 82 logs 473

CHAPTER 40 Segment Routing over MPLS 475
Segment Routing over MPLS 475
Prerequisites 476
Configure Segment Routing over MPLS 476
Configure Segment Routing MPLS on OSPF network 477
Configure Prefix-SID for OSPF 478
Verify SR-MPLS configuration 479

CHAPTER 41 SGT Inline Tagging on LAN Ports 481

SGT Inline Tagging support 481
Prerequisites 482
Limitations 482
Licensing 482
Configure SGT Inline Tagging 482
Scenarios for Troubleshooting 483
Verify CTS 483
Debug commands for CTS 484

CHAPTER 42 **IKEv2 and IPsec PFS Post Quantum Cryptography 487**
IKEv2 and IPsec PFS Post Quantum Cryptography 487
Benefits of PQC 488
Prerequisites 488
Limitations 488
Phased migration in Hub-and-Spoke deployments 489
ML-KEM algorithm and parameter sets 489
PFS negotiation behavior 490
Configure an IKEv2 proposal for PQC 490
Configure Perfect Forward Secrecy with PQC 492
Verify the PQC hybrid key exchange 493
Troubleshooting PQC hybrid key exchange 494

CHAPTER 43 **Post Quantum Cryptography for SSHv2 sessions 495**
Implementation of Post Quantum Cryptography 495
Overview 496
Factors to consider for deploying PQC 496
Protocol requirements 496
Restrictions for PQC 496
PQC negotiation scenarios 497
Configure PQC for SSHv2 sessions 497
Verify PQC for SSHv2 sessions 498
Key elements of PQC 498
Supported platforms 500

CHAPTER 44 **Cisco Network Plug and Play Agent** 501

Prerequisites for Cisco Network Plug and Play Agent 501

Restrictions for Cisco Network Plug and Play Agent 502

Information About Cisco Network Plug and Play Agent 502

Cisco Network Plug and Play Deployment Solution 502

Cisco Network Plug and Play Features 503

Cisco Network Plug and Play Agent Services and Capabilities 504

Software Maintenance Upgrade 506

Cisco Network Plug and Play Agent 507

Cisco Network Plug and Play Server 507

Cisco Network Plug and Play Agent Deployment 508

Cisco Network Plug and Play Agent Network Topology 509

Cisco Network Plug and Play Agent Initialization 509

Cisco Network Plug and Play Agent Deployment Solutions 510

Cisco Network Plug and Play Agent Discovery Process 510

Plug-n-Play Agent Deployment Protocol 515

Plug-n-Play Agent Security 516

Security Methods for the PnP Discovery Process 517

Self-Signed Certificate Based Authentication 517

Mobile Device Based Secured Installation 517

CA-Signed Certificate based Authentication 518

DHCP Option based Discovery over an IPv4 Network 518

DHCP Option based Discovery over an IPv6 Network 519

DNS-based Discovery 520

DNS-based Discovery over an IPv6 Network 521

Cisco Cloud Redirection over an IPv4 and IPv6 Network 521

Cisco Network PnP Discovery Over 4G Interface 522

Cisco Network PnP Discovery over a Management Interface 523

Cisco PnP over an EtherChannel 523

Security Methods for Post-PnP Discovery Process 523

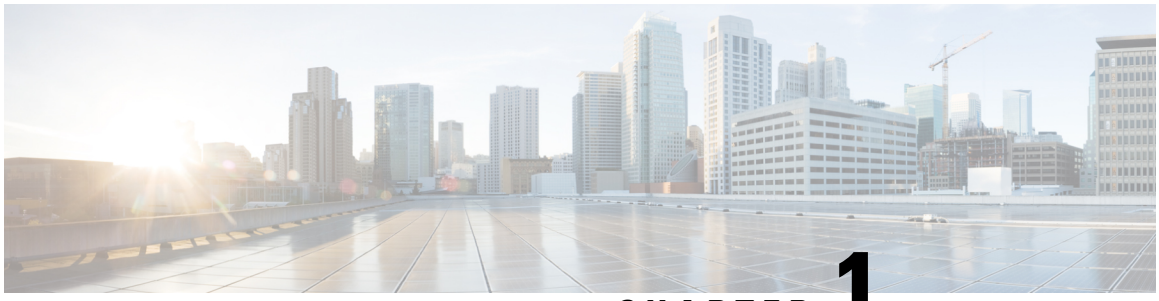
Certificate Install Service 523

SUDI-based PnP Application Level Authentication 524

SUDI-based Client Certificate Validation 524

SUDI-based Serial Number 524
SUDI-Based Device Authentication 524
How to Configure Cisco Network Plug and Play Agent 526
Configuring Cisco Network Plug and Play Agent Profile 526
Configuring Network Plug and Play Agent Device 527
Configuring Cisco Network Plug and Play Reconnect Factor 527
Configuring Cisco Network Plug and Play HTTP Transport Profile 528
Configuring Cisco Network Plug and Play HTTPS Transport Profile 530
Configuring Backup Cisco Network Plug and Play Device 531
Configuring Backup Cisco Network Plug and Play Reconnect Factor 532
Configuring Backup Cisco Network Plug and Play HTTP Transport Profile 533
Configuring Backup Cisco Network Plug and Play HTTPS Transport Profile 534
Configuring Cisco Network Plug and Play Agent Tag 536
Troubleshooting and Debugging 537
Glossary 538
Additional References for Open Plug-n-Play Agent 538

CHAPTER 45 **Troubleshooting** 541
Troubleshooting 541
Understanding Diagnostic Mode 541
Before Contacting Cisco or Your Reseller 542
show interfaces Troubleshooting Command 542
Software Upgrade Methods 542
Change the Configuration Register 543
Configuring the Configuration Register for Autoboot 544
Reset the Router 544
Recovering a Lost Password 545
Reset the Configuration Register Value 546
Configuring a console port transport map 547
Viewing Console Port, SSH, and Telnet Handling Configurations 548
Using the factory reset Commands 550



CHAPTER 1

Overview

- Introduction, on page 1
- Accessing the CLI Using a Router Console, on page 2
- Initial Bootup Security, on page 6
- Accessing the CLI from a Remote Console , on page 8
- CLI Session Management, on page 11
- Communications, services, and additional information, on page 12

Introduction

The Cisco Catalyst IR8340 Rugged Series Router is Cisco's first industrial-grade fully integrated routing and switching platform. IR8340 is designed to provide outstanding flexibility and adaptability to address the latest needs of the network evolution.

Note The terms *IR8340* and *router* are used throughout this document in text and CLI examples to refer to the Cisco Catalyst IR8340 Rugged Series Router, unless otherwise noted.

Note The documentation set for this product strives to use bias-free language. For 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.

The IR8340 Router features 2 Pluggable Interface Module (PIM) slots, 2 single-wide IRM-NIM slots, 1 timing module slot, plus 12 onboard LAN ports and 2 WAN ports, and supports the following:

- 150W/250W PSU, Low Voltage DC and High Voltage AC/DC options
- Timing and SyncE support–IRIG-B (Analog & Digital), GNSS, and PTP

Note For more information about timing related configuration, see *Timing and Synchronization Configuration Guide, Cisco IOS XE 17 (Cisco Catalyst IR8340 Rugged Series Router)*.

- LTE PIM modules
- Network Interface Modules (NIMs)
- mSATA module
- 2 x 1G Combo WAN ports
- 4 x 1G Copper LAN Ports
- 4 x 1G Combo LAN ports
- 4 x 1G SFP LAN Ports
- PoE/PoE+/UPoE (up to 120w) support on LAN port 1-4
- 2 x IN and 1 x OUT Alarm ports (RJ45)

Accessing the CLI Using a Router Console

Cisco IR8340 routers have an RJ45 RS232 serial console port located on the router front panel. The default baud rate is 9600. You can use an RJ45 Roll Over console cable that is available in the market.

On a device fresh from the factory, you are greeted with a System Configuration Dialog. If the router was ordered for the use of Cisco PnP connect services, in the case of centralized provisioning, the router skips the initial dialog. The following is an example, and names and IP addresses are shown as examples.

Note Autoinstall will terminate if any input is detected on console.

```
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]:

WARNING: ** NOTICE ** This is the final IOS XE release to provide support for the H.323
protocol. Consider switching to SIP for multimedia applications before upgrading to 17.7.1.
*Jan 27 23:51:55.579: %TAMPER_ALARM-0-TAMPER_ALARM_ASSERT: Tamper alarm slot (Tamper alarm
 slot 2) asserted

*Jan 27 23:51:55.579: %TAMPER_ALARM-0-TAMPER_ALARM_ASSERT: Tamper alarm slot (Tamper alarm
 slot 3) asserted

Autoinstall trying DHCPv6 on GigabitEthernet0/0/0,GigabitEthernet0/0/1

Autoinstall trying DHCPv4 on GigabitEthernet0/0/0,GigabitEthernet0/0/1

AUTO IP is starting!!!!

start Autoip process
```

```
Acquired IPv4 address 192.168.0.202 on Interface GigabitEthernet0/0/0
Received following DHCPv4 options:
dns-server-ip : 192.168.0.2
si-addr : 192.168.0.2
hostname : Router

stop Autoip process


Press RETURN to get started!


*Jan 27 23:53:08.903: %SYS-5-USERLOG_NOTICE: Message from tty0(user id: ): Device in day0
workflow, some non user-configured options may be enabled by default
*Jan 27 23:53:08.920: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
OK to enter CLI now...

pnp-discovery can be monitored without entering enable mode

Entering enable mode will stop pnp-discovery

*Jan 27 23:53:08.921: %PNP-6-HTTP_CONNECTING: PnP Discovery trying to connect to PnP server
 (https://devicehelper.cisco.com.:443/pnp/HELLO)
*Jan 27 23:53:09.788: AUTOINSTALL: Obtain siaddr 192.168.0.2 (as config server)
*Jan 27 23:53:09.788: AUTOINSTALL: Setting hostname Router from DHCP reply
*Jan 27 23:53:10.899: %LINK-5-CHANGED: Interface GigabitEthernet0/0/1, changed state to
administratively down
*Jan 27 23:53:11.899: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/1,
 changed state to down
*Jan 27 23:53:29.880: %PNP-6-HTTP_CONNECTED: PnP Discovery connected to PnP server
(https://devicehelper.cisco.com.:443/pnp/HELLO)
*Jan 27 23:53:29.883: %PNP-6-PNP_PROFILE_CREATED: PnP profile (pnp_cco_profile) created
(1/3) by (pid=656, pname=PnP Agent Discovery, time=23:53:29 UTC Wed Jan 27 2021)
*Jan 27 23:53:30.893: %PNP-6-PNP_SUDI_UPDATE: Device SUDI [PID:IR8340-K9,SN:FDO2523J7GF]
identified
*Jan 27 23:53:30.893: %PNP-6-PNP_RELOAD_INFO_ENCODED: Reload reason (PnP Service Info
2408-Unknown reason) encoded (1/3) by (pid=656, pname=PnP Agent Discovery)
*Jan 27 23:53:30.894: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 27 23:53:35.635: %PNP-6-PNP_RELOAD_INFO_STOPPED: Reload reason (PnP Service Info
2408-Unknown reason) stopped by (profile=pnp_cco_profile, host=devicehelper.cisco.com.,
port=443)
*Jan 27 23:53:56.755: %PNP-6-PNP_BACKOFF_NOW: PnP Backoff now for (60) seconds requested
(1/3) by (profile=pnp_cco_profile, host=devicehelper.cisco.com., port=443)
*Jan 27 23:54:07.900: %PNP-3-PNP_CCO_PROFILE_UNCONFIGURED: CCO Server profile
(pnp_cco_profile) unconfigured (1/10) by (pid=656, pname=PnP Agent Discovery, time=23:54:07
 UTC Wed Jan 27 2021)
*Jan 27 23:54:07.900: %PNP-6-PNP_PROFILE_DELETED: PnP profile (pnp_cco_profile) deleted
(1/3) by (pid=656, pname=PnP Agent Discovery, time=23:54:07 UTC Wed Jan 27 2021)
*Jan 27 23:54:07.901: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 27 23:54:07.909: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 27 23:54:13.907: %PNP-3-PNP_DOMAIN_NAME_NOT_FOUND: Domain name not found (4/10) on
(GigabitEthernet0/0/0) by (pid=656, pname=PnP Agent Discovery, time=23:54:13 UTC Wed Jan
27 2021)
*Jan 27 23:54:13.907: %PNP-3-PNP_DOMAIN_NAME_NOT_FOUND: Domain name not found (5/10) on
(WPAN0/1/0) by (pid=656, pname=PnP Agent Discovery, time=23:54:13 UTC Wed Jan 27 2021)
*Jan 27 23:54:29.911: %PNP-3-PNP_DOMAIN_NAME_NOT_FOUND: Domain name not found (6/10) on
(GigabitEthernet0/0/0) by (pid=656, pname=PnP Agent Discovery, time=23:54:29 UTC Wed Jan
27 2021)
*Jan 27 23:54:29.911: %PNP-3-PNP_DOMAIN_NAME_NOT_FOUND: Domain name not found (7/10) on
(WPAN0/1/0) by (pid=656, pname=PnP Agent Discovery, time=23:54:29 UTC Wed Jan 27 2021)
```

```
*Jan 27 23:54:37.911: %PNP-3-PNP_DOMAIN_NAME_NOT_FOUND: Domain name not found (8/10) on
(GigabitEthernet0/0/0) by (pid=656, pname=PnP Agent Discovery, time=23:54:37 UTC Wed Jan
27 2021)
*Jan 27 23:54:37.911: %PNP-3-PNP_DOMAIN_NAME_NOT_FOUND: Domain name not found (9/10) on
(WPAN0/1/0) by (pid=656, pname=PnP Agent Discovery, time=23:54:37 UTC Wed Jan 27 2021)
*Jan 27 23:54:53.914: %PNP-3-PNP_DOMAIN_NAME_NOT_FOUND: Domain name not found (10/10) on
(GigabitEthernet0/0/0) by (pid=656, pname=PnP Agent Discovery, time=23:54:53 UTC Wed Jan
27 2021)
*Jan 27 23:55:20.100: %PNP-6-PNP_CCO_SERVER_IP_RESOLVED: CCO server (devicehelper.cisco.com.)
 resolved to ip (18.205.166.131) by (pid=656, pname=PnP Agent Discovery, time=23:55:20 UTC
 Wed Jan 27 2021)
*Jan 27 23:55:20.100: %PNP-6-PNP_PROFILE_CREATED: PnP profile (pnp_cco_profile) created
(2/3) by (pid=656, pname=PnP Agent Discovery, time=23:55:20 UTC Wed Jan 27 2021)
*Jan 27 23:55:21.107: %PNP-6-PNP_RELOAD_INFO_ENCODED: Reload reason (PnP Service Info
2408-Unknown reason) encoded (2/3) by (pid=656, pname=PnP Agent Discovery)
*Jan 27 23:55:21.108: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 27 23:55:32.751: %PNP-6-PNP_BACKOFF_NOW: PnP Backoff now for (60) seconds requested
(2/3) by (profile=pnp_cco_profile, host=devicehelper.cisco.com., port=443)
Autoinstall trying DHCPv6 on GigabitEthernet0/0/0

*Jan 27 23:55:43.108: %PNP-3-PNP_CCO_PROFILE_UNCONFIGURED: CCO Server profile
(pnp_cco_profile) unconfigured (2/10) by (pid=656, pname=PnP Agent Discovery, time=23:55:43
 UTC Wed Jan 27 2021)
*Jan 27 23:55:43.108: %PNP-6-PNP_PROFILE_DELETED: PnP profile (pnp_cco_profile) deleted
(2/3) by (pid=656, pname=PnP Agent Discovery, time=23:55:43 UTC Wed Jan 27 2021)
*Jan 27 23:55:43.109: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 27 23:55:43.113: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
Autoinstall trying DHCPv6 on GigabitEthernet0/0/0

Autoinstall trying DHCPv6 on GigabitEthernet0/0/0

Autoinstall trying DHCPv6 on GigabitEthernet0/0/0

Autoinstall trying DHCPv6 on GigabitEthernet0/0/0

Autoinstall trying DHCPv6 on GigabitEthernet0/0/0

*Jan 27 23:56:55.316: %PNP-6-PNP_PROFILE_CREATED: PnP profile (pnp_cco_profile) created
(3/3) by (pid=656, pname=PnP Agent Discovery, time=23:56:55 UTC Wed Jan 27 2021)
*Jan 27 23:56:56.323: %PNP-6-PNP_RELOAD_INFO_ENCODED: Reload reason (PnP Service Info
2408-Unknown reason) encoded (3/3) by (pid=656, pname=PnP Agent Discovery)
*Jan 27 23:56:56.324: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
Autoinstall trying DHCPv6 on GigabitEthernet0/0/0

*Jan 27 23:57:09.810: AUTOINSTALL: script execution not successful for Gi0/0/0.
*Jan 27 23:57:10.829: %SYS-5-CONFIG_P: Configured programmatically by process DHCP Autoinstall
 from console as vty0
*Jan 27 23:58:10.003: %PNP-6-PNP_BACKOFF_NOW: PnP Backoff now for (60) seconds requested
(3/3) by (profile=pnp_cco_profile, host=devicehelper.cisco.com., port=443)
*Jan 27 23:58:21.323: %PNP-3-PNP_CCO_PROFILE_UNCONFIGURED: CCO Server profile
(pnp_cco_profile) unconfigured (3/10) by (pid=656, pname=PnP Agent Discovery, time=23:58:21
 UTC Wed Jan 27 2021)
*Jan 27 23:58:21.323: %PNP-6-PNP_PROFILE_DELETED: PnP profile (pnp_cco_profile) deleted
(3/3) by (pid=656, pname=PnP Agent Discovery, time=23:58:21 UTC Wed Jan 27 2021)
*Jan 27 23:58:21.324: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 27 23:58:21.327: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 27 23:59:34.507: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
```

```
*Jan 27 23:59:59.507: %PNP-3-PNP_CCO_PROFILE_UNCONFIGURED: CCO Server profile
(pnp_cco_profile) unconfigured (4/10) by (pid=656, pname=PnP Agent Discovery, time=23:59:59
 UTC Wed Jan 27 2021)
*Jan 27 23:59:59.508: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 27 23:59:59.511: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 28 00:01:12.715: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 28 00:02:22.715: %PNP-3-PNP_CCO_PROFILE_UNCONFIGURED: CCO Server profile
(pnp_cco_profile) unconfigured (5/10) by (pid=656, pname=PnP Agent Discovery, time=00:02:22
 UTC Thu Jan 28 2021)
*Jan 28 00:02:22.716: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 28 00:02:22.719: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
Router>en
Router#sh ip in
*Jan 28 00:02:42.724: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as console
*Jan 28 00:02:42.724: %PNP-6-PNP_SAVING_TECH_SUMMARY: Saving PnP tech summary
(/pnp-tech/pnp-tech-discovery-summary)... Please wait. Do not interrupt.t b
*Jan 28 00:02:42.877: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 28 00:02:42.924: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 28 00:02:43.394: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0
*Jan 28 00:02:43.494: %PNP-6-PNP_TECH_SUMMARY_SAVED_OK: PnP tech summary
(/pnp-tech/pnp-tech-discovery-summary) saved successfully (elapsed time: 1 seconds).
*Jan 28 00:02:43.494: %PNP-6-PNP_DISCOVERY_STOPPED: PnP Discovery stopped (Config Wizard)
Interface IP-Address OK? Method Status Protocol
GigabitEthernet0/0/0 192.168.0.202 YES DHCP up up
GigabitEthernet0/0/1 unassigned YES unset administratively down down
WPAN0/1/0 unassigned YES unset up up
Router#
```

The device now has a basic configuration that you can build upon.

Access the router using the console interface

Procedure

Step 1 Use the **enable** command to enable the router session.

Example:

```
Router > enable
```

Step 2 (Go to Step 3 if the enable password has not been configured.) Enter your system password when prompted for the password.

Example:

```
Password: enablepass
```

Once your password is accepted, the privileged EXEC mode prompt is displayed.

Example:

```
Router#
```

You now have access to the CLI in privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.

Step 3 Enter the **quit** command to exit the console session.

Example:

```
Router > quit
```

Initial Bootup Security

This section contains the following:

Enforce Changing Default Password

When the device is first booted after factory reset or fresh from the factory, the following prompt is received on the console:

Would you like to enter the initial configuration dialog? [yes/no]:

In previous documentation, Cisco recommended using the **enable secret** command instead of the **enable password** command because this offers an improved encryption algorithm.

The initial dialog forces setting a new enable password, and also using the **enable secret** command instead. The following is an example:

```
Would you like to enter the initial configuration dialog? [yes/no]: no

Autoinstall trying DHCP on GigabitEthernet0/0/0
Autoinstall trying DHCPv6 on GigabitEthernet0/0/0

The enable secret is a password used to protect
access to privileged EXEC and configuration modes.
This password, after entered, becomes encrypted in
the configuration.
------------------------------------------------
secret should be of minimum 10 characters with
at least 1 upper case, 1 lower case, 1 digit and
should not contain [cisco]
------------------------------------------------
Enter enable secret: **********
Confirm enable secret: **********

The following configuration command script was created:

enable secret 9 $9$rDzH3rLqjlFhek$G9UDZE7moWqsKJEZfJAH2yO.SPhKZeKJsEe./CPEzl.
!
end


[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.
```

```
Enter your selection [2]: 2
Building configuration...
[OK]
Use the enabled mode 'configure' command to modify this configuration.



Press RETURN to get started!
*Feb 12 00:14:14.305: %LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to
administratively down
*Feb 12 00:14:14.308: %LINK-5-CHANGED: Interface GigabitEthernet0/0/1, changed state to
administratively down
*Feb 12 00:14:15.306: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0,
 changed state to down
Router>
*Feb 12 00:14:15.653: %PKI-6-TRUSTPOINT_CREATE: Trustpoint: SLA-TrustPoint created succesfully
*Feb 12 00:14:15.657: %PKI-6-CONFIGAUTOSAVE: Running configuration saved to NVRAM[OK]
Router>
Router>en
Password:
*Feb 12 00:14:18.878: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 file
*Feb 12 00:14:18.910: %CALL_HOME-6-CALL_HOME_ENABLED: Call-home is enabled by Smart Agent
for Licensing.
Router#sh run | inc sec
*Feb 12 00:14:26.299: %SYS-5-CONFIG_P: Configured programmatically by process PnP Agent
Discovery from console as vty0ret
enable secret 9 $9$rDzH3rLqjlFhek$G9UDZE7moWqsKJEZfJAH2yO.SPhKZeKJsEe./CPEzl.
Router#
```

After the enable secret is prompted during the first login, and the admin enters a password, the admin entered password will be always masked. If the admin enters a weak password, they will be prompted again to enter strong password (i.e. the standard mix of upper/lower case characters, special characters, numbers etc.). The prompting will continue until the admin enters a strong password. The admin will be prompted to enter the strong secret password twice for confirming that admin is sure that it is the secret that they want to configure.

Telnet and HTTP

When the device is first booted after factory reset or fresh from the factory, by default the following takes place:

- Disable telnet
- Disable http server. HTTP client works.
- Enable SSH
- Enable https server

Examples

From a freshly booted device, below configurations for HTTPS and SSH are enabled.

```
ip http server
ip http authentication local
ip http secure-server
.
.

line vty 0 4
```

```
login
transport input ssh
line vty 5 14
login
transport input ssh
```

To enable Telnet, use the **line vty** command.

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#line vty 0
R1(config-line)#password Pass123
R1(config-line)#login
R1(config-line)#logging synchronous
R1(config-line)#exec-timeout 40
R1(config-line)#motd-banner
R1(config-line)#exit
R1(config)#
R1(config)#enable password Password
R1(config)#exit
```

To test the Telnet connectivity, from a client PC, type **telnet 192.168.10.1** and press **Enter**, then enter the telnet password. Execute the **enable** command and press **Enter**, then type the router password.

```
Packet Tracer PC Command Line 1.0
PC>telnet 192.168.10.1
Trying 192.168.10.1 …Open
User Access Verification
Password:
R1>enable
Password:
R1#
```

Now you are remotely connected to router R1 and you can execute all router commands through Telnet command line interface.

Accessing the CLI from a Remote Console

The remote console of the IR8340 can be accessed through Telnet or SSH. Telnet is disabled by default, and the more secure SSH should be used. For details on SSH access see the SSH chapter.

The following topics describe the procedure to access the CLI from a remote console:

Preparing to Connect to the Router Console

See the Cisco IOS-XE Device hardening guide at https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html for details.

Configuring the diagnostic and wait banners is optional, but recommended. The banners are especially useful as indicators to users about the status of their Telnet or SSH attempts.

To access the router remotely using Telnet from a TCP/IP network, configure the router to support virtual terminal lines using the **line vty** global configuration command. Configure the virtual terminal lines to require users to log in and specify a password.

See the Cisco IOS Terminal Services Command Reference document for more information about the line **vty global** configuration command.

To prevent disabling login on a line, specify a password with the **password** command when you configure the **login** command.

If you are using authentication, authorization, and accounting (AAA), configure the **login authentication** command. To prevent disabling login on a line for AAA authentication when you configure a list with the login authentication command, you must also configure that list using the **aaa authentication login** global configuration command.

For more information about AAA services, see the Cisco IOS XE Security Configuration Guide: Secure Connectivity and the Cisco IOS Security Command Reference documents. For more information about the **login line-configuration** command, see the Cisco IOS Terminal Services Command Reference document.

In addition, before you make a Telnet connection to the router, you must have a valid hostname for the router or have an IP address configured on the router. For more information about the requirements for connecting to the router using Telnet, information about customizing your Telnet services, and using Telnet key sequences, see the Cisco IOS Configuration Fundamentals Configuration Guide.

Setting Up the IR8340 to Run SSH

Follow the procedure given below to set up your device to run SSH:

Before you begin

Configure user authentication for local or remote access. This step is required.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**hostname** *hostname* **Example:** `Router(config)# hostname your_hostname`	Configures a hostname and IP domain name for your device. **Note** Follow this procedure only i
Step 3	**ip domain-name** *domain_name* **Example:** `Router(config)# ip domain-name your_domain_name`	Configures a host domain for your device.
Step 4	**crypto key generate rsa modulus** *size* **Example:** `Router(config)# crypto key generate rsa modulus 2048`	Enables the SSH server for local and remote authentication on the device and generates an RSA key pair. Generating an RSA key pair for the device automatically enables SSH. We recommend that a minimum modulus size of 2048 bits.

	Command or Action	Purpose
		When you generate RSA keys, you are prompted to enter a modulus length. A longer modulus length might be more secure, but it takes longer to generate and to use.  **Note** Follow this procedure only if yo
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Using Telnet to Access a Console Interface

Procedure

Step 1 From your terminal or PC, enter one of the following commands:

- **connect host** [*port*] [*keyword*]
- **telnet host** [*port*] [*keyword*]

Here, *host* is the router hostname or IP address, *port* is a decimal port number (23 is the default), and *keyword* is a supported keyword. For more information about these commands, see the Cisco IOS Terminal Services Command Reference document.

The following example shows how to use the **telnet** command to connect to a router named **router**:

```
unix_host% telnet router
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.
unix_host% connect
```

Step 2 Enter your login password:

```
User Access Verification
Password: mypassword
```



Note If no password has been configured, press **Return**.

Step 3 From user EXEC mode, enter the **enable** command:

```
Router> enable
```

Step 4 At the password prompt, enter your system password:

```
Password: enablepass
```

Step 5 When the **enable** password is accepted, the privileged EXEC mode prompt is displayed:

```
Router#
```

Step 6 You now have access to the CLI in privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.

Step 7 To exit the Telnet session, use the **exit** or **logout** command.

```
Router# logout
```

CLI Session Management

An inactivity timeout is configurable and can be enforced. Session locking provides protection from two users overwriting changes that the other has made. To prevent an internal process from using all the available capacity, some spare capacity is reserved for CLI session access. For example, this allows a user to remotely access a router.

Information About CLI Session Management

An inactivity timeout is configurable and can be enforced. Session locking provides protection from two users overwriting changes that each other has made. To prevent an internal process from using all the available capacity, some spare capacity is reserved for CLI session access. For example, this allows a user to remotely access the router.

Changing the CLI Session Timeout

Procedure

Step 1 **configure terminal**

Enters global configuration mode

Step 2 **line console 0**

Step 3 **session-timeout** *minutes*

The value of *minutes* sets the amount of time that the CLI waits before timing out. Setting the CLI session timeout increases the security of a CLI session. Specify a value of 0 for *minutes* to disable session timeout.

Step 4 **show line console 0**

Verifies the value to which the session timeout has been set, which is shown as the value for "Idle Session".

Locking a CLI Session

Before you begin

To configure a temporary password on a CLI session, use the **lock** command in EXEC mode. Before you can use the **lock** command, you need to configure the line using the **lockable** command. In this example the line is configured as **lockable**, and then the **lock** command is used and a temporary password is assigned.

Procedure

Step 1 `Router# configure terminal`

Enters global configuration mode.

Step 2 Enter the line upon which you want to be able to use the **lock** command.

```
Router(config)# line console 0
```

Step 3 `Router(config)# lockable`

Enables the line to be locked.

Step 4 `Router(config)# exit`

Step 5 `Router# lock`

The system prompts you for a password, which you must enter twice.

```
Password: <password>
Again: <password>
Locked
```

Communications, services, and additional information

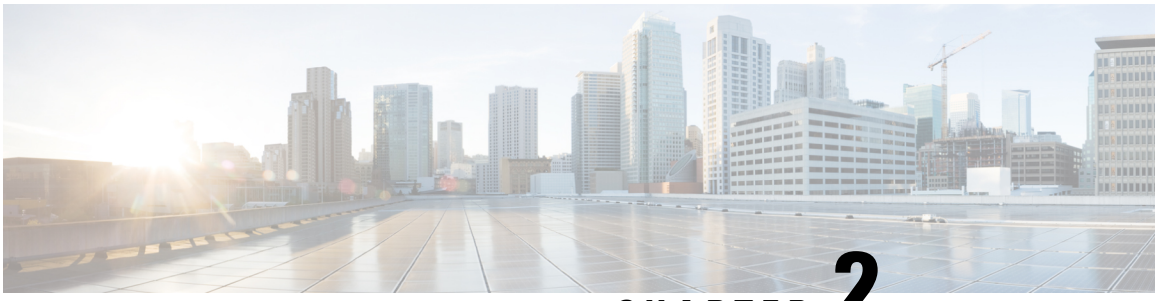
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class 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 2

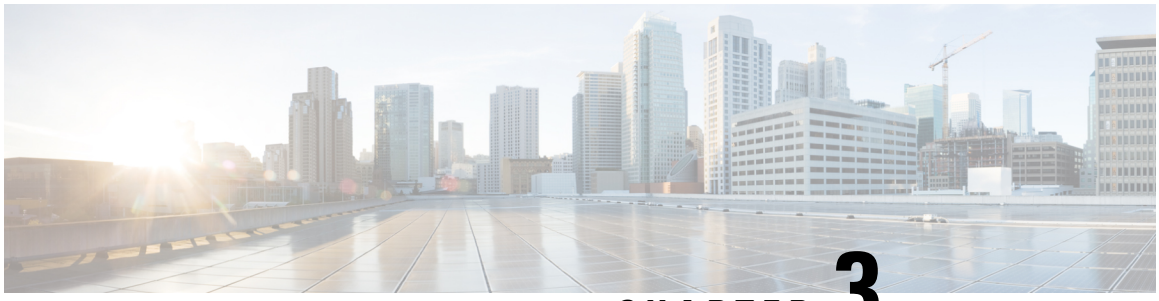
New Features

- New Features for Cisco IOS XE 26.2.1, on page 15

New Features for Cisco IOS XE 26.2.1

New features in this release are:

- DHCP Option 82
- Segment Routing over MPLS
- IKEv2 and IPsec PFS Post Quantum Cryptography
- Post Quantum Cryptography for SSHv2 sessions



CHAPTER 3

Using Cisco IOS XE Software

- Understanding Command Modes, on page 17
- Keyboard Shortcuts, on page 19
- Using the no and default Forms of Commands, on page 19
- Using the History Buffer to Recall Commands, on page 20
- Managing Configuration Files, on page 20
- Saving Configuration Changes, on page 20
- Filtering Output from the show and more Commands, on page 21
- Finding Support Information for Platforms and Cisco Software Images, on page 22

Understanding Command Modes

The command modes available in Cisco IOS XE are the same as those available in traditional Cisco IOS. Use the CLI to access Cisco IOS XE software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode that you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode, you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS XE software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

The following table describes how to access and exit various common command modes of the Cisco IOS XE software. It also shows examples of the prompts displayed for each mode.

Table 1: Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure terminal** command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.
Diagnostic	The router boots up or accesses diagnostic mode in the following scenarios: • In some cases, diagnostic mode will be reached when the Cisco IOS process or processes fail. In most scenarios, however, the router will reload. • A user-configured access policy is configured using the **transport-map** command that directs a user into diagnostic mode. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) is entered and the router is configured to go to diagnostic mode when the break signal is received.	`Router(diag)#`	If failure of the Cisco IOS process is the reason for entering diagnostic mode, the Cisco IOS problem must be resolved and the router rebooted to get out of diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or by using a method that is configured to connect to the Cisco IOS CLI.

Command Mode	Access Method	Prompt	Exit Method
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon#>`	To exit ROM monitor mode, manually boot a valid image or perform a reset with autoboot set so that a valid image is loaded.

Keyboard Shortcuts

Commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters.

The following table lists the keyboard shortcuts for entering and editing commands.

Table 2: Keyboard Shortcuts

Key Name	Purpose
Ctrl-B or the **Left Arrow** key	Move the cursor back one character.
Ctrl-F or the **Right Arrow** key	Move the cursor forward one character.
Ctrl-A	Move the cursor to the beginning of the command line.
Ctrl-E	Move the cursor to the end of the command line.
Esc B	Move the cursor back one word.
Esc F	Move the cursor forward one word.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to re-enable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to re-enable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Many CLI commands also have a **default** form. By issuing the <*command*> **default** command-name, you can configure the command to its default setting. The Cisco IOS software command reference publications describe the function from a **default** form of the command when the **default** form performs a different function than the plain and **no** forms of the command. To see what default commands are available on your system, enter **default ?** in the appropriate command mode.

Using the History Buffer to Recall Commands

The history buffer stores the last 20 commands you entered. History substitution allows you to access these commands without retyping them, by using special abbreviated commands.

The following table lists the history substitution commands.

Table 3: History Substitution Commands

Command	Purpose
Ctrl-P or the **Up Arrow** key[1]	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the **Down Arrow** key[1]	Returns to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the **Up Arrow** key.
Router# **`show history`**	While in EXEC mode, lists the last few commands you entered.

[1] The arrow keys function only on ANSI-compatible terminals such as VT100s.

Managing Configuration Files

The startup configuration file is stored in the nvram: file system and the running configuration files are stored in the system: file system. This configuration file storage setup is also used on several other Cisco router platforms.

IOS XE provides encryption of the configuration file. Encryption is discussed in length in the IOS XE hardening device guide which can be found here:https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html

As a matter of routine maintenance on any Cisco router, users should back up the startup configuration file by copying the startup configuration file from NVRAM to one of the router's other file systems and, additionally, to a network server. Backing up the startup configuration file provides an easy method of recovering the startup configuration file if the startup configuration file in NVRAM becomes unusable for any reason.

The **copy** command can be used to back up startup configuration files.

For more detailed information on managing configuration files, see the "Managing Configuration Files" section in the Cisco IOS XE Configuration Fundamentals Configuration Guide.

Saving Configuration Changes

Use the **copy running-config startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

```
Building configuration...
[OK]
Router#
*Sep 8 05:41:03.450: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 file
Router#
```

Note It may take a few minutes to save the configuration.

This task saves the configuration to the NVRAM.

Filtering Output from the show and more Commands

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case sensitive):

show *command* | {**append** | **begin** | **exclude** | **include** | **redirect** | **section** | **tee**} *regular-expression*

The output matches certain lines of information in the configuration file.

Example

In this example, a modifier of the **show interface** command (**include protocol**) is used to provide only the output lines in which the expression **protocol** is displayed:

```
Router# show interface | include protocol
GigabitEthernet0/0/0 is up, line protocol is up
1401 unknown protocol drops
GigabitEthernet0/0/1 is up, line protocol is up
3073 unknown protocol drops
WPAN0/1/0 is up, line protocol is up
0 unknown protocol drops
Cellular0/4/0 is up, line protocol is up
0 unknown protocol drops
Cellular0/4/1 is up (spoofing), line protocol is up (spoofing)
0 unknown protocol drops
Cellular0/5/0 is up, line protocol is up
0 unknown protocol drops
Cellular0/5/1 is down, line protocol is down
0 unknown protocol drops
Loopback1 is up, line protocol is up
0 unknown protocol drops
Tunnel1 is up, line protocol is up
Tunnel protocol/transport GRE/IP
0 unknown protocol drops
Tunnel2 is up, line protocol is up
Tunnel protocol/transport GRE/IP
0 unknown protocol drops
VirtualPortGroup1 is up, line protocol is up
0 unknown protocol drops
```

Finding Support Information for Platforms and Cisco Software Images

The Cisco IOS XE software is packaged in feature sets consisting of software images that support specific platforms.

All of the Cisco IOS-XE configuration guides can be found here: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-17/series.html

The group of feature sets that are available for a specific platform depends on which Cisco software images are included in a release. To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS XE software image, you can use Cisco Feature Navigator or see the https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-17/series.html.

Using Cisco Feature Navigator

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator is a tool that enables you to determine which Cisco IOS XE software images support a specific software release, feature set, or platform. To use the navigator tool, an account on Cisco.com is not required.

Getting Help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help that is specific to a command mode, a command, a keyword, or an argument, use one of the following commands.

Command	Purpose
`help`	Provides a brief description of the help system in any command mode.
`abbreviated-command-entry?`	Provides a list of commands that begin with a particular character string. **Note** There is no space between the command and the question mark.
`abbreviated-command-entry<Tab>`	Completes a partial command name.
`?`	Lists all the commands that are available for a particular command mode.

Command	Purpose
command ?	Lists the keywords or arguments that you must enter next on the command line. **Note** There is a space between the command and the question mark.

Finding Command Options: Example

This section provides information about how to display the syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering a part of a command followed by a space. The Cisco IOS XE software displays a list and brief descriptions of the available keywords and arguments. For example, if you are in global configuration mode and want to see all the keywords and arguments for the **arap** command, you should type **arap ?**.

The <cr> symbol in command help output stands for carriage return. On older keyboards, the carriage return key is the **Return** key. On most modern keyboards, the carriage return key is the **Enter** key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available, and that you must press **Enter** to complete the command.

The following table shows examples of using the question mark (**?**) to assist you in entering commands.

Table 4: Finding Command Options

Command	Comment

`Router> enable`
`Password: <password>`
`Router#`
	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to a " # " from the " > ", for example, `Router>` to `Router#`

`Router# configure terminal`
`Enter configuration commands, one per line. End with CNTL/Z.`
`Router(config)#`
	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to `Router (config)#`

`Router(config)# interface GigabitEthernet ?`
`<0-0> GigabitEthernet interface number`
 	
`Router(config)# interface GigabitEthernet 0/?`
`<0-0> Port Adapter number`
 	
`Router (config)# interface GigabitEthernet 0/0/?`
`<0-1> GigabitEthernet interface number`
 	
`Router (config)# interface GigabitEthernet 0/0/0?`
`<0-1>`
 	
`Router(config-if)#`
	Enter interface configuration mode by specifying the interface that you want to configure, using the **interface GigabitEthernet** global configuration command.
	Enter **?** to display what you must enter next on the command line.
	When the <cr> symbol is displayed, you can press **Enter** to complete the command.
	You are in interface configuration mode when the prompt changes to `Router(config-if)#`

Command	Comment

`Router(config-if)# ?`
`Interface configuration commands:`
`  .`
`  .`
`  .`
`  ip  Interface Internet Protocol`
`  config commands`
`  keepalive  Enable keepalive`
`  lan-name  LAN Name command`
`  llc2  LLC2 Interface Subcommands`
`  load-interval  Specify interval for load calculation`
`  for an interface`
`  locaddr-priority  Assign a priority group`
`  logging  Configure logging for interface`
`  loopback  Configure internal loopback on an`
`  interface`
`  mac-address  Manually set interface MAC address`
`  mls  mls router sub/interface commands`
`  mpoa  MPOA interface configuration commands`
`  mtu  Set the interface Maximum Transmission Unit (MTU)`
`  netbios  Use a defined NETBIOS access list`
`  or enable name-caching`
`  no  Negate a command or set its defaults`
`  nrzi-encoding  Enable use of NRZI encoding`
`  ntp  Configure NTP`
`  .`
`  .`
`  .`
`Router(config-if)#`
	Enter **?** to display a list of all the interface configuration commands available for the interface. This example shows only some of the available interface configuration commands.

Command	Comment
	Enter the command that you want to configure for the interface. This example uses the **ip** command. Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

Command	Comment
Router(config-if)# **ip ?** Interface IP configuration subcommands: access-group Specify access control for packets address Set the IP address of an interface admission Apply Network Admission Control auth-proxy Apply authenticaton proxy authentication authentication subcommands bandwidth-percent Set EIGRP bandwidth limit bgp BGP interface commands broadcast-address Set the broadcast address of an interface cef Cisco Express Forwarding interface commands cgmp Enable/disable CGMP clear-dont-fragment Enable clear dont fragment (Currently Only SDWAN Tunnel Interface) dampening-change Percent interface metric must change to cause update dampening-interval Time in seconds to check interface metrics ddns Configure dynamic DNS dhcp Configure DHCP parameters for this interface directed-broadcast Enable forwarding of directed broadcasts dlep DLEP interface commands dns Configure DNS server flow NetFlow related commands flowspec FlowSpec configuration header-compression IPHC options hello-interval Configures EIGRP-IPv4 hello interval helper-address Specify a destination address for UDP broadcasts hold-time Configures EIGRP-IPv4 hold time igmp IGMP interface commands information-reply Enable sending ICMP Information Reply messages irdp ICMP Router Discovery Protocol load-sharing Style of load sharing local-proxy-arp Enable local-proxy ARP mask-reply Enable sending ICMP Mask Reply messages mfib Interface Specific MFIB Control mrm Configure IP Multicast Routing Monitor tester mroute-cache Enable switching cache for incoming multicast packets mtu Set IP Maximum Transmission Unit multicast IP multicast interface commands mux Enable IP multiplexing for outgoing packets nat NAT interface commands nbar Network-Based Application Recognition network-broadcast Accept and respond to network-prefix-directed broadcasts next-hop-self Configures EIGRP-IPv4 next-hop-self ospf OSPF interface commands pim PIM interface commands policy Enable policy routing portbundle IP Portbundle configuration	

Command	Comment
probe Enable HP Probe support proxy-arp Enable proxy ARP rarp-server Enable RARP server for static arp entries reassembly Reassembly redirects Enable sending ICMP Redirect messages rgmp Enable/disable RGMP rip Router Information Protocol route-cache Enable fast-switching cache for outgoing packets router IP router interface commands rsvp RSVP Interface Commands rtp RTP parameters sap Session Advertisement Protocol interface commands sdr Session Directory Protocol interface commands security DDN IP Security Option service IP service split-horizon Perform split horizon sticky-arp Allow the creation of sticky ARP entries subscriber IP session configuration options summary-address Perform address summarization tcp TCP interface commands topology-accounting Enable accounting for IP on all topologies of Router(config-if)# **ip**	
Router(config-if)# **ip address ?** A.B.C.D IP address dhcp IP Address negotiated via DHCP pool IP Address autoconfigured from a local DHCP pool Router(config-if)# **ip address**	Enter the command that you want to configure for the interface. This example uses the **ip address** command. Enter **?** to display what you must enter next on the command line. A carriage return (<cr>) is not displayed. Therefore, you must enter additional keywords or arguments to complete the command.
Router(config-if)# **ip address 172.16.0.1 ?** A.B.C.D IP subnet mask Router(config-if)# **ip address 172.16.0.1**	Enter the keyword or argument that you want to use. This example uses the 172.16.0.1 IP address. Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask. <cr> is not displayed. Therefore, you must enter additional keywords or arguments to complete the command.

Command	Comment

`Router(config-if)# ip address 172.16.0.1 255.255.255.0 ?`
`  secondary  Make this IP address a secondary address`
`  <cr>`
`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask.
	Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**.
	<cr> is displayed. Press **Enter** to complete the command, or enter another keyword.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0` `Router(config-if)#`	Press **Enter** to complete the command.

Using Software Advisor

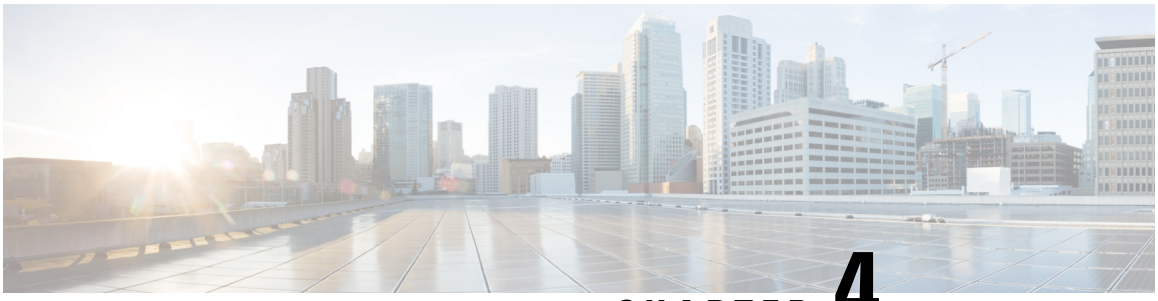
Cisco maintains the Software Advisor tool. See Tools and Resources. Use the Software Advisor tool to see if a feature is supported in a Cisco IOS XE release, to locate the software document for that feature, or to check the minimum software requirements of Cisco IOS XE software with the hardware installed on your router. You must be a registered user on Cisco.com to access this tool.

Using Software Release Notes

See the release notes for information about the following:

- Open and resolved severity 1 and 2 caveats

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in previous releases. For cumulative feature information, refer to the Cisco Feature Navigator at: http://www.cisco.com/go/cfn/.



CHAPTER 4

Basic Router Configuration

This chapter contains the following sections:

- IR8340 Interface Naming, on page 29
- Basic Configuration, on page 30
- Configuring Global Parameters, on page 35
- Configuring the Gigabit Ethernet Interface, on page 36
- Support for sub-interface on GigabitEthernet0/0/0, on page 37
- Configuring a Loopback Interface, on page 37
- Configuring CPU Allocation, on page 38
- Enabling Cisco Discovery Protocol, on page 39
- Configuring Command-Line Access, on page 39
- Configuring Static Routes, on page 41
- Configuring Dynamic Routes, on page 43
- Modular QoS (MQC), on page 44

IR8340 Interface Naming

The supported hardware interfaces and their naming conventions are in the following table:

Hardware Interface	Naming Convention
Gigabit Ethernet WAN ports	GigabitEthernet0/0/0 GigabitEthernet0/0/1

Hardware Interface	Naming Convention
Gigabit Ethernet LAN ports	GigabitEthernet0/1/0 GigabitEthernet0/1/1 GigabitEthernet0/1/2 GigabitEthernet0/1/3 GigabitEthernet0/1/4 GigabitEthernet0/1/5 GigabitEthernet0/1/6 GigabitEthernet0/1/7 GigabitEthernet0/1/8 GigabitEthernet0/1/9 GigabitEthernet0/1/10 GigabitEthernet0/1/11
NIM Interface	0/2/0 0/2/1 0/3/0 0/3/1
Cellular Interface	cellular 0/4/0 cellular 0/4/1 cellular 0/5/0 cellular 0/5/1
mSATA SSD	msata
GPIO (External Alarm Interface)	alarm contact 1-2

Basic Configuration

The basic configuration is a result of the entries you made during the initial configuration dialog. This means the router has at least one interface set with an IP address to be reachable, either through WebUI or to allow the PnP process to work. Use the **show running-config** command to view the initial configuration, as shown in the following example:

```
IR8340#show running-config
Building configuration...

Current configuration : 6937 bytes
!
! Last configuration change at 17:35:35 UTC Wed Mar 23 2022
!
version 17.8
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
!
hostname Sumatra-DUT
!
boot-start-marker
boot system flash
0081684787552360_ir8340-universalk9.BLD_V178_THROTTLE_LATEST_20220321_143500_V17_8_0_65.SSA.bin
boot-end-marker
!
!
! card type command needed for slot/bay 0/3
no aaa new-model
!
!
!
!
!
!
!
!
!
!
login on-success log
!
!
!
!
!
!
!
subscriber templating
!
!
!
!
vtp mode transparent
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
crypto pki trustpoint TP-self-signed-829458546
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-829458546
 revocation-check none
 rsakeypair TP-self-signed-829458546
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
!
crypto pki certificate chain TP-self-signed-829458546
```

```
 certificate self-signed 01
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 38323934 35383534 36301E17 0D323230 33323331 37333433
  395A170D 33323033 32323137 33343339 5A303031 2E302C06 03550403 1325494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3832 39343538
  35343630 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 90B8DF91 C27E2942 0C342B4E 5C7A03C5 91E9ACE7 C29E8D1B 291B8799
  3C468901 6B4DA7E9 E4480CC0 B4C2E1D9 918BBDB9 26EB7EE4 E3D7F76B 42FD0642
  D29E33DA 01ECAE68 2A7254DE 32163A82 959C75B7 DAA84B34 670B7CA3 F94B0803
  A9B11490 A350030D 5DD8178F 1579E30B 8D0B079A 090917F3 60031B68 F961A0CE
  9E958239 973E6B08 3D293F10 38136F15 83D8C801 A58D71E4 3CC128AE 8B9BF214
  AA4B09E2 CB9841E3 0F455D66 504F89E9 F6B70AC7 1B2C3C48 5ED5679C 121A5415
  198D7D0D 0AE444E1 76309174 67FF66E6 BBAFA506 EE10C77D EDD89004 BE72EC05
  F960AAC0 F7DD0A3E 7769C277 3529B0CC 06788C4C B0A82D51 F8A74DAB 997D6716
  3F0F694B 02030100 01A35330 51300F06 03551D13 0101FF04 05300301 01FF301F
  0603551D 23041830 16801439 96B67C33 D166686C 2C7A99C0 698D085E B69B0C30
  1D060355 1D0E0416 04143996 B67C33D1 66686C2C 7A99C069 8D085EB6 9B0C300D
  06092A86 4886F70D 01010505 00038201 01008D8D 7BED46B9 4DDF60D3 01BE178A
  D4B97142 9BF6991E 70693302 878693DE 5C39373B FE6D2D77 3B353F2C 21707C5F
  67C99A51 E07D5FF6 B59FCDF1 EC6751B0 8DB7ED72 FAD9AED5 7D7F7895 9DA6FAD5
  72304A73 869F4013 1559F607 7F8303E2 8E8F1011 525BC32B A32EF28A 7EC811C2
  45268BC9 225B65AB 94998717 0BFC0F4F C772233D 7B6635DB AA554FB4 67EC7F7B
  258BDD81 855B64A5 6236CE38 58B795D7 FD5096BE 8DA304CF 987450BE 4AD62994
  CCC2D3FB 540A8BA5 A8CE6109 4DAB37C4 A692F2C9 02B653F8 7B539BC1 5B338E26
  A71F8C43 192521BD F10401D8 0E23D095 1C943214 A01B9E48 40299F2C 35183755
  26956737 8CF47405 E7757043 80AA8C93 8B41
       quit
crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
       quit
!
!
no license feature hseck9
license udi pid IR8340-K9 sn FDO2523J7GF
memory free low-watermark processor 63132
!
diagnostic bootup level minimal
```

```
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
no power main redundant
!
!
!
!
redundancy
 mode none
!
!
!
!
controller Cellular 0/4/0
!
controller Cellular 0/5/0
!
!
vlan internal allocation policy ascending
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface GigabitEthernet0/0/0
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/0/1
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/1/0
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface GigabitEthernet0/1/4
!
interface GigabitEthernet0/1/5
!
interface GigabitEthernet0/1/6
!
interface GigabitEthernet0/1/7
```

```
!
interface GigabitEthernet0/1/8
!
interface GigabitEthernet0/1/9
!
interface GigabitEthernet0/1/10
!
interface GigabitEthernet0/1/11
!
interface Serial0/2/0
!
interface Serial0/2/1
!
interface Serial0/2/2
!
interface Serial0/2/3
!
interface Serial0/2/4
!
interface Serial0/2/5
!
interface Serial0/2/6
!
interface Serial0/2/7
!
interface Cellular0/4/0
 no ip address
 shutdown
!
interface Cellular0/4/1
 no ip address
 shutdown
!
interface Cellular0/5/0
 no ip address
 shutdown
!
interface Cellular0/5/1
 no ip address
 shutdown
!
interface Vlan1
 no ip address
!
ip http server
ip http authentication local
ip http secure-server
ip forward-protocol nd
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
line con 0
 stopbits 1
line aux 0
```

```
line vty 0 4
 login
 transport input ssh
line vty 5 14
 login
 transport input ssh
!
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email
 address to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
!
!
!
!
!
!
end
```

Configuring Global Parameters

To configure global parameters for your router, follow these steps.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router> enable`
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode when using the console port.
		Use the following to connect to the router with a remote terminal:
		`telnet router-name or address`
		`Login: login-id`
		`Password: *********`
		`Router> enable`

Step 2	**hostname** *name* **Example:** `Router(config)# hostname Router`	Specifies the name for the router.
Step 3	**enable secret** *password* or **enable password** *password* **Example:** `Router(config)# enable secret password`	Specifies a password to prevent unauthorized access to the router.

Configuring the Gigabit Ethernet Interface

The router features two Gigabit Ethernet (GE) ports that can be used to enable WAN uplink connectivity:

- Two GigE Copper port (RJ45) on the midplane board. It supports standard 3-speed (10/100/1000) Ethernet features including auto-MDIX.
- Two SFP socket. It supports standard 1000Base-X or 100Base-FX Ethernet over single-mode or multi-mode fiber.

Note These ports are combo mode. Either copper or SFP can be used distinctly, but not at the same time for each interface.

To configure the Gigabit Ethernet interface, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	
Step 2	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables forwarding of IPv6 unicast data packets.
Step 3	**interface GigabitEthernet** *slot/bay/port* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Enters the configuration mode for an interface on the router.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.12.2 255.255.255.0`	Sets the IP address and subnet mask for the specified interface. Use this Step if you are configuring an IPv4 address.
Step 5	**ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001.db8::ffff:1/128`	Sets the IPv6 address and prefix for the specified interface. Use this step instead of Step 2, if you are configuring an IPv6 address. IPv6 unicast-routing needs to be set-up as well, see further information in the IPv6 Addressing and Basic Connectivity Configuration Guide.
Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface and changes its state from administratively down to administratively up.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits the configuration mode of interface and returns to the global configuration mode.

Support for sub-interface on GigabitEthernet0/0/0

Cisco IOS XE supports sub-interfaces and dot1q configuration on the g0/0/0 interface. For example:

```
Router(config)#interface g0/0/0.?
  <1-4294967295>  GigabitEthernet interface number
Router(config-subif)#encapsulation ?
  dot1Q           IEEE 802.1Q Virtual LAN
```

Configuring a Loopback Interface

Before you begin

The loopback interface acts as a placeholder for the static IP address and provides default routing information.

To configure a loopback interface, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	
Step 2	**interface** *type number* **Example:** `Router(config)# interface Loopback 0`	Enters configuration mode on the loopback interface.
Step 3	(Option 1) **ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.0.2.0 255.255.0.0`	Sets the IP address and subnet mask on the loopback interface. (If you are configuring an IPv6 address, use the **ipv6 address** *ipv6-address/prefix* command described below.
Step 4	(Option 2) **ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001:db8::ffff:1/128`	Sets the IPv6 address and prefix on the loopback interface.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the loopback interface and returns to global configuration mode.

Example

Verifying Loopback Interface Configuration

Enter the **show interface loopback** command. You should see an output similar to the following example:

```
Router# show interface loopback 0
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 192.0.2.0/16
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Alternatively, use the **ping** command to verify the loopback interface, as shown in the following example:

```
Router# ping 192.0.2.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.0, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Configuring CPU Allocation

You can change allocation of the available 8 cores of the IR8340 router to different functions in a flexible way. The core allocation is based on the customer configuration of the different services.

From Cisco IOS XE Release 17.9 onwards, you can use the **platform resource {control-plane-heavy|data-plane-heavy|service-plane-heavy}** command to adjust the cores across control plane, service plane, and data plane. However, you have to save the configuration and reboot the device for the configured profile to take effect.

```
Router#configure terminal
Router#platform resource ?
  control-plane-heavy  Use Control Plane Heavy template
  data-plane-heavy     Use Data Plane Heavy template
  service-plane-heavy  Use Service Plane Heavy template
```

The following show command output shows the default CPU cores allocation:

```
Router#show platform software cpu alloc
CPU alloc information:
Control plane cpu alloc: 0
Data plane cpu alloc: 4-7
Service plane cpu alloc: 1-3
Slow control plane cpu alloc: 1-3
Template used: default-service_plane_heavy
```

The following show command output shows the CPU cores allocaiton for the service plane:

```
Router#show platform software cpu alloc
CPU alloc information:
Control plane cpu alloc: 0
Data plane cpu alloc: 4-7
Service plane cpu alloc: 1-3
Slow control plane cpu alloc: 1-3
Template used: CLI-service_plane_heavy
```

The following show command output shows the CPU cores allocaiton for the control plane:

```
Router#show platform software cpu alloc
CPU alloc information:
Control plane cpu alloc: 0
Data plane cpu alloc: 4-7
Service plane cpu alloc: 2-3
Slow control plane cpu alloc: 1-3
Template used: CLI-control_plane_heavy
```

The following show command output shows the CPU cores allocaiton for the data plane:

```
Router#show platform software cpu alloc
CPU alloc information:
Control plane cpu alloc: 0
Data plane cpu alloc: 2-7
Service plane cpu alloc: 1
Slow control plane cpu alloc: 1-2
Template used: CLI-data_plane_heavy
```

Enabling Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is enabled by default on the router. It may be disabled if needed for security purposes.

For more information on using CDP, see Cisco Discovery Protocol Configuration Guide, Cisco IOS XE Release 3S.

Configuring Command-Line Access

To configure parameters to control access to the router, follow these steps.

Note Transport input must be set as explained in the previous Telnet and SSH sections of the guide.

Procedure

	Command or Action	Purpose
Step 1	**line** [**aux** | **console** | **tty** | **vty**] *line-number* **Example:** `Router(config)# line console 0`	Enters line configuration mode, and specifies the type of line. The example provided here specifies a console terminal for access.
Step 2	**password** *password* **Example:** `Router(config-line)# password userpass`	Specifies a unique password for the console terminal line.
Step 3	**login** **Example:** `Router(config-line)# login`	Enables password checking at terminal session login.

Step 4	**exec-timeout** *minutes* [*seconds*]
	Example:
	`Router(config-line)# exec-timeout 5 30`
	`Router(config-line)#`
		Sets the interval during which the EXEC command interpreter waits until user input is detected. The default is 10 minutes. Optionally, adds seconds to the interval value.
		The example provided here shows a timeout of 5 minutes and 30 seconds. Entering a timeout of **0 0** specifies never to time out.

Step 5	**exit** **Example:** `Router(config-line)# exit`	Exits line configuration mode to re-enter global configuration mode.

Step 6	**line** [**aux** | **console** | **tty** | **vty**] *line-number*
	Example:
	`Router(config)# line vty 0 4`
	`Router(config-line)#`
		Specifies a virtual terminal for remote console access.

Step 7	**end** **Example:** `Router(config-line)# end`	Exits line configuration mode, and returns to privileged EXEC mode.

Example

The following configuration shows the command-line access commands. Note that transport input none is the default, but if SSH is enabled this must be set to ssh.

You do not have to input the commands marked **default**. These commands appear automatically in the configuration file that is generated when you use the **show running-config** command.

```
!
line console 0
exec-timeout 10 0
password 4youreyesonly
login
transport input none (default)
stopbits 1 (default)
line vty 0 4
password secret
login
!
```

Configuring Static Routes

Static routes provide fixed routing paths through the network. They are manually configured on the router. If the network topology changes, the static route must be updated with a new route. Static routes are private routes unless they are redistributed by a routing protocol.

To configure static routes, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	(Option 1) **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **Example:** `Router(config)# ip route 192.10.2.3 255.255.0.0 10.10.10.2`	Specifies a static route for the IP packets. (If you are configuring an IPv6 address, use the **ipv6 route** command described below.)
Step 2	(Option 2) **ipv6 route** *prefix/mask* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} **Example:** `Router(config)# ipv6 route 2001:db8:2::/64 2001:db8:3::0`	Specifies a static route for the IP packets. See additional information for IPv6 here: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-16-10/ip6b-xe-16-10-book/read-me-first.html
Step 3	(Option 3) **ip route vrf** *vrf-name ip-address*	Configure IP router vrf.

	Command or Action	Purpose
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

In the following configuration example, the static route sends out all IP packets with a destination IP address of 192.168.1.0 and a subnet mask of 255.255.255.0 on the Gigabit Ethernet interface to another device with an IP address of 10.10.10.2. Specifically, the packets are sent to the configured layer 3 adjacent device.

You do not have to enter the command marked **default**. This command appears automatically in the configuration file generated when you use the **running-config** command.

```
!
ip classless (default)
ip route 2001:db8:2::/64 2001:db8:3::0
```

Verifying Configuration

To verify that you have configured static routing correctly, enter the **show ip route** command (or **show ipv6 route** command) and look for static routes marked with the letter S.

When you use an IPv4 address, you should see verification output similar to the following:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
S*   0.0.0.0/0 is directly connected, GigabitEthernet0/0/0
```

When you use an IPv6 address, you should see verification output similar to the following:

```
Router# show ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE -
Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       ls - LISP site, ld - LISP dyn-EID, a - Application

C   2001:DB8:3::/64 [0/0]
       via GigabitEthernet0/0/2, directly connected
S   2001:DB8:2::/64 [1/0]
       via 2001:DB8:3::1
```

Configuring Dynamic Routes

In dynamic routing, the network protocol adjusts the path automatically, based on network traffic or topology. Changes in dynamic routes are shared with other routers in the network.

All of the Cisco IOS-XE configuration guides can be found here: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-17/series.html

Configuring Routing Information Protocol

To configure the RIP on a router, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode, and enables RIP on the router.
Step 2	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies use of RIP version 1 or 2.

Step 3	**network** *ip-address*
	Example:
	`Router(config-router)# network 192.168.1.1`
	`Router(config-router)# network 10.10.7.1`
		Specifies a list of networks on which RIP is to be applied, using the address of the network of each directly connected network.

Step 4	**no auto-summary** **Example:** `Router(config-router)# no auto-summary`	Disables automatic summarization of subnet routes into network-level routes. This allows subprefix routing information to pass across classful network boundaries.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode, and enters privileged EXEC mode.

Example

Verifying Configuration

To verify that you have configured RIP correctly, enter the **show ip route** command and look for RIP routes marked with the letter R. You should see an output similar to the one shown in the following example:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
R    3.0.0.0/8 [120/1] via 2.2.2.1, 00:00:02, Ethernet0/0/0
```

Configuring Enhanced Interior Gateway Routing Protocol

The Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. The convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP, and IGRP is now obsolete.

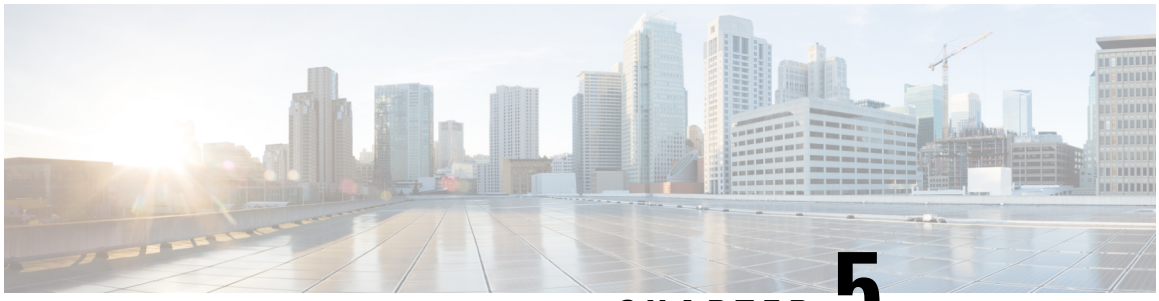
The convergence technology of EIGRP is based on an algorithm called the Diffusing Update Algorithm (DUAL). The algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize. Devices that are not affected by topology changes are not involved in recomputations

Details on configuring Enhanced Interior Gateway Routing Protocol (EIGRP), are found in the following guide: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/xe-16-10/ire-xe-16-10-book/ire-enhanced-igrp.html

Modular QoS (MQC)

This section provides an overview of Modular QoS CLI (MQC), which is how all QoS features are configured on the IoT Integrated Services Router. MQC is a standardized approach to enabling QoS on Cisco routing and switching platforms.

Follow the procedures that are in the QoS Modular QoS Command-Line Interface Configuration Guide, Cisco IOS XE 17 guide.



CHAPTER 5

Configuring Secure Shell

This section contains the following topics:

- Information About Secure Shell , on page 45
- How to Configure Secure Shell, on page 47
- Information about Secure Copy, on page 51
- Additional References, on page 53

Information About Secure Shell

Secure Shell (SSH) is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. This software release supports SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2).

Prerequisites for Configuring Secure Shell

The following are the prerequisites for configuring the device for secure shell (SSH):

- For SSH to work, the switch needs an RSA public/private key pair.
- The Secure Shell (SSH) server requires an IPsec (Data Encryption Standard [DES] or 3DES) encryption software image; the SSH client requires an IPsec (DES or 3DES) encryption software image.)
- Configure a hostname and host domain for your device by using the hostname and ip domain-name commands in global configuration mode. Use the **hostname** and **ip domain-name** commands in global configuration mode.

Restrictions for Configuring Secure Shell

The following are restrictions for configuring the IR8340 for secure shell.

- The router supports RSA authentication.
- SSH supports only the execution-shell application.
- The SSH server and the SSH client are supported only on Data Encryption Standard (DES) (56-bit) and 3DES (168-bit) data encryption software. In DES software images, DES is the only encryption algorithm available. In 3DES software images, both DES and 3DES encryption algorithms are available.

Note Cisco highly recommends the 3DES encryption as it is stronger.

See the Cisco IOS-XE Device hardening guide at https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html for details.

- This software release supports IP Security (IPSec).
- The IR8340 supports the Advanced Encryption Standard (AES) encryption algorithm with a 128-bit key, 192-bit key, or 256-bit key. However, symmetric cipher AES to encrypt the keys is not supported.
- The login banner is not supported in Secure Shell Version 1. It is supported in Secure Shell Version 2, which Cisco recommends due to its better security.
- The -l keyword and userid :{number} {ip-address} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for console access.

SSH And Router Access

Secure Shell (SSH) is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. This software release supports SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2). SSH functions the same in IPv6 as in IPv4. For IPv6, SSH supports IPv6 addresses and enables secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

SSH Servers, Integrated Clients, and Supported Versions

The Secure Shell (SSH) Integrated Client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running the SSH server. This connection provides functionality similar to that of an outbound Telnet connection except that the connection is encrypted. With authentication and encryption, the SSH client allows for secure communication over an unsecured network.

The SSH server and SSH integrated client are applications that run on the switch. The SSH server works with the SSH client supported in this release and with non-Cisco SSH clients. The SSH client works with publicly and commercially available SSH servers. The SSH client supports the ciphers of Data Encryption Standard (DES), 3DES, and password authentication.

Note The SSH client functionality is available only when the SSH server is enabled.

User authentication is performed like that in the Telnet session to the device. SSH also supports the following user authentication methods:

- TACACS+
- RADIUS
- Local authentication and authorization

SSH Configuration Guidelines

Follow these guidelines when configuring the device as an SSH server or SSH client:

- An RSA key pair generated by a SSHv1 server can be used by an SSHv2 server, and the reverse.
- If you get CLI error messages after entering the **crypto key generate rsa** global configuration command, an RSA key pair has not been generated. Reconfigure the hostname and domain, and then enter the **crypto key generate rsa** command.
- When generating the RSA key pair, the message *No hostname specified* might appear. If it does, you must configure an IP hostname by using the **hostname** global configuration command.
- When generating the RSA key pair, the message No domain specified might appear. If it does, you must configure an IP domain name by using the **ip domain-name** global configuration command.
- When configuring the local authentication and authorization authentication method, make sure that AAA is disabled on the console.

How to Configure Secure Shell

Setting Up the IR8340 to Run SSH

Follow the procedure given below to set up your device to run SSH:

Before you begin

Configure user authentication for local or remote access. This step is required.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**hostname** *hostname* **Example:** `Router(config)# hostname your_hostname`	Configures a hostname and IP domain name for your device. **Note** Follow this procedure only i
Step 3	**ip domain-name** *domain_name* **Example:** `Router(config)# ip domain-name your_domain_name`	Configures a host domain for your device.

	Command or Action	Purpose
Step 4	**crypto key generate rsa modulus** *size* **Example:** `Router(config)# crypto key generate rsa modulus 2048`	Enables the SSH server for local and remote authentication on the device and generates an RSA key pair. Generating an RSA key pair for the device automatically enables SSH. We recommend that a minimum modulus size of 2048 bits. When you generate RSA keys, you are prompted to enter a modulus length. A longer modulus length might be more secure, but it takes longer to generate and to use. **Note** Follow this procedure only if yo
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring the SSH Server

Follow these steps to configure the SSH server:

Note This procedure is only required if you are configuring the device as an SSH server.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `IR8340# configure terminal`	Enters global configuration mode.
Step 2	**ip ssh version** [2] **Example:** `IR8340(config)# ip ssh version 2`	(Optional) Configures the device to run SSH Version 2. If you do not enter this command or do not specify a keyword, the SSH server selects the latest SSH version supported by the SSH client. For example, if the SSH client supports SSHv1 and SSHv2, the SSH server selects SSHv2.

	Command or Action	Purpose

Step 3	**ip ssh** {**timeout** *seconds* | **authentication-retries** *number*}
	Example:
	`IR8340(config)# ip ssh timeout 90`
	`ip ssh authentication-retries 2`
		Configures the SSH control parameters:
		• Specify the time-out value in seconds; the default is 120 seconds. The range is 0 to 120 seconds. This parameter applies to the SSH negotiation phase. After the connection is established, the device uses the default time-out values of the CLI-based sessions.
		By default, up to five simultaneous, encrypted SSH connections for multiple CLI-based sessions over the network are available (session 0 to session 4). After the execution shell starts, the CLI-based session time-out value returns to the default of 10 minutes.
		• Specify the number of times that a client can re-authenticate to the server. The default is 3; the range is 0 to 5.
		Repeat this step when configuring both parameters.

Step 4	Use one or both of the following:
	• **line vty** *line_number* [*ending line number*]
	• **transport input ssh**
	Example:
	`IR8340(config)# line vty 1 10`
	or
	`IR8340(config-line)# transport input ssh`
		(Optional) Configures the virtual terminal line settings.
		• Enters line configuration mode to configure the virtual terminal line settings. For the *line_number* and *ending_line_number* arguments, the range is from 0 to 15.
		• Specifies that the device prevents non-SSH Telnet connections, limiting the device to only SSH connections.

Step 5	**end** **Example:** `IR8340(config-line)# end`	Exits line configuration mode and returns to privileged EXEC mode.

Monitoring the SSH Configuration and Status

Table 5: Commands for Displaying the SSH Server Configuration and Status

Command	Purpose
show ip ssh	Shows the version and configuration information for the SSH server.
show ssh	Shows the status of the SSH server.

Configuring the Router for Local Authentication and Authorization

You can configure AAA to operate without a server by setting the switch to implement AAA in local mode. The router then handles authentication and authorization. No accounting is available in this configuration.

Follow these steps to configure AAA to operate without a server by setting the router to implement AAA in local mode:

Note To secure the router for HTTP access by using AAA methods, you must configure the router with the ip http authentication aaa global configuration command. Configuring AAA authentication does not secure the router for HTTP access by using AAA methods.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `IR8340# configure terminal`	Enters global configuration mode.
Step 2	**aaa new-model** **Example:** `IR8340(config)# aaa new-model`	Enables AAA
Step 3	**aaa authentication login default local** **Example:** `IR8340(config)# aaa authentication login default local`	Sets the login authentication to use the local username database. The default keyword applies the local user database authentication to all ports.

Step 4	**line vty** *line-number*
	Example:
	`IR8340(config)# line vty 0 4`
	`IR8340(config-line)#`	

	Command or Action	Purpose
Step 5	**aaa authorization exec default local** **Example:** `IR8340(config-line)# aaa authorization exec default local`	Configures user AAA authorization, check the local database, and allow the user to run an EXEC shell.
Step 6	**aaa authorization network local** **Example:** `IR8340(config-line)# aaa authorization network local`	Configures user AAA authorization for all network-related service requests.
Step 7	**username** *name* **privilege** *level* **password** *encryption-type password* **Example:** `IR8340(config-line)# username your_user_name privilege 1 password 7 secret567`	Enters the local database, and establishes a username-based authentication system. Repeat this command for each user. **a.** For *name*, specify the user ID as one word. Spaces and quotation marks are not allowed. **b.** (Optional) For *level*, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access. Level 0 gives user EXEC mode access. **c.** For encryption-type, enter 0 to specify that an unencrypted password follows. Enter 7 to specify that a hidden password follows. **d.** For password, specify the password the user must enter to gain access to the switch. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the username command.
Step 8	**end** **Example:** `IR8340(config-line)# end`	Exits line configuration mode and returns to privileged EXEC mode.

Information about Secure Copy

The Secure Copy Protocol (SCP) feature provides a secure and authenticated method for copying router configuration or router image files. SCP relies on Secure Shell (SSH), an application and a protocol that provide a secure replacement for the Berkeley r-tools.

Prerequisites for Secure Copy

The following are the prerequisites for configuring the device for secure shell (SSH):

- Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the switch.
- Because SCP relies on SSH for its secure transport, the router must have an RSA key pair.
- SCP relies on SSH for security.
- SCP requires that authentication, authorization, and accounting (AAA) authorization be configured so the router can determine whether the user has the correct privilege level.
- A user must have appropriate authorization to use SCP.
- A user who has appropriate authorization can use SCP to copy any file in the Cisco IOS File System (IFS) to and from a switch by using the **copy** command. An authorized administrator can also do this from a workstation.

Restrictions for Configuring Secure Copy

- Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the router.
- When using SCP, you cannot enter the password into the **copy** command. You must enter the password when prompted.

Configuring Secure Copy

To configure the Cisco IR8340 for Secure Copy (SCP) server-side functionality, perform the following steps.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Sets AAA authentication at login.
Step 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]	Enables the AAA access control system.

	Command or Action	Purpose
	Example: `Device(config)# aaa authentication login default group tacacs+`	
Step 5	**username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password* **Example:** `Device(config)# username superuser privilege 2 secret superpassword`	Establishes a username-based authentication system. **Note** You may omit this step if a n been configured.
Step 6	**ip scp server enable** **Example:** `Device(config)# ip scp server enable`	Enables SCP server-side functionality.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	(Optional) Displays the SCP server-side functionality.
Step 9	**debug ip scp** **Example:** `Device# debug ip scp`	(Optional) Troubleshoots SCP authentication problems.

Example

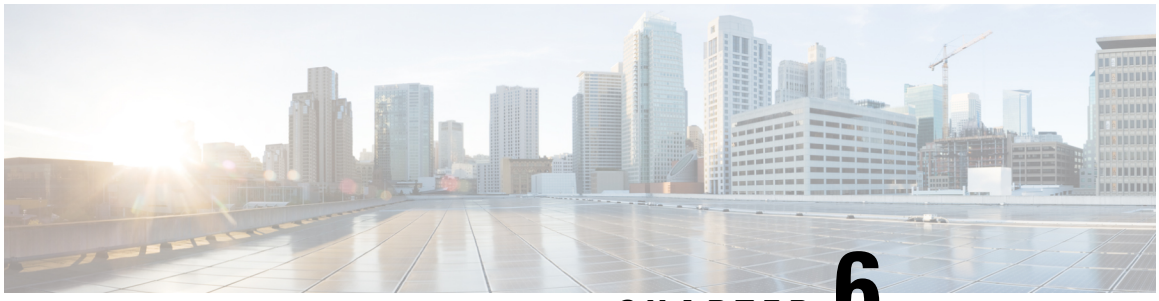
```
scp://your_username@remotehost:/some/remote/directory/file flash:
```

Additional References

The following sections provide references related to the SSH feature.

Related Topic	Document Title
Configuring Identity Control policies and Identity Service templates for Session Aware networking.	Session Aware Networking Configuration Guide, Cisco IOS XE Release 3SE: https://www.cisco.com/en/US/docs/ios-xml/ios/san/configuration/xe-3se/3850/san-xe-3se-3850-book.pdf

Related Topic	Document Title
Configuring RADIUS, TACACS+, Secure Shell, 802.1X and AAA.	Secure Shell Configuration Guide, Cisco IOS XE Gibraltar 16.11.x: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-11/configuration_guide/sec/b_1611_sec_9500_cg/configuring_secure_shell__ssh_.html



CHAPTER 6

Installing the Software

This chapter contains the following sections:

- Installing the Software, on page 55
- Installing the Cisco IOS XE Release, on page 56
- ROMMON Images, on page 57
- File Systems, on page 57
- Option to Enable or Disable USB Access, on page 58
- Autogenerated File Directories and Files, on page 59
- Flash Storage, on page 60
- Related Documentation, on page 60

Installing the Software

Installing software on the router involves installing a consolidated package (bootable image). This consists of a bundle of subpackages (modular software units), with each subpackage controlling a different set of functions.

It is better to upgrade software in a planned period of maintenance when an interruption in service is acceptable. The router needs to be rebooted for a software upgrade to take effect.

Cisco Software Licensing

Cisco software licensing consists of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.

The IR8340 uses Enhanced Smart Licensing, which is discussed in detail in Smart Licensing Using Policy (SLP), on page 69.

Consolidated Packages

To obtain software images for the router, go to: https://software.cisco.com/download/home/286200112

Note All of the IOS-XE feature set may not apply to the IR8340. Some features may not have been implemented yet, or are not appropriate for this platform.

An image-based license is used to help bring up all the subsystems that correspond to a license. This license is enforced only at boot time.

One of the following image-based licenses can be pre-installed on the IR8340 router:

- Network-Essentials
- Network-Advantage
- HSEC license

Note Details of the Network-Essentials and Network-Advantage contents can be found in the IR8340 product data sheet.

Network-Essentials

The **Network-Essentials** technology package includes the baseline features. It also supports security features.

The **Network-Essentials_npe** technology package (npe = No Payload Encryption) includes all the features in the Network-Essentials technology package without the payload encryption functionality. This is to fulfill export restriction requirements. The Network-Essentials_npe is available only in the Network-Essentials_npe image. The difference in features between the Network-Essentials package and the Network-Essentials_npe package is therefore the set of payload encryption features such as IPsec and Secure VPN.

Network-Advantage

The **Network-Advantage** technology package includes all crypto features.

The **Network-Advantage_npe** package (npe = No Payload Encryption) includes all the features in the **Network-Advantage** technology package without the payload-encryption functionality. This is to fulfill export restriction requirements. The **Network-Advantage_npe** package is available only in the **Network-Advantage_npe** image. The difference in features between the **Network-Advantage** package and the **Network-Advantage_npe** package is therefore the set of payload-encryption-enabling features such as IPsec and Secure VPN.

Related Documentation

For further information on software licenses, see the Smart Licensing chapter.

Installing the Cisco IOS XE Release

When the device boots up with Cisco IOS XE image for the first time, the device checks the installed version of the ROMMON, and upgrades if the system is running an older version. During the upgrade, do not power cycle the device. The system automatically power cycles the device after the new ROMMON is installed. After the installation, the system will boot up with the Cisco IOS XE image as normal.

Note When the device boots up for first time and if the device requires an upgrade, the entire boot process may take several minutes. This process will be longer than a normal boot due to the ROMMON upgrade.

The following example illustrates the boot process of a consolidated package:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#

Router#show running-config | i license
no license feature hseck9
license udi pid IR8340-K9 sn FDO2523J6N1
license boot level network-advantage
Router#

Router#reload
System configuration has been modified. Save? [yes/no]: yes
Building configuration...
[OK]
Proceed with reload? [confirm]
Sep 8 10:28:36.135: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: process exit
 with reload chassis code

Initializing Hardware ...
Disable swap drive feature. Skip swap drive checking !!

System Bootstrap, Version v0.33, DEVELOPMENT SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.
Compiled Mon Jul 12 22:57:23 2021 by gilchen

Current image running : Boot ROM1

Last reset cause (0x00000002): LocalSoft
IR8340-K9 platform with 8388608 Kbytes of main memory

.....
```

ROMMON Images

A ROMMON image is a software package used by ROM Monitor (ROMMON) software on a router. The software package is separate from the consolidated package normally used to boot the router.

An independent ROMMON image (software package) may occasionally be released and the router can be upgraded with the new ROMMON software. For detailed instructions, see the documentation that accompanies the ROMMON image.



Note A new version of the ROMMON image is not necessarily released at the same time as a consolidated package for a router.

File Systems

The following table provides a list of file systems that can be seen on the Cisco IR8340 router.

Table 6: Router File Systems

File System	Description
bootflash:	Boot flash memory file system.
cns:	Cisco Networking Services file directory.
crashinfo:	Directory or Filename
flash:	Alias to the boot flash memory file system above.
null:	Directory or Filename
nvram:	Router NVRAM. You can copy the startup configuration to NVRAM or from NVRAM.
sdcard:	SD card file system.
system:	System memory file system, which includes the running configuration.
tar:	Archive file system.
tmpsys:	Temporary system files file system.
usb:	USB file system.
webui:	Web UI file system.

Use the **?** help option if you find a file system that is not listed in the table above.

Option to Enable or Disable USB Access

USB flash drives offer inexpensive and easy storage space for the routers to store the images, configuration files and other files.



Note The IR8340 does not support ext3, ext4, and above for USB flash drives.

The IR8340 supports hot plug/unplug of USB flash drives. To access the USB flash drive, insert the device into router's USB interface.

While hot plug/unplug of a USB flash drive is supported, the functionality comes with security vulnerabilities. To prevent users from copying sensitive information to the USB flash drive, USB enable/disable functionality has been added.

By default, the USB flash drive is enabled. If you want to disable USB, use the following disable command:

```
Router#config terminal
Router(config)#platform usb disable
Router(config)#end
```

Once the USB flash drive has been disabled, the file system is not shown on the device and syslog messages will not be displayed when the USB is inserted. You will not be able to access the contents of the USB.

The USB is enabled by issuing a '**no** ' with the disable command:

```
Router#config terminal
Router(config)#no platform usb disable
Router(config)#end
```

The USB status can be displayed using the following command:

```
Router#show platform usb status
USB enabled
Router#
```

Autogenerated File Directories and Files

This section discusses the autogenerated files and directories that can be created, and how the files in these directories can be managed.

Table 7: Autogenerated Files

File or Directory	Description
crashinfo files	Crashinfo files may appear in the bootflash: file system. These files provide descriptive information of a crash and may be useful for tuning or troubleshooting purposes. However, the files are not part of router operations, and can be erased without impacting the functioning of the router.
core directory	The storage area for .core files. If this directory is erased, it will automatically regenerate itself at bootup. The .core files in this directory can be erased without impacting any router functionality, but the directory itself should not be erased.
managed directory	This directory is created on bootup if a system check is performed. Its appearance is completely normal and does not indicate any issues with the router.
tracelogs directory	The storage area for trace files. Trace files are useful for troubleshooting. If the Cisco IOS process fails, for instance, users or troubleshooting personnel can access trace files using diagnostic mode to gather information related to the Cisco IOS failure. Trace files, however, are not a part of router operations, and can be erased without impacting the router's performance.

Important Notes About Autogenerated Directories

Important information about autogenerated directories include:

- Autogenerated files on the bootflash: directory should not be deleted, renamed, moved, or altered in any way unless directed by Cisco customer support.

Note Altering autogenerating files on the bootflash: may have unpredictable consequences for system performance.

- Crashinfo files and files in the core and tracelogs directory can be deleted.

Flash Storage

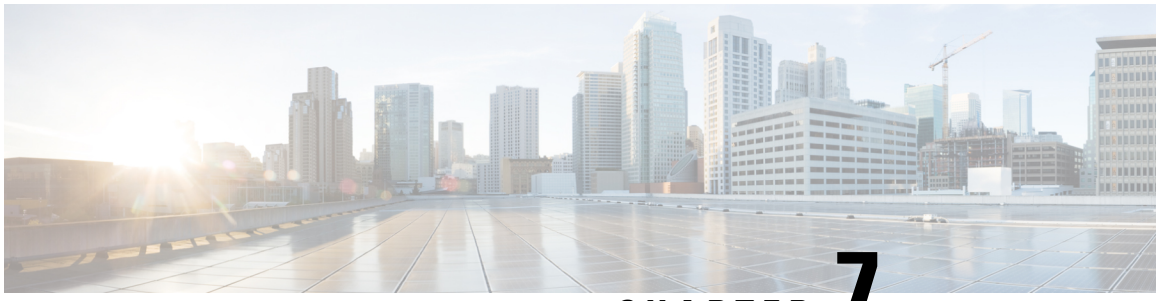
Subpackages are installed to local media storage, such as flash. For flash storage, use the **dir bootflash:** command to list the file names.

Note Flash storage is required for successful operation of a router.

Related Documentation

For further information on software licenses, see Smart Licensing Using Policy (SLP), on page 69.

For further information on obtaining and installing feature licenses, see Configuring the Cisco IOS Software Activation Feature.



CHAPTER 7

Software Maintenance Upgrade (SMU)

Note SMU installation was supported in both bundle boot and install mode. From Cisco IOS XE Release 17.9.x, SMU installation will be stopped if the router is booted up in bundle mode. If the router is booted up in install mode, SMU installation will keep working as it is in previous releases.

- Software Maintenance Upgrade (SMU), on page 61
- SMU Workflow and Basic Requirements, on page 62
- SMU Example, on page 62
- Installing a Patch Image, on page 62
- Uninstalling the Patch Image, on page 64

Software Maintenance Upgrade (SMU)

The Software Maintenance Upgrade (SMU) is a package that can be installed on a system to provide a patch fix or security resolution to a released image for a specific defect in order to respond to immediate issues. It does not contain new features.

Some of the caveats of the SMU are:

- Provided on a per release, per component basis and is specific to the platform. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.
- SMUs are not an alternative to maintenance releases. All defects fixed by SMUs are then automatically integrated into the upcoming maintenance releases.
- The Cisco IOS XE platform internally validates the SMU compatibility and does not allow you to install non-compatible SMUs. This is based on rules/limitations for a SMU change-set.
- An SMU provides a significant benefit over classic IOS software as it allows you to address the network issue quickly while reducing the time and scope of the testing required.
- SMU is a method to fix bugs in an existing release, and allows the application of a PSIRT fix in an existing release
- SMU is NOT an upgrade path from release X to maintenance release X.1
- SMU is NOT an upgrade path from release X to release Y

The device only supports “Hot Patching”. This means:

- The running image is modified in-place or in-service

- This avoids downtime and interruption of service
- The updated code to fix the defect is written in a different location, and where the patch redirects the program run

SMU Workflow and Basic Requirements

The workflow for the patch requires that you complete the following sequence of operation in exec mode:

1. Addition of the SMU to the file system.
2. Activation of the SMU onto the system.
3. Committing the SMU change.
4. Removal and uninstallation of the SMU.

The basic requirements for SMU are:

- The image where the defect was discovered.
- The patch file that contains the fix for the defect must be formatted as ir8340-image_name.release_version.CSCxxyyyyy.SPA.smu.bin.

SMU Example

This section shows an example of a patch created as a test. Your patch will have a name associated with a CDET to be installed as a fix.

Installing a Patch Image

Perform the following steps to install the patch image:

Procedure

Step 1 Show a standard command.

```
Router#show power
Main PSU :
    Total Power Consumed: 11.37 Watts
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 30 Watts
Router#
```

Step 2 Add the image.

```
Router# install add file
sdcard:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bininstall_add: START
Wed Aug 11 17:05:59 UTC 2021
Copying sdcard:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin to
bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin
install_add: Adding SMU
install_add: Checking whether new add is allowed ....

--- Starting SMU Add operation ---
Performing SMU_ADD on Active/Standby
[1] SMU_ADD package(s) on R0
[1] Finished SMU_ADD on R0
Checking status of SMU_ADD on [R0]
SMU_ADD: Passed on [R0]
Finished SMU Add operation

SUCCESS: install_add Wed Aug 11 17:06:22 UTC 2021

Router#
```

Step 3 Activate the patch image.

```
Router# install activate file
bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bininstall_activate:
 START Thu Aug  6 11:53:59 PDT 2020
install_activate: START Wed Aug 11 17:06:35 UTC 2021

System configuration has been modified.
Press Yes(y) to save the configuration and proceed.
Press No(n) for proceeding without saving the configuration.
Press Quit(q) to exit, you may save configuration and re-enter the command. [y/n/q]
167: +++ SLE_Sumatra: receive +++
y

Building configuration...
[OK]Modified configuration has been saved
install_activate: Activating SMU

This operation may require a reload of the system. Do you want to proceed? [y/n]
```

Step 4 Commit the installation.

```
Router# install commitinstall_commit: START Wed Aug 11 17:12:40 UTC 2021
install_commit: Committing SMU
Executing pre scripts....
Executing pre sripts done.
--- Starting SMU Commit operation ---
Performing SMU_COMMIT on Active/Standby
[1] SMU_COMMIT package(s) on R0
[1] Finished SMU_COMMIT on R0
Checking status of SMU_COMMIT on [R0]
SMU_COMMIT: Passed on [R0]
Finished SMU Commit operation

SUCCESS: install_commit
/bootflash/ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin Wed Aug 11
17:13:24 UTC 2021
Router#
```

Step 5 Show the status summary of the installation procedure.

```
Router# show install summary[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
```

```
C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type St Filename/Version
--------------------------------------------------------------------------------
SMU C bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin
IMG C 17.07.01.0.139972

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------

Router#
```

Step 6 Verify the result of the patch by showing the same command.

```
Router#show power
Main PSU :
    Total Power Consumed: 11.04 Watts
Device HOT SMU works!

    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 0 Watts
Router#
```

Uninstalling the Patch Image

There are two methods to remove or uninstall the patch image.

- Restoring the image to its original version by using the following command:
  - **install rollback to base**
- Specific removal of a patch by using the following commands in sequence:
  - **install deactivate file flash:**<*file*>
  - **install commit**
  - **install remove file flash:**<*file*>

Uninstalling the Patch Image Using Rollback

This section shows an example of using the rollback method.

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type St Filename/Version
--------------------------------------------------------------------------------
```

```
SMU C bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin
IMG C 17.07.01.0.139972

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------

Router#
```

The following commands are available:

```
Router# install ?
  abort             Abort the current install operation
  activate          Activate an installed package
  add               Install a package file to the system
  auto-abort-timer  Install auto-abort-timer
  commit            Commit the changes to the loadpath
  deactivate        Deactivate an install package
  label             Add a label name to any installation point
  prepare           Prepare package for operation
  remove            Remove installed packages
  rollback          Rollback to a previous installation point
Router# install rollback to ?
  base       Rollback to the base image
  committed  Rollback to the last committed installation point
  id         Rollback to a specific install point id
  label      Rollback to a specific install point label
```

The **install rollback to base** command removes the entire patch and returns to the base image version with the found defect.

```
Router# install rollback to base

install_rollback: START Thu Aug  6 12:04:04 PDT 2020
install_rollback: Rolling back SMU
Executing pre scripts....
Executing pre sripts done.

--- Starting SMU Rollback operation ---
Performing SMU_ROLLBACK on Active/Standby
  [1] SMU_ROLLBACK package(s) on R0
  [1] Finished SMU_ROLLBACK on R0
Checking status of SMU_ROLLBACK on [R0]
SMU_ROLLBACK: Passed on [R0]
Finished SMU Rollback operation

CSCxx12345:SUCCESS
SUCCESS: install_rollback
/flash1/ir8300-universalk9.2020-08-06_10.38_shchang2.0.CSCxx12345.SSA.smu.bin Thu Aug  6
12:04:57 PDT 2020
Router#
```

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type St Filename/Version
--------------------------------------------------------------------------------
SMU C bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin
IMG C 17.07.01.0.139972
```

```
--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------

Router#
```



Note In the above command output, the patch has been removed and the device returns to the base image version prior to the upgrade.

Uninstalling the Patch Image Using Deactivate, Commit, and Remove

Show what patches are installed.

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type St Filename/Version
--------------------------------------------------------------------------------
SMU C bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin
IMG C 17.07.01.0.139972

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

Procedure

Step 1 Deactivate the patch.

```
Router# install deactivate file
bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin
install_deactivate: START Wed Aug 11 17:14:35 UTC 2021

System configuration has been modified.
Press Yes(y) to save the configuration and proceed.
Press No(n) for proceeding without saving the configuration.
Press Quit(q) to exit, you may save configuration and re-enter the command. [y/n/q]
332: +++ SLE_Sumatra: receive +++
y

Building configuration...
[OK]Modified configuration has been saved
install_deactivate: Deactivating SMU

This operation may require a reload of the system. Do you want to proceed? [y/n]
```

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
```

```
Type St Filename/Version
--------------------------------------------------------------------------------
SMU D bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin
IMG C 17.07.01.0.139972

--------------------------------------------------------------------------------
Auto abort timer: active on install_deactivate, time before rollback - 01:55:21
--------------------------------------------------------------------------------
```

Step 2 Commit the action.

```
Router# install commit
install_commit: START Wed Aug 11 17:21:23 UTC 2021
install_commit: Committing SMU
Executing pre scripts....
Executing pre sripts done.
--- Starting SMU Commit operation ---
Performing SMU_COMMIT on Active/Standby
[1] SMU_COMMIT package(s) on R0
[1] Finished SMU_COMMIT on R0
Checking status of SMU_COMMIT on [R0]
SMU_COMMIT: Passed on [R0]
Finished SMU Commit operation

SUCCESS: install_commit
/bootflash/ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin Wed Aug 11
17:21:53 UTC 2021
```

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type St Filename/Version
--------------------------------------------------------------------------------
SMU I bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin
IMG C 17.07.01.0.139972

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

Step 3 Remove the patch.

```
Router# install remove file
bootflash:ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bininstall_remove:
START Wed Aug 11 17:22:03 UTC 2021
install_remove: Removing SMU
Executing pre scripts....
Executing pre sripts done.

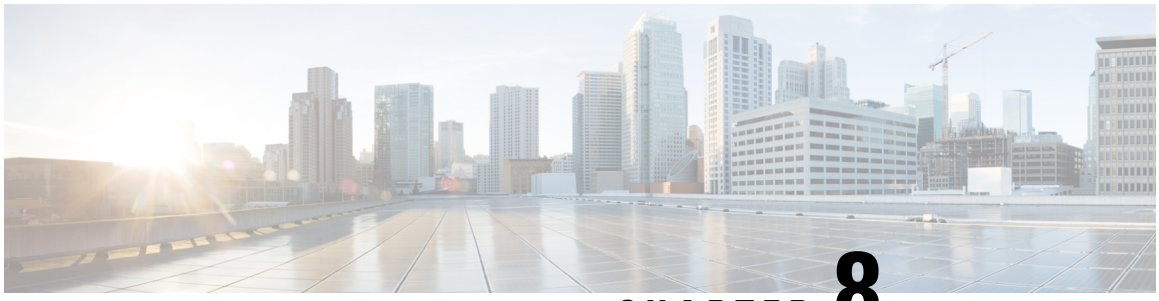
--- Starting SMU Remove operation ---
Performing SMU_REMOVE on Active/Standby
[1] SMU_REMOVE package(s) on R0
[1] Finished SMU_REMOVE on R0
Checking status of SMU_REMOVE on [R0]
SMU_REMOVE: Passed on [R0]
Finished SMU Remove operation

SUCCESS: install_remove
/bootflash/ir8340-universalk9.2021-06-14_14.55_suraiyer.0.cold.SSA.smu.bin Wed Aug 11
17:22:20 UTC 2021
```

Show what patches are installed:

```
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type St Filename/Version
--------------------------------------------------------------------------------
IMG C 17.07.01.0.139972

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```



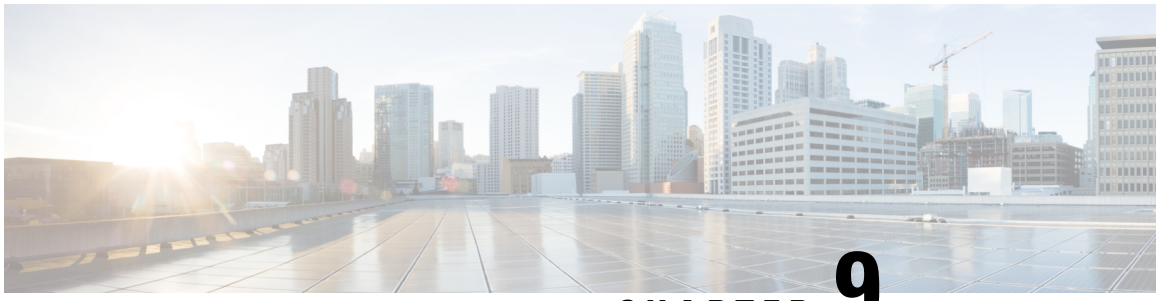
CHAPTER 8

Smart Licensing Using Policy (SLP)

- Smart licensing using policy on Cisco routers, on page 69

Smart licensing using policy on Cisco routers

The Smart Licensing Using Policy chapter has been replaced by a standalone guide called Smart Licensing Using Policy on the Cisco Catalyst IR1101, IR1800, IR8140, IR8340 and ESR6300 Routers.



CHAPTER 9

Configuring Ethernet Switch Ports

- Configuring VLANs, on page 71
- Configuring Private VLANs, on page 75
- Configuring Voice VLANs, on page 83
- Configuring VXLAN Tunneling, on page 85
- IEEE 802.1x Protocol, on page 86
- Spanning Tree Protocol Overview, on page 90
- MAC Table Manipulation, on page 98
- Assigning IP Addresses to Switch Virtual Interfaces, on page 101
- SVI Supported Features, on page 102
- IGMP Snooping for IPv4, on page 104
- IGMP Filtering and Throttling, on page 104
- MLD Snooping, on page 108
- Configuring UniDirectional Link Detection, on page 111
- Configuring the Switched Port Analyzer, on page 112
- EtherChannel Overview, on page 126
- Modular Quality of Service Command-Line Interface, on page 133

Configuring VLANs

A VLAN is a switched network that is logically segmented by function or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs. However, you can group end-stations even if they are not physically located on the same LAN segment. Any device port can belong to a VLAN, unicast, broadcast, and multicast packets are forwarded and flooded only to end-stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router or a device supporting fallback bridging. In a device stack, VLANs can be formed with ports across the stack. Because a VLAN is considered a separate logical network, it contains its own bridge Management Information Base (MIB) information and can support its own implementation of spanning tree.

VLANs are often associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. Interface VLAN membership on the device is assigned manually on an interface-by-interface basis. When you assign device interfaces to VLANs by using this method, it is known as interface-based, or static, VLAN membership.

The device can route traffic between VLANs by using device virtual interfaces (SVIs). An SVI must be explicitly configured and assigned an IP address to route traffic between VLANs.

Access Ports

An access port belongs to and carries the traffic of only one VLAN (unless it is configured as a voice VLAN port). Traffic is received and sent in native formats with no VLAN tagging. Traffic arriving on an access port is assumed to belong to the VLAN assigned to the port. If an access port receives a tagged packet IEEE 802.1Q tagged), the packet is dropped, and the source address is not learned.

Trunk Ports

A trunk port carries the traffic of multiple VLANs and by default is a member of all VLANs in the VLAN database. These trunk port types are supported:

- An IEEE 802.1Q trunk port supports simultaneous tagged and untagged traffic. An IEEE 802.1Q trunk port is assigned a default port VLAN ID (PVID), and all untagged traffic travels on the port default PVID. All untagged traffic and tagged traffic with a NULL VLAN ID are assumed to belong to the port default PVID. A packet with a VLAN ID equal to the outgoing port default PVID is sent untagged. All other traffic is sent with a VLAN tag.

Although by default, a trunk port is a member of every VLAN known to the VTP, you can limit VLAN membership by configuring an allowed list of VLANs for each trunk port. The list of allowed VLANs does not affect any other port but the associated trunk port. By default, all possible VLANs (VLAN ID 1 to 4094) are in the allowed list. A trunk port can become a member of a VLAN only if VTP knows of the VLAN and if the VLAN is in the enabled state. If VTP learns of a new, enabled VLAN and the VLAN is in the allowed list for a trunk port, the trunk port automatically becomes a member of that VLAN and traffic is forwarded to and from the trunk port for that VLAN. If VTP learns of a new, enabled VLAN that is not in the allowed list for a trunk port, the port does not become a member of the VLAN, and no traffic for the VLAN is forwarded to or from the port.

For more information on VLANs, see VLAN Configuration Guide, Cisco IOS XE Gibraltar 16.10.x.

Creating a VLAN

With VTP version 1 and 2, if the device is in VTP transparent mode, you can assign VLAN IDs greater than 1006, but they are not added to the VLAN database.

To configure the Vlan, perform these steps. You can configure the Vlan in access or trunk mode. The procedure is same for the both the modes.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**vlan** *vlan-id* **Example:** `Router(config)# vlan 20`	Enters a VLAN ID, and enters VLAN configuration mode. Enter a new VLAN ID to create a VLAN, or enter an existing VLAN ID to modify that VLAN.

	Command or Action	Purpose
		Note The available VLAN ID rang
Step 3	**name** *vlan-name* **Example:** `Router(config-vlan)# name test20`	(Optional) Enters a name for the VLAN. If no name is entered for the VLAN, the default is to append the *vlan-id* value with leading zeros to the word VLAN. For example, VLAN0004 is a default VLAN name for VLAN 4.
Step 4	**exit** **Example:** `Router(config-vlan)# exit`	Returns to configuration mode.
Step 5	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Specifies the physical port to be configured, and enter interface configuration mode.
Step 6	**switchport mode access** **Example:** `Router(config-if)# switchport mode access`	Configures the interface as a VLAN access port.
Step 7	**switchport access vlan** *vlan id* **Example:** `Router(config-if)# switchport access vlan 20`	Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic.
Step 8	**end** **Example:** `Router(config-if)# end`	Returns to configuration mode.

Configuring LAN Ports for Layer 2 Switching

This section describes how configure all three types of ethernet LAN ports for Layer 2 switching on the Cisco IR8340 routers. The configuration tasks in this section apply to LAN ports on the router.

Layer 2 LAN Port Modes

The following table lists the Layer 2 LAN port modes and describes how they function on LAN ports.

Table 8: Layer 2 LAN Port Modes

Mode	Function
switchport mode access	Puts the LAN port into permanent nontrunking mode and negotiates to convert the link into a nontrunk link. The LAN port becomes a nontrunk port even if the neighboring LAN port does not agree to the change.
switchport mode dynamic desirable	Makes the LAN port actively attempt to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to **trunk** , **desirable** , or **auto** mode. This is the default mode for all LAN ports.
switchport mode dynamic auto	Makes the LAN port willing to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to **trunk** or **desirable** mode.
switchport mode trunk	Puts the LAN port into permanent trunking mode and negotiates to convert the link into a trunk link. The LAN port becomes a trunk port even if the neighboring port does not agree to the change.
switchport nonegotiate	Puts the LAN port into permanent trunking mode but prevents the port from generating DTP frames. You must configure the neighboring port manually as a trunk port to establish a trunk link.

Note DTP is a point-to-point protocol. However, some internetworking devices might forward DTP frames improperly. To avoid this problem, ensure that LAN ports connected to devices that do not support DTP are configured with the **access** keyword if you do not intend to trunk across those links. To enable trunking to a device that does not support DTP, use the **nonegotiate** keyword to cause the LAN port to become a trunk but not generate DTP frames.

Default Layer 2 LAN Interface Configuration

The following table shows the Layer 2 LAN port default configuration.

Table 9: Layer 2 LAN Interface Default Configuration

Feature	Default
Interface mode:	
• Before entering the **switchport** command	
• After entering the **switchport** command	**switchport mode dynamic desirable**

Feature	Default
Default access VLAN	VLAN 1
Native VLAN (for 802.1Q trunks)	VLAN 1

Configuring LAN Interfaces for Layer 2 Switching

These sections describe how to configure Layer 2 switching on the Cisco IR8340 routers:

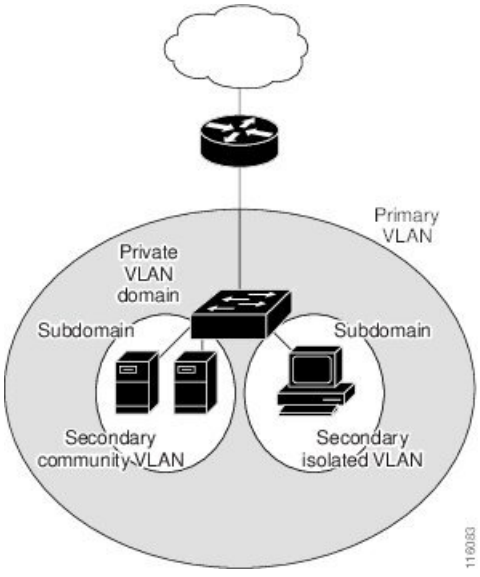
Note Use the default **default interface** *interface -type slot/subslot/port* command to revert an interface to its default configuration.

Configuring Private VLANs

Information About Private VLANs

Using private VLANs addresses the scalability problem and provides IP address management benefits for service providers and Layer 2 security for customers. Private VLANs partition a regular VLAN domain into subdomains. A subdomain is represented by a pair of VLANs: a primary VLAN and a secondary VLAN. A private VLAN can have multiple VLAN pairs, one pair for each subdomain. All VLAN pairs in a private VLAN share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another.

Figure 1: Private VLAN Domain



Primary and Secondary VLANs in Private VLANs

A private VLAN domain has only one primary VLAN. Each port in a private VLAN domain is a member of the primary VLAN; the primary VLAN is the entire private VLAN domain.

Secondary VLANs provide isolation between ports within the same private VLAN domain. The following two types are secondary VLANs within a primary VLAN:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate directly with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other community VLANs or in any isolated VLANs at the Layer 2 level.

Private VLAN Ports

The three types of PVLAN ports are as follows:

- Promiscuous port—A promiscuous port belongs to the primary VLAN. The promiscuous port can communicate with all interfaces, including the community and isolated host ports, that belong to those secondary VLANs associated to the promiscuous port and associated with the primary VLAN. You can have several promiscuous ports in a primary VLAN. Each promiscuous port can have several secondary VLANs or no secondary VLANs that are associated to that port. You can associate a secondary VLAN to more than one promiscuous port, as long as the promiscuous port and secondary VLANs are within the same primary VLAN. You may want to do this for load-balancing or redundancy purposes. You can also have secondary VLANs that are not associated to any promiscuous port.

  A promiscuous port can be configured as an access port.

- Isolated port—An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete isolation from other ports within the same PVLAN domain, except that it can communicate with associated promiscuous ports. PVLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports. You can have more than one isolated port in a specified isolated VLAN. Each port is completely isolated from all other ports in the isolated VLAN.

  An isolated port can be configured an access port.

- Community port—A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with associated promiscuous ports. These interfaces are isolated from all other interfaces in other communities and from all isolated ports within the PVLAN domain.

  A community port must be configured as an access port.

Guidelines and Limitations for Private VLANs

The following are guidelines and limitations for configuring Private VLANs on IR8340:

- A primary VLAN can have only one isolated VLAN and multiple community VLANs.
- Secondary VLANs must be part of one and only one primary VLAN.
- SVI for secondary VLAN is disabled when primary-secondary VLAN association is configured. Deleting the association will bring back the secondary VLAN SVI.
- Deletion of a secondary VLAN puts the ports in that VLAN in inactive state.

- VLANs that cannot be configured as PVLAN—1, 1001-1005
- All switches in the network must be manually configured with the primary-secondary VLAN association. Otherwise, the MAC addresses will not be replicated from primary VLAN to secondary and vice versa, in that switch. That will lead to flooding of PVLAN traffic.
- Maximum number of primary or secondary VLANs that can be configured is limited by the number of VLANs that can be supported by the switch.
- Maximum number of end devices that can be configured in PVLAN is limited by the L2 TCAM entry limitation.

The following features are supported on IR8340:

- **Isolated access port**—Access port which can only communicate with promiscuous port
- **Promiscuous access port**—Access ports which can communicate with all ports in private VLAN
- **Community access port**—Access ports which can communicate with ports in same community and promiscuous ports
- **Private VLAN across switches**—Private VLAN traffic can be carried across normal trunk ports and the feature can span across switches
- **Multicast in Private VLAN**—Multicast communication in and out of private VLAN

The following features are not supported on IR8340:

- **2-way community VLAN**—The community ports send and receive traffic in the same VLAN
- **Promiscuous trunk port**—A trunk port carrying primary VLAN traffic for multiple private VLAN. The secondary VLANs are explicitly mapped to primary VLAN for multiple private VLAN
- **Trunk isolated/community ports**—Isolated and community ports are trunk with secondary VLANs of multiple private VLAN
- **Layer 3 communication between isolated ports**—Isolated ports can communicate at layer 3

Configuring a Private VLAN

Configuring a VLAN as a Private VLAN

To create a private VLAN, you first create a VLAN, and then configure that VLAN to be a private VLAN.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**vlan** *vlan-id* **Example:** `Router(config)# vlan 202`	Enters VLAN configuration submode.
Step 3	[**no**] **private-vlan** {**community** | **isolated** | **primary**} **Example:** `Router(config-vlan)# private-vlan primary`	Configures a VLAN as a private VLAN. Use the no form of this command to clear the private VLAN configuration. **Note** These commands will not take e
Step 4	**end** **Example:** `Router(config-vlan)# end`	Exits VLAN configuration mode.
Step 5	**show vlan private-vlan** [*type*] **Example:** `Router# show vlan private-vlan`	Verifies the configuration. **Note** When a VLAN is configured as will be brought down, waiting fo

Example

The following example shows how to configure VLAN 202 as a primary VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 202
Router(config-vlan)# private-vlan primary
Router(config-vlan)# end
Router# show vlan private-vlan
Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
```

The following example shows how to configure VLAN 303 as a community VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 303
Router(config-vlan)# private-vlan community
Router(config-vlan)# end
Router# show vlan private-vlan
Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
        303       community
```

The following example shows how to configure VLAN 440 as an isolated VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 440
Router(config-vlan)# private-vlan isolated
Router(config-vlan)# end
Router# show vlan private-vlan
Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
        303       community
        440       isolated
```

Associating Secondary VLANs with a Primary Private VLAN

To associate secondary VLANs with a primary VLAN, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vlan** *primary-vlan-id* **Example:** `Router(config)# vlan 202`	Enters VLAN configuration submode for the primary VLAN.
Step 3	**private-vlan association** {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*} **Example:** `Router(config-vlan)# private-vlan association 303-307,309,440`	Associates the secondary VLANs with the primary VLAN. Use **no private-vlan association** to clear all secondary VLAN associations.
Step 4	**end** **Example:** `Router(config-vlan)# end`	Exits VLAN configuration mode.
Step 5	**show vlan private-vlan** [*type*] **Example:** `Router# show vlan private-vlan`	Verifies the configuration.

Example

The following example shows how to associate community VLANs 303 through 307, 309, and isolated VLAN 440 with primary VLAN 202, and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 202
Router(config-vlan)# private-vlan association 303-307,309,440
Router(config-vlan)# end
Router# show vlan private-vlan
Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202     303       community
202     304       community
202     305       community
202     306       community
202     307       community
202     309       community
202     440       isolated
        308       community
```

Configuring a Layer 2 Interface as a Private VLAN Host Port

To configure a Layer 2 interface as a private VLAN host port, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type slot/port* **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Selects the LAN interface to configure.
Step 3	**switchport** **Example:** `Router(config-if)# switchport`	Configures the LAN interface for Layer 2 switching.
Step 4	**switchport mode private-vlan host** **Example:** `Router(config-if)# switchport mode private-vlan host`	Configures the Layer 2 port as a private VLAN host port. Use **no switchport mode private-vlan host** to clears private VLAN port configuration.
Step 5	**switchport private-vlan host host-association** *primary_vlan_ID secondary_vlan_ID* **Example:** `Router(config-if)# switchport private-vlan host-association 202 303`	Associates the Layer 2 port with a private VLAN. Use **no switchport private-vlan host host-association** to clear the association.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode.

	Command or Action	Purpose
Step 7	**show interface** *type slot/port* **switchport** **Example:** `Router# show interface gigabitethernet 0/1/0 switchport`	Verifies the configuration.

Example

This example shows how to configure interface gigabitethernet 0/1/0 as a private VLAN host port and verify the configuration:

```
Router# configure terminal
Router(config)# interface gigabitethernet 0/1/0
Router(config-if)# switchport mode private-vlan host
Router(config-if)# switchport private-vlan host-association 202 303
Router(config-if)# end
Router# show interfaces gigabitethernet 0/1/0 switchport
Name: Ge0/1/0
Switchport: Enabled
 Administrative Mode: private-vlan host
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
 Administrative private-vlan host-association: 202 (VLAN0202) 303 (VLAN0303)
Administrative private-vlan mapping: none
 Operational private-vlan: 202 (VLAN0202)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port

To configure a Layer 2 interface as a private VLAN promiscuous port, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type slot/port* **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Selects the LAN interface to configure.
Step 3	**switchport** **Example:**	Configures the LAN interface for Layer 2 switching.

	Command or Action	Purpose
	`Router(config-if)# switchport`	
Step 4	**switchport mode private-vlan promiscuous** **Example:** `Router(config-if)# switchport mode private-vlan promiscuous`	Configures the Layer 2 port as a private VLAN promiscuous port. Use **no switchport mode private-vlan** to clear the private VLAN port configuration.
Step 5	**switchport private-vlan mapping** *primary_vlan_ID* {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*} *secondary_vlan_ID* **Example:** `Router(config-if)# switchport private-vlan mapping 202 303,440`	Maps the private VLAN promiscuous port to a primary VLAN and to selected secondary VLANs. Use **no switchport private-vlan mapping** to clear all mapping between the private VLAN promiscuous port and the primary VLAN and any secondary VLANs.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode.
Step 7	**show interface** *type slot/port* **switchport** **Example:** `Router# show interface gigabitethernet 0/1/0 switchport`	Verifies the configuration.

Example

The following example shows how to configure interface gigabitethernet 0/1/0 as a private VLAN promiscuous port and map it to a private VLAN:

```
Router# configure terminal
Router(config)# interface gigabitethernet 0/1/0
Router(config-if)# switchport mode private-vlan promiscuous
Router(config-if)# switchport private-vlan mapping 202 303,440
Router(config-if)# end
```

The following example shows how to verify the configuration:

```
Router# show interfaces gigabitethernet 0/1/0 switchport
Name: Ge0/1/0
Switchport: Enabled
 Administrative Mode: private-vlan promiscuous
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: none ((Inactive))
 Administrative private-vlan mapping: 202 (VLAN0202) 303 (VLAN0303) 440 (VLAN0440)
 Operational private-vlan: 202 (VLAN0202)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

Configuring Voice VLANs

The voice VLAN feature provides support for connecting an IP phone to an access switch port. Voice VLAN on an access port is desirable so that feature like port security, dot1x, dynamic access port, protected port can be configured.

The voice VLAN requires the access port to support dedicated VLAN for voice traffic (as the data traffic on the phone link might deteriorate the voice traffic quality) thus the device can differentiate voice traffic from data traffic and provide QoS for voice traffic and ensure quality.

The Ethernet port will be associated with two VLANs on a voice VLAN port as following:

1. A native VLAN to carry data traffic
2. An auxiliary or Voice VLAN to carry voice traffic

The data traffic will be sent either tagged or untagged with the access VLAN id. The phone will send voice traffic tagged with configured voice VLAN id. The voice VLAN id used by phone can either configured manually or learned through CDP. When voice VLAN is configured on the access port, the device will instruct IP phone to send voice traffic over the configured voice VLAN. This is achieved through sending CDP messages to IP phone indicating the same. QoS configurations can be done on the voice VLAN port in order to provide predictable forwarding of voice traffic and thus ensure voice quality.

Limitations and Restrictions

- Voice VLAN configuration is only supported on device access ports.
- Voice VLAN configuration will not be applicable to port channels.
- Private VLAN configuration will not be allowed on a voice VLAN port and vice versa.

How to Configure Voice VLANs

The following sections provide information about configuring Voice VLANs:

Configuring Cisco IP Phone Voice Traffic

You can configure a port connected to the Cisco IP Phone to send CDP packets to the phone to configure the way in which the phone sends voice traffic. The phone can carry voice traffic in IEEE 802.1Q frames for a specified voice VLAN with a Layer 2 CoS value. It can use IEEE 802.1p priority tagging to give voice traffic a higher priority and forward all voice traffic through the native (access) VLAN. The Cisco IP Phone can also send untagged voice traffic or use its own configuration to send voice traffic in the access VLAN. In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5).

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 2	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Specifies the interface connected to the phone, and enters interface configuration mode.
Step 3	**switchport voice vlan** {*vlan-id* | **dot1p** | **none** | **untagged**} **Example:** `Router(config-if)# switchport voice vlan dot1p`	Configures the voice VLAN. • *vlan-id* —Configures the phone to forward all voice traffic through the specified VLAN. By default, the Cisco IP Phone forwards the voice traffic with an IEEE 802.1Q priority of 5. Valid VLAN IDs are 1 to 4094. • **dot1p** —Configures the phone to use IEEE 802.1p priority tagging for voice traffic and to use the default native VLAN (VLAN 0) to carry all traffic. By default, the Cisco IP Phone forwards the voice traffic with an IEEE 802.1p priority of 5. • **none** —Allows the phone to use its own configuration to send untagged voice traffic. • **untagged** —Configures the phone to send untagged voice traffic.
Step 4	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode.

What to do next

To configure port security for Voice VLAN:

```
Router#configure terminal
Router(config)#interface <interface-id>
Router(config-if)# switchport mode access
Router(config-if)# switchport voice vlan vlan-id
Router(config-if)# switchport port-security
Router(config-if)# switchport port-security [maximum value [vlan {access | voice}}]]
Router(config-if)# switchport port-security violation {protect | restrict | shutdown}
Router(config-if)# switchport port-security [mac-address mac-address [vlan {access | voice}]]
Router(config-if)#end
Router#show port-security
```

Removing Voice VLAN

To remove the voice VLAN configuration, use the **no switchport voice vlan** command.

Monitoring Voice VLAN

To display voice VLAN configuration for an interface, use the **show interfaces** *interface-id* **switchport** privileged EXEC command.

Configuring VXLAN Tunneling

VXLAN is an extension to the Layer 2 VLAN. It was designed to provide the same VLAN functionality with greater extensibility and flexibility. VXLAN offers the following benefits:

- VLAN flexibility in multitenant segments: It provides a solution to extend Layer 2 segments over the underlying network infrastructure so that tenant workload can be placed across physical pods in the data center.
- Higher scalability: VXLAN uses a 24-bit segment ID known as the VXLAN network identifier (VNID), which enables up to 16 million VXLAN segments to coexist in the same administrative domain.
- Improved network utilization: VXLAN solved Layer 2 STP limitations. VXLAN packets are transferred through the underlying network based on its Layer 3 header and can take complete advantage of Layer 3 routing, equal-cost multipath (ECMP) routing, and link aggregation protocols to use all available paths.

VXLAN uses the VXLAN tunnel endpoint (VTEP) to map tenants' end devices to VXLAN segments and to perform VXLAN encapsulation and decapsulation. Each VTEP function has two interfaces: one is a switch interface on the local LAN segment to support local endpoint communication, and the other is an IP interface to the transport IP network.

Infrastructure VLAN is a unique IP address that identifies the VTEP device on the transport IP network. The VTEP device uses this IP address to encapsulate Ethernet frames and transmits the encapsulated packets to the transport network through the IP interface.

A VTEP device also discovers the remote VTEPs for its VXLAN segments and learns remote MAC Address-to-VTEP mappings through its IP interface.

The following example shows that two IR8340 routers act as VTEPs:



Table 10: VXLAN Configuration

IR8340-1	IR8340-2
bridge-domain 1 member vni 6001 member Vlan100 service-instance 1 ! interface Loopback1 ip address 200.200.200.200 255.255.255.255 ! interface GigabitEthernet0/0/1 ip address 192.168.1.2 255.255.255.0 media-type rj45 ! Interface GigabitEthernet0/1/2 switchport access vlan 100 ! interface Vlan100 no ip address service instance 1 ethernet encapsulation dot1q 100 ! interface nve1 no ip address source-interface Loopback1 member vni 6001 ingress-replication 100.100.100.100 ! ip forward-protocol nd ip pim rp-address 200.200.200.200 ip http server ip http secure-server ip route 0.0.0.0 0.0.0.0 192.168.1.3 !	bridge-domain 1 member vni 6001 member Vlan100 service-instance 1 ! interface Loopback1 ip address 100.100.100.100 255.255.255.255 ! interface GigabitEthernet0/0/1 ip address 192.168.1.3 255.255.255.0 media-type rj45 ! interface GigabitEthernet0/1/2 switchport access vlan 100 ! interface Vlan100 no ip address service instance 1 ethernet encapsulation dot1q 100 ! interface nve1 no ip address source-interface Loopback1 member vni 6001 ingress-replication 200.200.200.200 ! ip forward-protocol nd ip pim rp-address 100.100.100.100 no ip http server ip http secure-server ip route 0.0.0.0 0.0.0.0 192.168.1.2 !

For more details and multiple VTEP configuration with multicast, see https://www.cisco.com/c/en/us/support/docs/ip/multicast/200791-Configuration-and-Troubleshooting-of-VxL.html.

IEEE 802.1x Protocol

The IEEE 802.1x standard defines a client/server-based access control and authentication protocol that prevents clients from connecting to a LAN through publicly accessible ports unless they are authenticated. The authentication server authenticates each client connected to a port before making available any services offered by the router or the LAN.

Until the client is authenticated, IEEE 802.1x access control allows only Extensible Authentication Protocol over LAN (EAPOL), Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (STP) traffic through the port to which the client is connected. After authentication, normal traffic can pass through the port. For more information on IEEE 802.1x port-based authentication, see the Configuring IEEE 802.1x Port-Based Authentication chapter of the *Security Configuration Guide, Cisco IOS XE Gibraltar 16.10.x*.

Configuring IEEE 802.1X Port-Based Authentication

IEEE 802.1X port-based authentication is configured on a device to prevent unauthorized devices (supplicants) from gaining access to the network. The device can combine the function of a router, switch, and access point,

depending on the fixed configuration or installed modules. The switch functions are provided by either built-in switch ports or a plug-in module with switch ports. This feature supports both access ports and trunk ports. For more information on 802.1X port-based authentication, see the Configuring IEEE 802.1X Port-Based Authentication Guide.

Enabling AAA Authorization for VLAN Assignment

AAA authorization limits the services available to a user. When AAA authorization is enabled, the device uses information retrieved from the user's profile, which is in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA.
Step 4	**aaa authorization network radius if-authenticated** **Example:** `Router(config)# aaa authorization network radius if-authenticated`	Configures the device for user RADIUS authorization for all network-related service requests. RADIUS authorization succeeds if the user has authenticated.
Step 5	**aaa authorization exec radius if-authenticated** **Example:** `Router(config)# aaa authorization exec radius if-authenticated`	Configures the device for user RADIUS authorization if the user has privileged EXEC access. RADIUS authorization succeeds if the user has authenticated.
Step 6	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling IEEE 802.1X Authentication and Authorization

Follow these steps to enable IEEE 802.1X authentication and authorization.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**aaa authentication dot1x** {**default** | **listname**} **method1** [**method2**...] **Example:** `Router(config)# aaa authentication dot1x default group radius`	Creates a series of authentication methods that are used to determine user privilege to access the privileged command level so that the device can communicate with the AAA server.
Step 4	**dot1x system-auth-control** **Example:** `Router(config)# dot1x system-auth-control`	Globally enables 802.1X port-based authentication.
Step 5	**identity profile default** **Example:** `Router(config)# identity profile default`	Creates an identity profile and enters dot1x profile configuration mode.
Step 6	**exit** **Example:** `Router(config-identity-prof)# exit`	Exits dot1x profile configuration mode and returns to global configuration mode.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface Gigabitethernet 0/1/0`	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 8	**access-session port-control** {**auto** | **force-authorized** | **force-unauthorized**} **Example:** `Router(config-if)# access-session port-control auto`	Enables 802.1X port-based authentication on the interface. • **auto** —Enables IEEE 802.1X 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 changes from down to up or when an EAPOL-start frame is received. The Device requests the identity

	Command or Action	Purpose
		of the supplicant and begins relaying authentication messages between the supplicant and the authentication server. Each supplicant attempting to access the network is uniquely identified by the Device by using the supplicant MAC address. • **force-authorized** -—Disables IEEE 802.1X authentication and causes the port to change to the authorized state without any authentication exchange required. The port sends and receives normal traffic without IEEE 802.1X-based authentication of the client. This is the default setting. • **force-unauthorized** —Causes the port to remain in the unauthorized state, ignoring all attempts by the supplicant to authenticate. The Device cannot provide authentication services to the supplicant through the port.
Step 9	**dot1x pae** [**supplicant** | **authenticator** | **both**] **Example:** `Device(config-if)# dot1x pae authenticator`	Sets the Port Access Entity (PAE) type. • **supplicant** —The interface acts only as a supplicant and does not respond to messages that are meant for an authenticator. • **authenticator** -—The interface acts only as an authenticator and does not respond to any messages meant for a supplicant. • **both** —The interface behaves both as a supplicant and as an authenticator and thus does respond to all dot1x messages.
Step 10	**end** **Example:** `router(config)# end`	Returns to privileged EXEC mode.
Step 11	**show dot1x** **Example:** `Router# show dot1x`	Displays whether 802.1X authentication has been configured on the device.

Spanning Tree Protocol Overview

Spanning Tree Protocol (STP) is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Device might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

The STP uses a spanning-tree algorithm to select one device of a redundantly connected network as the root of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- Root—A forwarding port elected for the spanning-tree topology
- Designated—A forwarding port elected for every switched LAN segment
- Alternate—A blocked port providing an alternate path to the root bridge in the spanning tree
- Backup—A blocked port in a loopback configuration

The device that has *all* of its ports as the designated role or as the backup role is the root device. The device that has at least *one* of its ports in the designated role is called the designated device.

Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Device send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The device do not forward these frames but use them to construct a loop-free path. BPDUs contain information about the sending device and its ports, including device and MAC addresses, device priority, port priority, and path cost. Spanning tree uses this information to elect the root device and root port for the switched network and the root port and designated port for each switched segment.

When two ports on a device are part of a loop, the spanning-tree and path cost settings control which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed.

Note By default, the device sends keepalive messages (to ensure the connection is up) only on interfaces that do not have small form-factor pluggable (SFP) modules. You can change the default for an interface by entering the [**no**] **keepalive** interface configuration command with no keywords.

IR8340 uses STP (the IEEE 802.1D bridge protocol) on all VLANs. By default, a single instance of STP runs on each configured VLAN (provided you do not manually disable STP). You can enable and disable STP on a per-VLAN basis.

For more information on STP, see the **Configuring Spanning Tree Protocol** chapter of the Layer 2 Configuration Guide.

Default STP Configuration

The following table shows the default STP configuration.

Table 11: STP Default Configuration

Feature	Default Value
Disable state	STP disabled for all VLANs
Bridge priority	32768
STP port priority (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports)	128
STP port cost (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports)	Gigabit Ethernet: 4
STP VLAN port priority (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	128
STP VLAN port cost (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	Gigabit Ethernet:1000000000
Hello time	2 seconds
Forward delay time	15 seconds
Maximum aging time	20 seconds
Mode	PVST

Enabling STP

Note STP is disabled by default on all VLANs.

You can enable STP on a per-VLAN basis. The Cisco SM-X-16G4M2X or SM-X-40G8M2X Layer 2 Gigabit EtherSwitch Service Module maintain a separate instance of STP for each VLAN (except on VLANs on which you disable STP).

If you want to enable a mode that is different from the default mode, this procedure is required.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.

	Command or Action	Purpose
Step 2	**spanning-tree mode** {**pvst** | **mst** | **rapid-pvst**}	Configures a spanning-tree mode. All stack members run the same version of spanning tree. • Select **pvst** to enable PVST+. • Select **mst** to enable MSTP. • Select **rapid-pvst** to enable rapid PVST+.
Step 3	**interface** *interface-id*	Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports, VLANs, and port channels. The VLAN ID range is 1 to 4094. The port-channel range is 1 to 48.
Step 4	**spanning-tree link-type point-to-point** **Example:** `Device(config-if)# spanning-tree link-type point-to-point`	Specifies that the link type for this port is point-to-point. If you connect this port (local port) to a remote port through a point-to-point link and the local port becomes a designated port, the negotiates with the remote port and rapidly changes the local port to the forwarding state.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 6	**clear spanning-tree detected-protocols** **Example:** `Router# clear spanning-tree detected-protocols`	If any port on the device is connected to a port on a legacy IEEE 802.1D device, this command restarts the protocol migration process on the entire device. This step is optional if the designated device detects that this device is running rapid PVST+.
Step 7	**show spanning-tree vlan** *vlan_ID*	Verifies that STP is enabled.

What to do next

Caution Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some switches and bridges in a VLAN and leave it enabled on other switches and bridges in the VLAN. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network.

Caution We do not recommend disabling spanning tree, even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN.

This example shows how to enable STP on VLAN 200:

```
Device# configure terminal
Device(config)# spanning-tree vlan 200
Device(config)# end
Device#
```

Note STP is disabled by default.

This example shows how to verify the configuration:

```
Device# show spanning-tree vlan 200

G0:VLAN0200
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     00d0.00b8.14c8
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32768
             Address     00d0.00b8.14c8
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300
Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
Gi1/4            Desg FWD 200000    128.196  P2p
Gi1/5            Back BLK 200000    128.197  P2p
Device#
```

You must have at least one interface that is active in VLAN 200 to create a VLAN 200 spanning tree. In this example, two interfaces are active in VLAN 200.

Configuring Optional STP Features

This section describes how to configure the following optional STP features:

Enabling PortFast

Caution Use PortFast *only* when connecting a single end station to a Layer 2 access port. Otherwise, you might create a network loop.

To enable PortFast on a Layer 2 access port, perform this task:

Procedure

	Command or Action	Purpose
Step 1	Router(config)# **interface** {*type slot/port*}	Selects a port to configure.
Step 2	Router(config-if)# **spanning-tree portfast**	Enables PortFast on a Layer 2 access port connected to a single workstation or server.
Step 3	Router(config-if)# **spanning-tree portfast default**	Enables PortFast.
Step 4	Router(config-if)# **end**	Exits configuration mode.
Step 5	Router# **show running interface** {*type slot/port*}	Verifies the configuration.

Configuring PortFast BPDU Filtering

These sections describe how to configure PortFast BPDU filtering.

To enable PortFast BPDU filtering globally, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**spanning-tree portfast bpdufilter default** **Example:** `Router(config)# spanning-tree portfast bpdufilter default`	Enables BPDU filtering globally on the router.
Step 2	**show spanning-tree summary totals** **Example:** `Router(config)# show spanning-tree summary totals`	Verifies the configuration.

Enabling PortFast BPDU Filtering

BPDU filtering is set to default on each port. This example shows how to enable PortFast BPDU filtering on the port and verify the configuration in PVST+ mode:

```
Router(config)# spanning-tree portfast bpdufilter default

Router(config)# ^Z
Router# show spanning-tree summary totals

Switch is in pvst mode
Root bridge for: G0:VLAN0013, G0:VLAN0020, G1:VLAN0020
EtherChannel misconfig guard is enabled
Extended system ID           is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
```

```
Loopguard Default            is disabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Pathcost method used         is short
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
3 vlans                       0         0        0          3          3
```

To enable PortFast BPDU filtering on a nontrunking port, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Selects the interface to configure.
Step 2	**spanning-tree bpdufilter enable** **Example:** `Router(config-if)# spanning-tree bpdufilter enable`	Enables BPDU filtering.
Step 3	**show spanning-tree interface** *interface-id* **Example:** `Router# show spanning-tree interface gigabitethernet 0/1/0`	Verifies the configuration.

What to do next

This example shows how to enable PortFast BPDU filtering on a nontrunking port:

```
Router(config)# interface gigabitethernet 0/1/0
Router(config-if)# spanning-tree bpdufilter enable

Router(config-if)# ^Z
Router# show spanning-tree interface gigabitethernet 0/1/0
Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0010         Desg FWD 1000      160.196  Edge P2p
Router# show spanning-tree interface gigabitethernet 0/1/0 detail

 Port 196 (gigabitethernet 0/1/0) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   BPDU:sent 0, received 0
Router#
```

Enabling BPDU Guard

To enable BPDU Guard globally, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**spanning-tree portfast bpduguard default** **Example:** `Router(config)# no spanning-tree portfast bpduguard default`	Enables BPDU Guard globally. Disables BPDU Guard globally.
Step 2	**end** **Example:** `Router(config)# end`	Exits configuration mode.
Step 3	**show spanning-tree summary totals** **Example:** `Router# show spanning-tree summary totals`	Verifies the configuration.

What to do next

This example shows how to enable BPDU Guard:

```
Router# configure terminal
Router(config)# spanning-tree portfast bpduguard
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary totals
 default
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID   is disabled
Portfast             is enabled by default
PortFast BPDU Guard  is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard            is disabled by default
UplinkFast           is disabled
BackboneFast         is disabled
Pathcost method used is long
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0        0         0        3          3
Router#
```

Enabling UplinkFast

UplinkFast increases the bridge priority to 49152 and adds 3000 to the STP port cost of all Layer 2 LAN interfaces on the device, decreasing the probability that the router will become the root bridge. The *max_update_rate* value represents the number of multicast packets transmitted per second (the default is 150 packets per second). UplinkFast cannot be enabled on VLANs that have been configured for bridge priority.

To enable UplinkFast on a VLAN with bridge priority configured, restore the bridge priority on the VLAN to the default value by entering a **no spanning-tree vlan** *vlan_ID* **priority** command in global configuration mode.

Note When you enable UplinkFast, it affects all VLANs on the device. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

Procedure

	Command or Action	Purpose
Step 1	Router(config)# **spanning-tree uplinkfast** [**max-update-rate** *max_update_rate*]	Enables UplinkFast.
Step 2	Router(config)# **no spanning-tree uplinkfast max-update-rate**	Reverts to the default rate.
Step 3	Router(config)# **no spanning-tree uplinkfast**	Disables UplinkFast.
Step 4	Router(config)# **end**	Exits configuration mode.
Step 5	Router# **show spanning-tree vlan** *vlan_ID*	Verifies that UplinkFast is enabled.

What to do next

This example shows how to enable UplinkFast with an update rate of 400 packets per second:

```
Router# configure terminal
Router(config)# spanning-tree uplinkfast max-update-rate 400
Router(config)# exit
Router#
```

This example shows how to verify that UplinkFast is enabled:

```
Router# show spanning-tree uplinkfast

UplinkFast is enabled
Router#
```

Enabling BackboneFast

Note BackboneFast operates correctly only when enabled on all network devices in the network. BackboneFast is not supported on Token Ring VLANs. This feature is supported for use with third-party network devices.

To enable BackboneFast, perform this task:

Procedure

	Command or Action	Purpose
Step 1	Router(config)# **spanning-tree backbonefast**	Enables backbonefast.
Step 2	Router(config)# **no spanning-tree backbonefast**	Disables BackboneFast.
Step 3	Router(config)# **end**	Exits configuration mode.
Step 4	Router# **show spanning-tree vlan** *vlan_ID*	Verifies that BackboneFast is enabled.

What to do next

This example shows how to enable BackboneFast:

```
Router# configure terminal
Router(config)# spanning-tree backboneFast
Router(config)# end
Router#
```

This example shows how to verify that BackboneFast is enabled:

```
Router# show spanning-tree backbonefast

BackboneFast is enabled
BackboneFast statistics
-----------------------
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)     : 0
Number of RLQ request PDUs received (all VLANs)   : 0
Number of RLQ response PDUs received (all VLANs)  : 0
Number of RLQ request PDUs sent (all VLANs)       : 0
Number of RLQ response PDUs sent (all VLANs)      : 0
Router#
```

MAC Table Manipulation

This section includes the following:

Creating a Static Entry in the MAC Address Table

Perform the following task to create a static entry in the MAC address table.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**mac address-table static mac-address vlan** *vlan-id* **interface** *Interface-id* **Example:** `Router(config)# mac address-table static 00ff.ff0d.2dc0 vlan 1 interface gigabitethernet 0/1/0`	Creates a static entry in the MAC address table.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 5	**show mac address-table** **Example:** `Router# show mac address-table`	Verifies the MAC address table.

MAC Address-Based Traffic Blocking

Perform the following task to block all traffic to or from a MAC address in a specified VLAN.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**mac address-table static mac-address vlan** *vlan-id* **drop** **Example:** `Router(config)# mac address-table static 00ff.ff0d.2dc0 vlan 1 drop`	Creates a static entry with drop action in the MAC address table.
Step 4	**end** **Example:** `router(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show mac address-table** **Example:** `Router# show mac address-table`	Verifies the MAC address table.

Configuring and Verifying the Aging Timer

Perform this task to configure the aging timer.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.

Step 3	**mac address-table aging-time** *time*
	Example:
	`Router(config)# mac address-table aging-time 600`
	`Or`
	`Router(config)# mac address-table aging-time 0`
		Configures the MAC address aging timer age in seconds.
		• The accept value is either 0 or 10-1000000 seconds. Default value is 300 seconds.
		• The maximum aging timer supported by switch chipset is 634 seconds. If configure greater than 634 seconds, MAC address will age out after 634 seconds.
		• The value 0 means dynamic MAC entries will never age out.

Step 4	**end** **Example:** `router(config)# end`	Returns to privileged EXEC mode.
Step 5	**show mac address-table aging-time** **Example:** `Router# show mac address-table aging-time`	Verifies the MAC address table.

MAC Learning on a Vlan

To disable or enable MAC learning on specified vlan, perform these steps.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**mac address-table learning vlan** *vlan-id* **interface** *Interface-id* **Example:** `Router(config)# mac address-table learning vlan 10`	Creates a static entry in the MAC address table.
Step 4	**end** **Example:** `router(config)# end`	Returns to privileged EXEC mode.

Assigning IP Addresses to Switch Virtual Interfaces

To configure IP routing, you need to assign IP addresses to Layer 3 network interfaces. This enables communication with the hosts on those interfaces that use IP. IP routing is disabled by default, and no IP addresses are assigned to Switch Virtual Interfaces (SVIs).

An IP address identifies a destination for IP packets. Some IP addresses are reserved for special uses and cannot be used for host, subnet, or network addresses. RFC 1166, “Internet Numbers,” contains the official description of these IP addresses.

An interface can have one primary IP address. A a subnet mask identifies the bits that denote the network number in an IP address.

Beginning in privileged EXEC mode, follow these steps to assign an IP address and a network mask to an SVI.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface vlan** *vlan-id*	Enter interface configuration mode, and specify the Layer 3 VLAN to configure.
Step 3	**ip address** *ip-address subnet-mask*	Configure the IP address and IP subnet mask.

	Command or Action	Purpose
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show interfaces** [*interface-id*] **show ip interface** [*interface-id*] **show running-config interface** [*interface-id*]	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

SVI Supported Features

The following table provided the supported features on the SVI.

Table 12: SVI Supported Features

Techolongy	Feature	Use Case
Routing	Routing Protocol	Interconnects Layer 3 networks using protocols such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF) Protocol, and Enhanced Interior Gateway Routing Protocol (EIGRP) configured under SVI. For more informaton on routing protocol, see the IP Routing: Protocol-Independent Configuration Guide.
	Hot Standby Router Protocol (HSRP)	Supports redundancy and high availability with a secondary device connected to the LAN with SVI, using HSRP. For more informaton on HSRP, see the First Hop Redundancy Protocols Configuration Guide.
	DHCP	Cisco devices running Cisco software include Dynamic Host Configuration Protocol (DHCP) server and the relay agent software. The Cisco IOS DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the device to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the Domain Name System (DNS) server and the default device. For more informaton on HSRP, see the, IP Addressing: DHCP Configuration Guide
	Multicast (IPv4)	Provides multicast support for clients connected to the switch ports. For more informaton on HSRP, see the, IP Multicast: PIM Configuration Guide
	VRF	Associates a VRF instance with an SVI to map VLANs to different logical or physical VPN WAN connections. For more informaton on VRF protocol, see the IP Routing: Protocol-Independent Configuration Guide.
Security	ACL	Provides packet filtering to control network traffic and restrict the access of users and devices to the network For more informaton on ACL protocol, see the Security Configuration Guide: Access Control Lists.
	NAT	Provides NAT under SVI. For more information on NAT, see the IP Addressing: NAT Configuration Guide.

Techolongy	Feature	Use Case
Qos	Classification with standard and extended access list	Provides QoS classification with standard and extended access lists. For more informtion on QoS, see the Security Configuration Guide: Access Control Lists.
	Class-based marking	Provides QoS marking based on user-defined traffic class with DSCP and IP precedence values. For more information on QoS Marking, see the QoS: Classification Configuration Guide.
	Policing	Limits the input or output transmission rate on SVI and specifies traffic handling policies when the traffic either conforms to or exceeds the specified rate limits. For more informtion on Policing, see the QoS: Policing and Shaping Configuration Guide
Bridging	EVC under SVI	Supports a default encapsulation EFP under SVI, to have VLAN/BD integrated.
	EVC with MAC ACL under SVI	For more information on EVC, see *Layer 2 Configuration Guide, Cisco IOS XE Gibraltar 16.11*.

IGMP Snooping for IPv4

IGMP snooping allows switches to examine IGMP packets and make forwarding decisions based on their content. You can configure the switch to use IGMP snooping in subnets that receive IGMP queries from either IGMP or the IGMP snooping querier. IGMP snooping constrains IPv4 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv4 multicast traffic only to those ports that want to receive it.

Layer 2 switches can use IGMP snooping to constrain the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN switch to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the switch receives an IGMP report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients. For more information on this feature, see https://www.cisco.com/c/en/us/td/docs/routers/7600/ios/15S/configuration/guide/7600_15_0s_book/snooigmp.html.

IGMP Filtering and Throttling

In some environments, for example, metropolitan or multiple-dwelling unit (MDU) installations, you might want to control the set of multicast groups to which a user on a switch port can belong. You can control the distribution of multicast services, such as IP/TV, based on some type of subscription or service plan. You might also want to limit the number of multicast groups to which a user on a switch port can belong.

With the IGMP filtering feature, you can filter multicast joins on a per-port basis by configuring IP multicast profiles and associating them with individual switch ports. An IGMP profile can contain one or more multicast groups and specifies whether access to the group is permitted or denied. If an IGMP profile denying access to a multicast group is applied to a switch port, the IGMP join report requesting the stream of IP multicast traffic is dropped, and the port is not allowed to receive IP multicast traffic from that group. If the filtering action permits access to the multicast group, the IGMP report from the port is forwarded for normal processing. You can also set the maximum number of IGMP groups that a Layer 2 interface can join.

IGMP filtering controls only group-specific query and membership reports, including join and leave reports. It does not control general IGMP queries. IGMP filtering has no relationship with the function that directs the forwarding of IP multicast traffic. The filtering feature operates in the same manner whether CGMP or MVR is used to forward the multicast traffic.

IGMP filtering applies only to the dynamic learning of IP multicast group addresses, not static configuration.

With the IGMP throttling feature, you can set the maximum number of IGMP groups that a Layer 2 interface can join. If the maximum number of IGMP groups is set, the IGMP snooping forwarding table contains the maximum number of entries, and the interface receives an IGMP join report, you can configure an interface to drop the IGMP report or to replace the randomly selected multicast entry with the received IGMP report.

Default IGMP Filtering and Throttling Configuration

The following table displays the default IGMP filtering and throttling configuration for the device.

Table 13: Default IGMP Filtering Configuration

Feature	Default Setting
IGMP filters	None applied.
IGMP maximum number of IGMP groups	No maximum set. **Note** When the maximum number of groups is in the forwarding table, the default IGMP throttling action is to deny the IGMP report.
IGMP profiles	None defined.
IGMP profile action	Deny the range addresses.

Configuring IGMP Profiles

To configure an IGMP profile, use the **ip igmp profile** global configuration command with a profile number to create an IGMP profile and to enter IGMP profile configuration mode. From this mode, you can specify the parameters of the IGMP profile to be used for filtering IGMP join requests from a port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**ip igmp profile** *profile-number* **Example:** `Router(config)# ip igmp profile 3`	Enters IGMP profile configuration mode, and assigns a number to the profile you are configuring. The range is from 1 to 4294967295.
Step 3	**permit** | **deny** **Example:** `Router(config-igmp-profile)# permit`	(Optional) Sets the action to permit or deny access to the IP multicast address. If no action is configured, the default for the profile is to deny access.
Step 4	**range** *ip multicast address* **Example:** `Router(config-igmp-profile)# range 229.9.9.0`	Enter the IP multicast address or range of IP multicast addresses to which access is being controlled. If entering a range, enter the low IP multicast address, a space, and the high IP multicast address. You can use the **range** command multiple times to enter multiple addresses or ranges of addresses.
Step 5	**end** **Example:** `Router(config-igmp-profile)# end`	Returns to privileged EXEC mode.

Applying IGMP Profiles

To control access as defined in an IGMP profile, use the ip igmp filter interface configuration command to apply the profile to the appropriate interfaces. You can apply IGMP profiles to layer 2 access ports only; you cannot apply IGMP profiles to routed ports or SVIs. You cannot apply profiles to ports that belong to an EtherChannel port group. You can apply a profile to multiple interfaces, but each interface can only have one profile applied to it.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *interface-id* **Example:** `Router(config)# interface GigabitEthernet0/1/0`	Specifies the physical interface, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group.
Step 3	**ip igmp filter** *profile-number* **Example:** `Router(config-if)# ip igmp filter 123`	Applies the specified IGMP profile to the interface. The range is 1 to 4294967295.
Step 4	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Setting the Maximum Number of IGMP Groups

You can set the maximum number of IGMP groups that a Layer 2 interface can join by using the **ip igmp max-groups interface** configuration command. Use the **no** form of this command to set the maximum back to the default, which is no limit. This restriction can be applied to Layer 2 ports only; you cannot set a maximum number of IGMP groups on routed ports or SVIs. You also can use this command on a logical EtherChannel interface but cannot use it on ports that belong to an EtherChannel port group.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Router(config)# interface GigabitEthernet0/1/0`	Specifies the physical interface to be configured, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group.
Step 3	**ip igmp max-groups** *number* **Example:** `Router(config-if)# ip igmp max-groups 20`	Sets the maximum number of IGMP groups that the interface can join. The range is 0 to 4294967294. The default is to have no maximum set.
Step 4	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring the IGMP Throttling Action

After you set the maximum number of IGMP groups that a Layer 2 interface can join, you can configure an interface to remove a randomly selected multicast entry in the forwarding table and to add the next IGMP group to it by using the ip igmp max-groups action replace interface configuration command.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Router(config)# interface GigabitEthernet0/1/0`	Specifies the physical interface to be configured, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group.
Step 3	**ip igmp max-groups action** {**deny** | **replace**} **Example:** `Router(config-if)# ip igmp max-groups action replace`	When an interface receives an IGMP report and the maximum number of entries is in the forwarding table, specifies the action that the interface takes: • **deny** —Drop the report. • **replace** —Remove a randomly selected multicast entry in the forwarding table, and add the IGMP group in the report. To prevent the device from removing the forwarding-table entries, you can configure the IGMP throttling action before an interface adds entries to the forwarding table.
Step 4	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

MLD Snooping

In IP Version 4 (IPv4), Layer 2 switches can use Internet Group Management Protocol (IGMP) snooping to limit the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. In IPv6, MLD snooping performs a similar function. With MLD snooping, IPv6 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This list is constructed by snooping IPv6 multicast control packets.

MLD is a protocol used by IPv6 multicast routers to discover the presence of multicast listeners (nodes wishing to receive IPv6 multicast packets) on the links that are directly attached to the routers and to discover which multicast packets are of interest to neighboring nodes. MLD is derived from IGMP; MLD Version 1 (MLDv1) is equivalent to IGMPv2, and MLD Version 2 (MLDv2) is equivalent to IGMPv3. MLD is a subprotocol of Internet Control Message Protocol Version 6 (ICMPv6), and MLD messages are a subset of ICMPv6 messages, identified in IPv6 packets by a preceding Next Header value of 58.

MLD Snooping Configuration Guidelines

When configuring MLD snooping, consider these guidelines:

- You can configure MLD snooping characteristics at any time, but you must globally enable MLD snooping by using the **ipv6 mld snooping** global configuration command for the configuration to take effect.
- MLD snooping and IGMP snooping act independently of each other. You can enable both features at the same time on the switch.

Default MLD Snooping Configuration

Table 14: Default MLD Snooping Configuration

Feature	Default Setting
MLD snooping (Global)	Disabled.
MLD snooping (per VLAN)	Enabled. MLD snooping must be globally enabled for VLAN MLD snooping to take place.
IPv6 Multicast addresses	None configured.
IPv6 Multicast router ports	None configured.
MLD snooping Immediate Leave	Disabled.
MLD snooping robustness variable	Global: 2; Per VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Note	The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query count	Global: 2; Per VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Note	The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.

Feature	Default Setting
Last listener query interval	Global: 1000 (1 second); VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval.
Note	The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval.
TCN query solicit	Disabled.
TCN query count	2
MLD listener suppression	

Enabling or Disabling MLD Snooping on a VLAN

To enable MLD snooping on a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**ipv6 mld snooping** **Example:** `Router(config)# ipv6 mld snooping`	Enables MLD snooping on the switch.
Step 4	**ipv6 mld snooping vlan** *vlan-id* **Example:** `Device(config)# ipv6 mld snooping vlan 1`	Enables MLD snooping on the VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094. **Note** MLD snooping must be globally
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring UniDirectional Link Detection

UniDirectional Link Detection (UDLD) is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it disables the affected port and alerts you. Unidirectional links can cause a variety of problems, including spanning-tree topology loops.

Enabling UDLD Globally

Follow these steps to enable UDLD in the aggressive or normal mode and to set the configurable message timer on all fiber-optic ports on the device.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**udld** {**aggressive** | **enable** | **message time** *message-timer-interval*} **Example:** `Router(config)# udld enable message time 10`	Specifies the UDLD mode of operation: • **aggressive** —Enables UDLD in aggressive mode on all fiber-optic ports. • **enable** —Enables UDLD in normal mode on all fiber-optic ports on the . UDLD is disabled by default. An individual interface configuration overrides the setting of the **udld enable** global configuration command. • **message time** message-timer-interval —Configures the period of time between UDLD probe messages on ports that are in the advertisement phase and are detected to be bidirectional. The range is from 1 to 90 seconds; the default value is 15. **Note** This command affects fiber-command to enable UDLD Use the **no** form of this command, to disable UDLD.

	Command or Action	Purpose
Step 3	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling UDLD on an Interface

Follow these steps either to enable UDLD in the aggressive or normal mode or to disable UDLD on a port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Specifies the port to be enabled for UDLD, and enters interface configuration mode.
Step 3	**udld port** [**aggressive**] **Example:** `Router(config-if)# udld port aggressive`	UDLD is disabled by default. • **udld port** —Enables UDLD in normal mode on the specified port. • **udld port aggressive** —(Optional) Enables UDLD in aggressive mode on the specified port. **Note** Use the **no udld port** interface c
Step 4	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring the Switched Port Analyzer

This section describes how to configure a Switched Port Analyzer (SPAN) session.

- IR8340 can support 66 SPAN sessions in all ports. However, only eight of them can be used as source sessions which includes local SPAN sessions and remote SPAN source sessions. The remaining sessions can be used as remote SPAN destination sessions.

- The session ID range is from 1 to 66.

Note Tx, Rx, or both Tx and Rx monitoring is supported.

SPAN and RSPAN

You can analyze network traffic passing through ports or VLANs by using SPAN or RSPAN to send a copy of the traffic to another port on the device or on another device that has been connected to a network analyzer or other monitoring or security device. SPAN copies (or mirrors) traffic received or sent (or both) on source ports or source VLANs to a destination port for analysis. SPAN does not affect the switching of network traffic on the source ports or VLANs. You must dedicate the destination port for SPAN use. Destination ports do not receive or forward traffic by default. It can receive or forward traffic when ingress-forwarding is enabled on the destination ports.

Only traffic that enters or leaves source ports or traffic that enters or leaves source VLANs can be monitored by using SPAN; traffic routed to a source VLAN cannot be monitored. For example, if incoming traffic is being monitored, traffic that gets routed from another VLAN to the source VLAN cannot be monitored; however, traffic that is received on the source VLAN and routed to another VLAN can be monitored.

You can use the SPAN or RSPAN destination port to inject traffic from a network security device. For example, if you connect a Cisco Intrusion Detection System (IDS) sensor appliance to a destination port, the IDS device can send TCP reset packets to close down the TCP session of a suspected attacker.

Creating a Local SPAN Session

Follow these steps to create a SPAN session and specify the source (monitored) ports or VLANs and the destination (monitoring) ports.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Router(config)# no monitor session all`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all** —Removes all SPAN sessions. • **local** —Removes all local sessions.

	Command or Action	Purpose
		• **remote** —Removes all remote SPAN sessions.
Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [, | -] [**both** | **rx** | **tx**] **Example:** `Router(config)# monitor session 1 source interface gigabitethernet 0/1/0`	Specifies the SPAN session and the source port/Vlan (monitored port). • For session_number , the range is 1 to 66. • For interface-id , specify the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** port-channel-number). Valid port-channel numbers are 1 to 32. • For vlan-id , specify the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN). **Note** A single session can include m series of commands, but you ca in one session. • (Optional) [**,** | -] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **both** | **rx** | **tx** —Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic. • **both** —Monitors both received and sent traffic. • **rx** —Monitors received traffic. • **tx** —Monitors sent traffic. **Note** You can use the **monitor sessio** times to configure multiple sou

	Command or Action	Purpose
Step 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [, | -] [**encapsulation** {**replicate** | **dot1q**}]} **Example:** `Router(config)# monitor session 1 destination interface gigabitethernet 0/1/0 encapsulation replicate`	**Note** For local SPAN, you must us • For session_number , specify the session number entered in step 4. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged). (Optional) **encapsulation dot1q** specifies that the destination interface accepts the source interface incoming packets with IEEE 802.1Q encapsulation. **Note** You can use **monitor sessio** destination ports.
Step 6	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Router# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating a Local SPAN with Incoming Traffic Allowed on Destination

Follow these steps to create a SPAN session, to specify the source ports or VLANs and the destination ports, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Router(config)# no monitor session all`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all** —Removes all SPAN sessions. • **local** —Removes all local sessions. • **remote** —Removes all remote SPAN sessions.
Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [, | -] [**both** | **rx** | **tx**] **Example:** `Router(config)# monitor session 2 source gigabitethernet 0/1/0 rx`	Specifies the SPAN session and the source port (monitored port).
Step 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [, | -] [**encapsulation** {**replicate**] [**ingress** {**dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]} **Example:** `Router(config)# monitor session 2 destination interface gigabitethernet 0/1/0 encapsulation replicate ingress dot1q vlan 6`	Specifies the SPAN session, the destination port, the packet encapsulation, and the ingress VLAN and encapsulation. • For session_number , specify the session number entered in Step 4. • For interface-id , specify the destination port. The destination interface must be a physical port or port-channel; it cannot be an EtherChannel, and it cannot be a VLAN. • (Optional) [**,** | -] —Specifies a series or range of interfaces. Enter a space before and after the comma or hyphen. • (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the

	Command or Action	Purpose

		default is to send packets in native form (untagged).
		• (Optional) **encapsulation dot1q** specifies that the destination interface accepts the source interface incoming packets with IEEE 802.1Q encapsulation.
		• **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type:
		  • **dot1q vlan** vlan-id— Accepts incoming packets with IEEE 802.1Q encapsulation with the specified VLAN as the default VLAN.
		  • **untagged vlan** vlan-id or **vlan** vlan-id— Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

Step 6	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Router# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Specifying VLANs to Filter

Follow these steps to limit SPAN source traffic to specific VLANs.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Router(config)# no monitor session all`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all** —Removes all SPAN sessions. • **local** —Removes all local sessions. • **remote** —Removes all remote SPAN sessions.
Step 4	**monitor session** *session_number* **source** {**interface** *interface-id*] **Example:** `Router(config)# monitor session 2 source interface gigabitethernet 0/1/0 rx`	Specifies the characteristics of the source port (monitored port) and SPAN session. • For session_number , the range is 1 to 66. • For interface-id , specify the source port to monitor. The interface specified must already be configured as a trunk port.
Step 5	**monitor session** *session_number* **filter vlan** *vlan-id* [, | -] **Example:** `Router(config)# monitor session 2 filter vlan 1 - 5 , 9`	Limits the SPAN source traffic to specific VLANs. • For session_number , enter the session number specified in Step 4. • For vlan-id , the range is 1 to 4094. • (Optional) Use a comma (**,**) to specify a series of VLANs, or use a hyphen (**-**) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen.
Step 6	**monitor session** *session_number* **destination** {**interface** *interface-id* [, | -] [**encapsulation replicate** | **encapsulation dot1q**}]} **Example:** `Router(config)# monitor session 2 destination interface gigabitethernet 0/1/0`	Specifies the SPAN session and the destination port (monitoring port). • For session_number , specify the session number entered in Step 4. • For interface-id , specify the destination port. The destination interface must be a physical port or port-channel; it cannot be an EtherChannel, and it cannot be a VLAN. • (Optional) [**,** | -] Specifies a series or range of interfaces. Enter a space before and after

	Command or Action	Purpose
		the comma; enter a space before and after the hyphen. • (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged). • (Optional) **encapsulation dot1q** IEEE 802.1Q is a standard protocol for interconnecting multiple switches and routers and for defining VLAN topologies. Applies a VLAN ID to the subinterface.
Step 7	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Router# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Verifying the SPAN Session

Use the **show monitor session** command to verify the sources and destinations configured for the SPAN session.

```
Router#show monitor session 1

Session 1
---------
Session 1
---------
Type : Local Session
Source Ports :
Both : Gi0/1/0
Destination Ports : Gi0/1/1
```

Removing a SPAN Session

To remove sources or destinations from the SPAN session, use the **no monitor session** session command in global configuration mode as shown in the following example:

```
Router(config)#no monitor session 1
```

Configuring a VLAN as an RSPAN VLAN

Follow these steps to create a new VLAN, then configure it to be the RSPAN VLAN for the RSPAN session.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**vlan** *vlan-id* **Example:** `Router(config)# vlan 100`	Enters a VLAN ID to create a VLAN, or enters the VLAN ID of an existing VLAN, and enters VLAN configuration mode. The range is 2 to 1001 and 1006 to 4094. The RSPAN VLAN cannot be VLAN 1 (the default VLAN) or VLAN IDs 1002 through 1005 (reserved for Token Ring and FDDI VLANs).
Step 4	**remote-span** **Example:** `Router(config)# remote-span`	Configures the VLAN as an RSPAN VLAN.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Router# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

You must create the RSPAN VLAN in all devices that will participate in RSPAN. If the RSPAN VLAN-ID is in the normal range (lower than 1005) and VTP is enabled in the network, you can create the RSPAN VLAN in one device, and VTP propagates it to the other devices in the VTP domain. For extended-range VLANs (greater than 1005), you must configure RSPAN VLAN on both source and destination devices and any intermediate devices.

Use VTP pruning to get an efficient flow of RSPAN traffic, or manually delete the RSPAN VLAN from all trunks that do not need to carry the RSPAN traffic.

To remove the remote SPAN characteristic from a VLAN and convert it back to a normal VLAN, use the **no remote-span** VLAN configuration command.

To remove a source port or VLAN from the SPAN session, use the **no monitor session** session_number **source** {**interface** interface-id | **vlan** vlan-id} global configuration command. To remove the RSPAN VLAN from the session, use the **no monitor session** session_number {**Source|destination**} **remote vlan** vlan-id .

Creating an RSPAN Source Session

Follow these steps to create and start an RSPAN source session and to specify the monitored source and the destination RSPAN VLAN.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Router(config)# no monitor session all`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all** —Removes all SPAN sessions. • **local** —Removes all local sessions. • **remote** —Removes all remote SPAN sessions.
Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [, | -] [**both** | **rx** | **tx**] **Example:**	Specifies the RSPAN session and the source port (monitored port). • For session_number , the range is 1 to 66. • Enter a source port or source VLAN for the RSPAN session:

	Command or Action	Purpose
	`Router(config)# monitor session 1 source interface gigabitethernet 0/1/0 tx`	• For interface-id , specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** port-channel-number). Valid port-channel numbers are 1 to 32. • For vlan-id , specifies the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN). A single session can include multiple sources (ports or VLANs), defined in a series of commands, but you cannot combine source ports and source VLANs in one session. • (Optional) [**,** | -] —Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **both** | **rx** | **tx** —Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic. • **both** —Monitors both received and sent traffic. • **rx** —Monitors received traffic. • **tx** —Monitors sent traffic.
Step 5	**monitor session** *session_number* **destination remote vlan** *vlan_id* **Example:** `Router(config)# monitor session 1 destination remote vlan 100`	Specifies the RSPAN session, the destination RSPAN VLAN, and the destination-port group. • For session_number , enter the number defined in Step 4. • For vlan-id , specify the RSPAN VLAN in source session, which will transport mirrored traffic to destination session.
Step 6	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show running-config** **Example:** `Router# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Specifying VLANs to Filter on RSPAN Source Session

Follow these steps to configure the RSPAN source session to limit RSPAN source traffic to specific VLANs.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Router(config)# no monitor session 2`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all** —Removes all SPAN sessions. • **local** —Removes all local sessions. • **remote** —Removes all remote SPAN sessions.
Step 4	**monitor session** *session_number* **source interface** *interface-id* **Example:** `Router(config)# monitor session 2 source interface gigabitethernet 0/1/0 rx`	Specifies the characteristics of the source port (monitored port) and SPAN session. • For session_number , the range is 1 to 66. • For interface-id , specify the source port to monitor. The interface specified must already be configured as a trunk port.
Step 5	**monitor session** *session_number* **filter vlan** *vlan-id* [, | -]	Limits the SPAN source traffic to specific VLANs.

	Command or Action	Purpose
	Example: `Router(config)# monitor session 2 filter vlan 1 - 5 , 9`	• For session_number , enter the session number specified in step 4. • For vlan-id , the range is 1 to 4094. • (Optional) **,** | **-** Use a comma (**,**) to specify a series of VLANs or use a hyphen (**-**) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen.
Step 6	**monitor session** *session_number* **destination remote vlan** *vlan-id* **Example:** `Router(config)# monitor session 2 destination remote vlan 902`	Specifies the RSPAN session and the destination remote VLAN (RSPAN VLAN). • For session_number , enter the session number specified in Step 4. • For vlan-id , specify the RSPAN VLAN to carry the monitored traffic to the destination port.
Step 7	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Router# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating an RSPAN Destination Session and Configuring Incoming Traffic

Follow these steps to create an RSPAN destination session, to specify the source RSPAN VLAN and the destination port, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Router(config)# no monitor session 2`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all** —Removes all SPAN sessions. • **local** —Removes all local sessions. • **remote** —Removes all remote SPAN sessions.
Step 4	**monitor session** *session_number* **source remote vlan** *vlan-id* **Example:** `Router(config)# monitor session 2 source remote vlan 901`	Specifies the RSPAN session and the source RSPAN VLAN. • For session_number , the range is 1 to 66. • For vlan-id , specify the RSPAN VLAN in destination session, which will receive mirrored traffic from the source session.
Step 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [, | -] [**ingress** {**dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]} **Example:** `Router(config)# monitor session 2 destination interface gigabitethernet 0/1/0 ingress vlan 6`	Specifies the SPAN session, the destination port, the packet encapsulation, and the incoming VLAN and encapsulation. • For session_number , enter the number defined in Step 5. In an RSPAN destination session, you must use the same session number for the source RSPAN VLAN and the destination port. • For interface-id , specify the destination interface. The destination interface must be a physical interface. • Though visible in the command-line help string, **encapsulation replicate** is not supported for RSPAN. The original VLAN ID is overwritten by the RSPAN VLAN ID, and all packets appear on the destination port as untagged. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.

	Command or Action	Purpose
		• Enter **ingress** with additional keywords to enable forwarding of incoming traffic on the destination port and to specify the encapsulation type: • **dot1q vlan** vlan-id— Forwards incoming packets with IEEE 802.1Q encapsulation with the specified VLAN as the default VLAN. • **untagged vlan** vlan-id or **vlan** vlan-id— Forwards incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.
Step 6	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Router# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

EtherChannel Overview

EtherChannel provides fault-tolerant high-speed links between switches, routers, and servers. You can use the EtherChannel to increase the bandwidth between the wiring closets and the data center, and you can deploy it anywhere in the network where bottlenecks are likely to occur. EtherChannel provides automatic recovery for the loss of a link by redistributing the load across the remaining links. If a link fails, EtherChannel redirects traffic from the failed link to the remaining links in the channel without intervention.

An EtherChannel consists of individual Ethernet links bundled into a single logical link

The EtherChannel provides full-duplex bandwidth up to 4 Gb/s (Gigabit EtherChannel) between your switch and another switch or host.

Each EtherChannel can consist of up to four compatibly configured Ethernet ports.

Channel Groups and Port-Channel Interfaces

An EtherChannel comprises a channel group and a port-channel interface. The channel group binds physical ports to the port-channel interface. Configuration changes applied to the port-channel interface apply to all the physical ports bound together in the channel group. The channel-group command binds the physical port and the port-channel interface together. Each EtherChannel has a port-channel logical interface numbered from 1 to 32. This port-channel interface number corresponds to the one specified with the channel-group interface configuration command.

Port Aggregation Protocol

The Port Aggregation Protocol (PAgP) is a Cisco-proprietary protocol that can be run only on Cisco devices and on those devices licensed by vendors to support PAgP. PAgP facilitates the automatic creation of EtherChannels by exchanging PAgP packets between Ethernet ports.

By using PAgP, the device learns the identity of partners capable of supporting PAgP and the capabilities of each port. It then dynamically groups similarly configured ports (on a single device in the stack) into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, PAgP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, PAgP adds the group to the spanning tree as a single device port.

Link Aggregation Control Protocol

The LACP is defined in IEEE 802.3ad and enables Cisco devices to manage Ethernet channels between devices that conform to the IEEE 802.3ad protocol. LACP facilitates the automatic creation of EtherChannels by exchanging LACP packets between Ethernet ports.

By using LACP, the switch learns the identity of partners capable of supporting LACP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, LACP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, LACP adds the group to the spanning tree as a single device port.

Configuring Layer 2 EtherChannels

Configure Layer 2 EtherChannels by assigning ports to a channel group with the **channel-group** command in interface configuration mode. This command automatically creates the port-channel logical interface.

Use `show etherchannel summary` to view Layer 3 (L3) EtherChannel information. Use `show etherchannel swport summary` to view Layer 2 (L2) EtherChannel information.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** [*interface-id*] **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Specifies a physical port, and enters interface configuration mode. Valid interfaces are physical ports. For a PAgP EtherChannel, you can configure up to four ports of the same type and speed for the same group. For a LACP EtherChannel, you can configure up to 8 Ethernet ports of the same type. Up to eight ports can be active, and up to eight ports can be in standby mode.
Step 4	**switchport mode** {**access** | **trunk**} **Example:** `Router(config-if)# switchport mode access`	Assigns all ports as static-access ports in the same VLAN, or configure them as trunks. If you configure the port as a static-access port, assign it to only one VLAN. The range is 1 to 4094.
Step 5	**switchport access vlan** *vlan-id* **Example:** `Router(config-if)# switchport access vlan 22`	(Optional) If you configure the port as a static-access port, assign it to only one VLAN. The range is 1 to 4094.
Step 6	**channel-group channel-group-number mode {auto [non-silent] | desirable [non-silent] | on } | { active | passive}** **Example:** `Router(config-if)# channel-group 5 mode auto`	Assigns the port to a channel group, and specifies the PAgP or the LACP mode. For **mode** , select one of these keywords: • **auto —** Enables PAgP only if a PAgP device is detected. It places the port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not start PAgP packet negotiation. • **desirable —** Unconditionally enables PAgP. It places the port into an active negotiating state, in which the port starts negotiations with other ports by sending PAgP packets. • **on —** Forces the port to channel without PAgP or LACP. In the **on** mode, an EtherChannel exists only when a port group in the **on** mode is connected to another port group in the **on** mode.

	Command or Action	Purpose
		• **non-silent —** (Optional) If your device is connected to a partner that is PAgP-capable, configures the device port for nonsilent operation when the port is in the **auto** or **desirable** mode. If you do not specify **non-silent** , silent is assumed. The silent setting is for connections to file servers or packet analyzers. This setting allows PAgP to operate, to attach the port to a channel group, and to use the port for transmission. • **active** —Enables LACP only if a LACP device is detected. It places the port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets. • **passive —** Enables LACP on the port and places it into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation.
Step 7	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring EtherChannel Load-Balancing

You can configure EtherChannel load-balancing to use one of several different forwarding methods.

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**port-channel swport load-balance { dst-ip | dst-mac | dst-mixed-ip-port | dst-port | extended [dst-ip | dst-mac | dst-port | ipv6-label | l3-proto | src-ip | src-mac | src-port] | src-dst-ip | src-dst-mac**	Configures an EtherChannel load-balancing method. Select one of these load-distribution methods: • **dst-ip** —Specifies destination-host IP address.

	Command or Action	Purpose
	src-dst-mixed-ip-port src-dst-portsrc-ip | src-mac | src-mixed-ip-port | src-port } **Example:** `Router(config)# port-channel swport load-balance src-mac`	• **dst-mac** —Specifies the destination-host MAC address of the incoming packet. • **dst-mixed-ip-port** —Specifies the host IP address and TCP/UDP port. • **dst-port** —Specifies the destination TCP/UDP port. • **extended** —Specifies extended load balance methods--combinations of source and destination methods beyond those available with the standard command. • **ipv6-label** —Specifies the IPv6 flow label. • **l3-proto** —Specifies the Layer 3 protocol. • **src-dst-ip** —Specifies the source and destination host IP address. • **src-dst-mac** —Specifies the source and destination host MAC address. • **src-dst-mixed-ip-port** —Specifies the source and destination host IP address and TCP/UDP port. • **src-dst-port** —Specifies the source and destination TCP/UDP port. • **src-ip** —Specifies the source host IP address. • **src-mac** —Specifies the source MAC address of the incoming packet. • **src-mixed-ip-port** —Specifies the source host IP address and TCP/UDP port. • **src-port** —Specifies the source TCP/UDP port.
Step 3	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring the PAgP Learn Method and Priority

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** [*interface-id*] **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Specifies the port for transmission, and enters interface configuration mode.
Step 4	**pagp learn-method physical-port** **Example:** `Router(config-if)# pagp learn-method physical port`	Selects the PAgP learning method. By default, **aggregation-port learning** is selected, which means the device sends packets to the source by using any of the ports in the EtherChannel. With aggregate-port learning, it is not important on which physical port the packet arrives. Selects **physical-port** to connect with another device that **is** a physical learner. Make sure to configure the **port-channel load-balance** global configuration command to **src-mac** . The learning method must be configured the same at both ends of the link.
Step 5	**pagp port-priority priority** **Example:** `Router(config-if)# pagp port-priority 200`	Assigns a priority so that the selected port is chosen for packet transmission. For *priority*, the range is 0 to 255. The default is 128. The higher the priority, the more likely that the port will be used for PAgP transmission.
Step 6	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring the LACP Port Channel Min-Links Feature

You can specify the minimum number of active ports that must be in the link-up state and bundled in an EtherChannel for the port channel interface to transition to the link-up state. Using EtherChannel min-links,

you can prevent low-bandwidth LACP EtherChannels from becoming active. Port channel min-links also cause LACP EtherChannels to become inactive if they have too few active member ports to supply the required minimum bandwidth.

To configure the minimum number of links that are required for a port channel. Perform the following tasks.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface port-channel** [*channel-number*] **Example:** `Router(config)# interface port-channel 2`	Enters interface configuration mode for a port-channel. For *channel-number*, the range is 1 to 63.
Step 4	**port-channel min-links** *min-links-number* **Example:** `Router(config-if)# port-channel min-links 3`	Specifies the minimum number of member ports that must be in the link-up state and bundled in the EtherChannel for the port channel interface to transition to the link-up state. For *min-links-number* , the range is 2 to 8.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring LACP Fast Rate Timer

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the **lacp rate** command to set the rate at which LACP control packets are received by an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Router(config)# interface gigabitEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**lacp rate { normal | fast}** **Example:** `Router(config-if)# lacp rate fast`	Configures the rate at which LACP control packets are received by an LACP-supported interface. To reset the timeout rate to its default, use the **no lacp rate** command.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Step 6	**show lacp internal**
	Example:
	`Router# show lacp internal`
	`Router# show lacp counters`
		Verifies your configuration.

Modular Quality of Service Command-Line Interface

The MQC (Modular Quality of Service (QoS) Command-Line Interface (CLI)) enables you to set packet classification and marking based on a QoS group value. ith the device, QoS features are enabled through the Modular QoS command-line interface (MQC). The MQC is a command-line interface (CLI) structure that allows you to create traffic policies and attach these policies to interfaces. A traffic policy contains a traffic class and one or more QoS features. A traffic class is used to classify traffic, while the QoS features in the traffic policy determine how to treat the classified traffic. One of the main goals of MQC is to provide a platform-independent interface for configuring QoS across Cisco platforms. For more infomraton on the Modular Quality of Service, see the Quality of Service Configuration Guide, Cisco IOS XE Fuji 16.9.x.

Note **QoS limitations on IR8340 platforms**

1. Quality of Service (QoS) is supported for traffic moving between switch ports.
2. For inter-vlan routing between switch ports, QoS limitations are:
   - It is supported only when a policy is applied on the Switch Virtual Interface (SVI).
   - Only marking and policing actions based on Differentiated Services Code Point (DSCP) match are supported on SVI.
   - Queuing and shaping actions are not supported on SVI.
3. For traffic moving from LAN to WAN port, QoS limitations are:
   - It is supported only when a policy is applied on SVI.
   - Only marking and policing actions based on Differentiated Services Code Point (DSCP) match are supported on SVI.
   - Queuing and shaping actions are not supported on SVI.

Creating a Traffic Class

To create a traffic class containing match criteria, use the **class-map** command to specify the traffic class name, and then use the following **match** commands in class-map configuration mode, as needed.

Before you begin

All match commands specified in this configuration task are considered optional, but you must configure at least one match criterion for a class.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.

Step 2	**class-map** *class-map name* {**match-any**}
	Example:
	`Router(config)# class-map type ngsw-qos test_1000`
	`Router(config-cmap)#`
		Enters class map configuration mode.
		• Creates a class map to be used for matching packets to the class whose name you specify.
		• **match-any:** Any one of the match criteria must be met for traffic entering the traffic class to be classified as part of it.

	Command or Action	Purpose
Step 3	**match access-group** { *index number* | *name*} **Example:** `Router(config-cmap)# match access-group 100` `Router(config-cmap)#`	The following parameters are available for this command: • access-group • cos • dscp • group-object • ip • mpls • precedence • protocol • qos-group • vlan • wlan (Optional) For this example, enter the access-group ID: • Access list index (value from 1 to 2799) • Named access list
Step 4	**match cos** *cos value* **Example:** `Router(config-cmap)# match cos 2 3 4 5`	(Optional) Matches IEEE 802.1Q or ISL class of service (user) priority values. • Enters up to 4 CoS values separated by spaces (0 to 7).
Step 5	**match dscp** *dscp value* **Example:** `Router(config-cmap)# match dscp af11 af12`	(Optional) Matches the DSCP values in IPv4 and IPv6 packets.
Step 6	**match ip** { **dscp** *dscp value* | **precedence** *precedence value* } **Example:** `Router(config-cmap)# match ip dscp af11 af12`	(Optional) Matches IP values including the following: • **dscp**—Matches IP DSCP (DiffServ codepoints). • **precedence**—Matches IP precedence (0 to 7).
Step 7	**match qos-group** *qos group value* **Example:** `Router(config-cmap)# match qos-group 10`	(Optional) Matches QoS group value (from 0 to 31).

	Command or Action	Purpose
Step 8	**match vlan** *vlan value* **Example:** `Router(config-cmap)# match vlan 210`	(Optional) Matches a VLAN ID (from 1 to 4095).
Step 9	**end** **Example:** `Router(config)# end`	Saves the configuration changes.

What to do next

Configure the policy map.

Creating a Traffic Policy

To create a traffic policy, use the **policy-map** global configuration command to specify the traffic policy name.

The traffic class is associated with the traffic policy when the **class** command is used. The **class** command must be entered after you enter the policy map configuration mode. After entering the **class** command, the device is automatically in policy map class configuration mode, which is where the QoS policies for the traffic policy are defined.

The following policy map class-actions are supported:

- bandwidth—Bandwidth configuration options.
- exit—Exits from the QoS class action configuration mode.
- no—Negates or sets default values for the command.
- police—Policer configuration options.
- priority—Strict scheduling priority configuration options for this class.
- queue-buffers—Queue buffer configuration options.
- queue-limit—Queue maximum threshold for Weighted Tail Drop (WTD) configuration options.
- service-policy—Configures the QoS service policy.
- set—Sets QoS values using the following options:
  - CoS values
  - DSCP values
  - Precedence values
  - QoS group values
- shape—Traffic-shaping configuration options.

Before you begin

You should have first created a class map.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**policy-map type** *policy-map name* **Example:** `Router(config)# policy-map type ngsw-qos test_1000`	Enters policy map configuration mode. Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 3	**class** { *class-name* | **class-default**} **Example:** `Router(config-pmap)# class test_1000`	Specifies the name of the class whose policy you want to create or change. You can also create a system default class for unclassified packets.
Step 4	**bandwidth** { **kb/s** *kb/s value* | **percent** *percentage* | **remaining** {*percent* | *ratio*}} **Example:** `Router(config-pmap-c)# bandwidth 50`	(Optional) Sets the bandwidth using one of the following: • **kb/s**—Kilobits per second, enter a value between 20000 and 10000000 for Kb/s. • **percent**—Enter the percentage of the total bandwidth to be used for this policy map. • **remaining**—Enter the percentage ratio of the remaining bandwidth.
Step 5	**exit** **Example:** `Router(config-pmap-c)# exit`	(Optional) Exits from QoS class action configuration mode.
Step 6	**no** **Example:** `Router(config-pmap-c)# no`	(Optional) Negates the command.
Step 7	**police** { *target_bit_rate* | **cir** | **rate**} **Example:** `Router(config-pmap-c)# police 100000`	(Optional) Configures the policer: • *target_bit_rate*—Enter the bit rate per second, enter a value between 8000 and 10000000000. • **cir**—Committed Information Rate

	Command or Action	Purpose
		• **rate**—Specify police rate, PCR for hierarchical policies or SCR for single-level ATM 4.0 policer policies.
Step 8	**priority level** *level value* **Example:** `Router(config-pmap-c)# priority level 1`	(Optional) Sets the strict scheduling priority for this class. Command options include: • **level**—Establishes a multi-level priority queue. Enter a value (1 or 2).
Step 9	**queue-buffers ratio** *ratio limit* **Example:** `Router(config-pmap-c)# queue-buffers ratio 10`	(Optional) Configures the queue buffer for the class. Enter the queue buffers ratio limit (0 to 100).
Step 10	**queue-limit { packets | cos | dscp | percent}** **Example:** `Router(config-pmap-c)# queue-limit cos 7 percent 50`	(Optional) Specifies the queue maximum threshold for the tail drop: • *packets*—Packets by default, enter a value between 1 to 2000000. • **cos**—Enter the parameters for each COS value. • **dscp**—Enter the parameters for each DSCP value. • **percent**—Enter the percentage for the threshold.
Step 11	**service-policy** *policy-map name* **Example:** `Router(config-pmap-c)# service-policy test_2000`	(Optional) Configures the QoS service policy.
Step 12	**set { cos | dscp | ip | precedence | qos-group | wlan}** **Example:** `Router(config-pmap-c)# set cos 7`	(Optional) Sets the QoS values. Possible QoS configuration values include: • **cos**—Sets the IEEE 802.1Q/ISL class of service/user priority. • **dscp**—Sets DSCP in IP(v4) and IPv6 packets. • **ip**—Sets IP specific values. • **precedence**—Sets precedence in IP(v4) and IPv6 packet. • **qos-group**—Sets the QoS Group.

	Command or Action	Purpose
Step 13	**shape average** { *target _bit_rate* | **percent**} **Example:** `Router(config-pmap-c)# shape average percent 50`	(Optional) Sets the traffic shaping. Command parameters include: • *target_bit_rate*—Target bit rate. • **percent**—Percentage of interface bandwidth for Committed Information Rate.
Step 14	**end** **Example:** `Router(config)# end`	Saves the configuration changes.

What to do next

Configure the interface.

Configuring Class-Based Packet Marking

This is an important procedure that explains how to configure the following class-based packet marking features on your device:

- CoS value
- DSCP value
- IP value
- Precedence value
- QoS group value
- WLAN value

Before you begin

You should have created a class map and a policy map before beginning this procedure.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**policy-map type** *policy-map name* **Example:**	Enters policy map configuration mode.

	Command or Action	Purpose
	`Router(config)# policy-map type ngsw-qos policy1` `Device(config-pmap)#`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.

Step 3	**class** *class name*
	Example:
	`Router(config)# class class1`
	`Device(config-pmap)#`
		Enters policy class map configuration mode. Specifies the name of the class whose policy you want to create or change.
		Command options for policy class map configuration mode include the following:
		• **bandwidth**—Bandwidth configuration options.
		• **exit**—Exits from the QoS class action configuration mode.
		• **no**—Negates or sets default values for the command.
		• **police**—Policer configuration options.
		• **priority**—Strict scheduling priority configuration options for this class.
		• **queue-buffers**—Queue buffer configuration options.
		• **queue-limit**—Queue maximum threshold for Weighted Tail Drop (WTD) configuration options.
		• **service-policy**—Configures the QoS service policy.
		• **set**—Sets QoS values using the following options:
		  • CoS values
		  • DSCP values
		  • Precedence values
		  • QoS group values
		  • WLAN values
		• **shape**—Traffic-shaping configuration options.

	Command or Action	Purpose
		Note This procedure describes the (**bandwidth**) are described commands, only one **set** co
Step 4	**set cos** {*cos value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name* | **wlan user-priority table** *table-map name*} **Example:** `Router(config-pmap)# set cos 5`	(Optional) Sets the specific IEEE 802.1Q Layer 2 CoS value of an outgoing packet. Values are from 0 to7. You can also set the following values using the **set cos** command: • **cos table**—Sets the CoS value based on a table map. • **dscp table**—Sets the code point value based on a table map. • **precedence table**—Sets the code point value based on a table map. • **qos-group table**—Sets the CoS value from QoS group based on a table map. • **wlan user-priority table**—Sets the CoS value from the WLAN user priority based on a table map.
Step 5	**set dhcp** {*dhcp value* | **default** | **dscp table** *table-map name* | **ef** | **precedence table** *table-map name* | **qos-group table** *table-map name* | **wlan user-priority table** *table-map name*} **Example:** `Router(config-pmap)# set dscp af11`	(Optional) Sets the DSCP value. In addition to setting specific DSCP values, you can also set the following using the **set dscp** command: • **default**—Matches packets with default DSCP value (000000). • **dscp table**—Sets the packet DSCP value from DSCP based on a table map. • **ef**—Matches packets with EF DSCP value (101110). • **precedence table**—Sets the packet DSCP value from precedence based on a table map. • **qos-group table**—Sets the packet DSCP value from a QoS group based upon a table map.

	Command or Action	Purpose
		• **wlan user-priority table**—Sets the packet DSCP value based upon a WLAN user-priority based upon a table map.
Step 6	**set ip {dscp | precedence}** **Example:** `Router(config-pmap)# set ip dscp c3`	(Optional) Sets IP specific values. These values are either IP DSCP or IP precedence values. You can set the following values using the **set ip dscp** command: • *dscp value*—Sets a specific DSCP value. • **default**—Matches packets with default DSCP value (000000). • **dscp table**—Sets the packet DSCP value from DSCP based on a table map. • **ef**—Matches packets with EF DSCP value (101110). • **precedence table**—Sets the packet DSCP value from precedence based on a table map. • **qos-group table**—Sets the packet DSCP value from a QoS group based upon a table map. • **wlan user-priority table**—Sets the packet DSCP value based upon a WLAN user-priority based upon a table map. You can set the following values using the **set ip precedence** command: • *precedence value*—Sets the precedence value (from 0 to 7) . • **cos table**—Sets the packet precedence value from Layer 2 CoS based on a table map. • **dscp table**—Sets the packet precedence from DSCP value based on a table map. • **precedence table**—Sets the precedence value from precedence based on a table map • **qos-group table**—Sets the precedence value from a QoS group based upon a table map.

	Command or Action	Purpose
Step 7	**set precedence** {*precedence value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name*} **Example:** `Router(config-pmap)# set precedence 5`	(Optional) Sets precedence values in IPv4 and IPv6 packets. You can set the following values using the **set precedence** command: • *precedence value*—Sets the precedence value (from 0 to 7) . • **cos table**—Sets the packet precedence value from Layer 2 CoS on a table map. • **dscp table**—Sets the packet precedence from DSCP value on a table map. • **precedence table**—Sets the precedence value from precedence based on a table map. • **qos-group table**—Sets the precedence value from a QoS group based upon a table map.
Step 8	**set qos-group** {*qos-group value* | **dscp table** *table-map name* | **precedence table** *table-map name*} **Example:** `Router(config-pmap)# set qos-group 10`	(Optional) Sets QoS group values. You can set the following values using this command: • *qos-group value*—A number from 1 to 31. • **dscp table**—Sets the code point value from DSCP based on a table map. • **precedence table**—Sets the code point value from precedence based on a table map.
Step 9	**set wlan user-priority table** {*wlan user-priority table value* | **cos table** *table-map name* | **dscp table** *table-map name* | **qos-group table** *table-map name* | **wlan table** *table-map name*} **Example:** `Router(config-pmap)# set wlan user-priority 1`	(Optional) Sets the WLAN user priority value. You can set the following values using this command: • *wlan user-priority value*—A value between 0 to 7. • **cos table**—Sets the WLAN user priority value from CoS based on a table map. • **dscp table**—Sets the WLAN user priority value from DSCP based on a table map. • **qos-group table**—Sets the WLAN user priority value from QoS group based on a table map.

	Command or Action	Purpose
		• **wlan table**—Sets the WLAN user priority value from the WLAN user priority based on a table map.
Step 10	**end** **Example:** `Router(config)# end`	Saves the configuration changes.
Step 11	**show policy-map** **Example:** `Router(config)# show policy-map`	(Optional) Displays policy configuration information for all classes configured for all service policies.

What to do next

Attach the traffic policy to an interface using the **service-policy** command.

Attaching a Traffic Policy to an Interface

After the traffic class and traffic policy are created, you must use the **service-policy** interface configuration command to attach a traffic policy to an interface, and to specify the direction in which the policy should be applied (either on packets coming into the interface or packets leaving the interface).

Before you begin

A traffic class and traffic policy must be created before attaching a traffic policy to an interface.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**interface** *type*	
Step 3	**service-policy** {**input** *policy-map* | **output** *policy-map*} **Example:** `Router(config-if)# service-policy output policy_map_01`	Attaches a policy map to an input or output interface. This policy map is then used as the service policy for that interface. In this example, the traffic policy evaluates all traffic leaving that interface.
Step 4	**end** **Example:** `Router(config)# end`	Saves the configuration changes.

	Command or Action	Purpose
Step 5	**show policy map** **Example:** `Router(config)# show policy map`	(Optional) Displays statistics for the policy on the specified interface.

Example

What to do next

Proceed to attach any other traffic policy to an interface, and to specify the direction in which the policy should be applied.

Classifying, Policing, and Marking Traffic on Physical Ports by Using Policy Maps

You can configure a nonhierarchical policy map on a physical port that specifies which traffic class to act on. Actions supported are remarking and policing.

Before you begin

You should have already decided upon the classification, policing, and marking of your network traffic by policy maps prior to beginning this procedure.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.

Step 2	**class-map** { *class-map name* | **match-any**}
	Example:
	`Device(config)# class-map ipclass1`
	`Device(config-cmap)# exit`
		Enters class map configuration mode.
		• Creates a class map to be used for matching packets to the class whose name you specify.
		• If you specify **match-any**, one of the match criteria must be met for traffic entering the traffic class to be classified as part of the traffic class. This is the default.

Step 3	**match access-group** { *access list index* | *access list name* } **Example:**	The following parameters are available for this command: • access-group

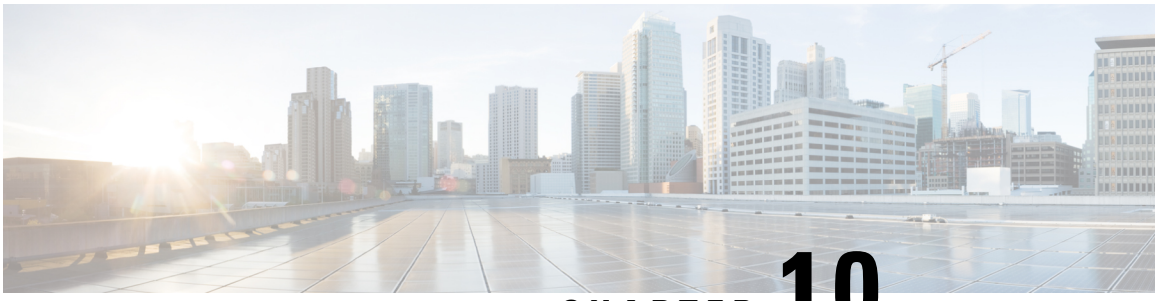
	Command or Action	Purpose
	Device(config-cmap)# **match access-group 1000** Device(config-cmap)# **exit**	• cos • dscp • group-object • ip • mpls • precedence • protocol • qos-group • vlan • wlan (Optional) For this example, enter the access-group ID: • Access list index (value from 1 to 2799) • Named access list
Step 4	**policy-map** *policy-map name* **Example:** Router(config)# **policy-map type ngsw-qos flowit** Device(config-pmap)#	Creates a policy map by entering the policy map name, and enters policy-map configuration mode. By default, no policy maps are defined.
Step 5	**class** {*class-map-name* | **class-default**} **Example:** Device(config-pmap)# **class ipclass1**	Defines a traffic classification, and enter policy-map class configuration mode. By default, no policy map class-maps are defined. If a traffic class has already been defined by using the **class-map** global configuration command, specify its name for class-map-name in this command. A **class-default** traffic class is predefined and can be added to any policy. It is always placed at the end of a policy map. With an implied **match any** included in the **class-default** class, all packets that have not already matched the other traffic classes will match **class-default**.
Step 6	**set { cos | dscp | ip | precedence | qos-group | wlan user-priority}** **Example:** Device(config-pmap-c)# **set dscp 45**	(Optional) Sets the QoS values. Possible QoS configuration values include: • **cos**—Sets the IEEE 802.1Q/ISL class of service/user priority.

	Command or Action	Purpose
		• **dscp**—Sets DSCP in IP(v4) and IPv6 packets. • **ip**—Sets IP specific values. • **precedence**—Sets precedence in IP(v4) and IPv6 packet. • **qos-group**—Sets QoS group. • **wlan user-priority**—Sets WLAN user priority. In this example, the **set dscp** command classifies the IP traffic by setting a new DSCP value in the packet.
Step 7	**police** { *target_bit_rate* | **cir** | **rate** } **Example:** `Device(config-pmap-c)# police 100000 conform-action transmit exceed-action drop`	(Optional) Configures the policer: • *target_bit_rate*—Specifies the bit rate per second, enter a value between 8000 and 10000000000. • **cir**—Committed Information Rate. • **rate**—Specifies the police rate PCR for hierarchical policies. In this example, the **police** command adds a policer to the class where any traffic beyond the 100000 set target bit rate is dropped.
Step 8	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy map configuration mode.
Step 9	**exit** **Example:** `Router(config-pmap)# exit`	Returns to global configuration mode.
Step 10	**interface** [*interface-id*] **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Specifies the port to attach to the policy map, and enters interface configuration mode. Valid interfaces include physical ports.
Step 11	**service-policy input** [*policy-map-name*] **Example:** `Device(config-if)# service-policy input flowit`	Specifies the policy-map name, and applies it to an ingress port. Only one policy map per ingress port is supported.

	Command or Action	Purpose
Step 12	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 13	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] **Example:** `Router(config)# show policy-map`	(Optional) Verifies your entries.
Step 14	**copy running-config startup-config** **Example:** `Router(config)# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

If applicable to your QoS configuration, configure classification, policing, and marking of traffic on SVIs by using policy maps.



CHAPTER 10

Configuring Switchport Blocking

- About Switchport Blocking, on page 149
- Configuring Switchport Blocking, on page 149

About Switchport Blocking

By default, the router floods packets with unknown destination MAC addresses to all ports. To prevent the forwarding of such traffic, you can configure a port to block unknown multicast or unicast packets.

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. The interface can be a physical interface or an EtherChannel group. When you block multicast or unicast traffic for a port channel, it is blocked on all ports in the port channel group.

Configuring Switchport Blocking

Follow these steps to configure switchport blocking. Blocking of unicast or multicast traffic is not automatically enabled on a switch port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**interface** {*interface-id* | **port-channel** *number*} **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Enters interface configuration mode.
Step 3	**switchport mode access** **Example:**	Configures the interface as an access port.

	Command or Action	Purpose
	`Router(config-if)# switchport mode access`	
Step 4	**switchport access vlan** *vlan-id* **Example:** `Router(config-if)# switchport access vlan 20`	Specifies the VLAN for which this access port will carry traffic.

Step 5	[**no**] **switchport block** {**multicast** | **unicast**}
	Example:
	`Router(config-if)# switchport block multicast`
	`Router(config-if)# switchport block unicast`
		Prevents the flooding of unknown multicast or unicast packets on the specified interface.
		Use the no form of this command to resume normal forwarding on the port.

Step 6	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 7	(Optional) **show interface** {*interface-id* | **port-channel** *number*} **switchport** **Example:** `Router# show interface gigabitethernet 0/1/1 switchport`	(Optional) Displays the switchport blocking configuration.

Example

The following example shows how to block multicast and unicast flooding on GigabitEthernet interface 0/1/1 and how to verify the configuration:

```
Router# configure terminal
Router(config)# interface GigabitEthernet0/1/1
Router(config-if)# switchport access vlan 20
Router(config-if)# switchport mode access
Router(config-if)# switchport block multicast
Router(config-if)# switchport block unicast
Router(config-if)# exit
Router(config)# end
Router#
```

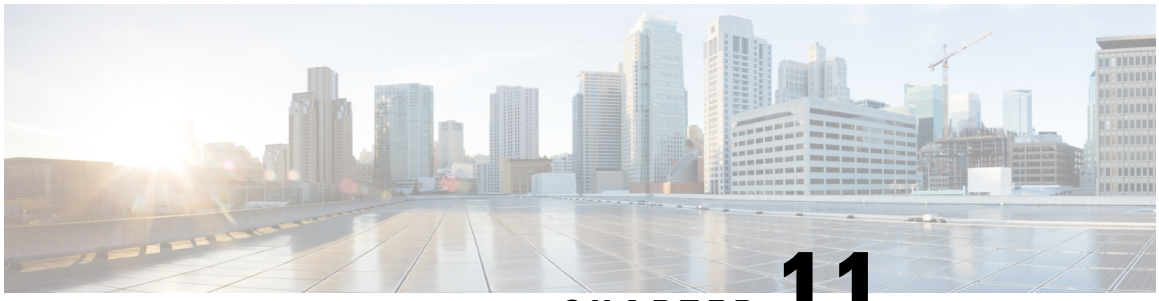
Following command shows the blocking state of unknown unicast and multicast on the interface:

```
Router#show interfaces gigabitEthernet 0/1/1 switchport
Name: Gi0/1/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 20 (VLAN0020)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Voice VLAN: none
```

```
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: enabled
Unknown multicast blocked: enabled
Appliance trust: none
Router#
```

Configuring Switchport Blocking



CHAPTER 11

Configuring Storm Control

- Information About Storm Control, on page 153
- Configuring Storm Control, on page 154

Information About Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. The traffic broadcast and multicast suppression (or storm control) feature prevents LAN ports from being disrupted by a broadcast, multicast and unicast traffic storm on physical interfaces.

A broadcast storm occurs when huge amount of broadcast, multicast, or unknown unicast packets flood the LAN, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation or in the network configuration can also cause a storm. The mechanism to prevent and control such events is known as storm control or broadcast suppression.

Broadcast and Multicast Suppression monitors incoming traffic levels over a 1-second traffic storm control interval and, during the interval compares the traffic level with the traffic storm control level configured. The traffic storm control threshold level is a percentage of the total available bandwidth of the port. Each port has different storm control levels for broadcast, multicast, and unicast type of traffic.

Storm control uses rising and falling thresholds to block and then restore the forwarding of broadcast, unicast, or multicast packets.

- The rising threshold is the traffic limit after which, that particular traffic is blocked.
- The falling threshold is the traffic limit below which, that particular starts forwarding again, if it was already blocked.

Note If a particular type of ingress traffic (unicast, broadcast and multicast) is more than the rising threshold configured on it, the interface goes to blocked state for that particular traffic.

Storm control prevents traffic on a LAN from being disrupted by a broadcast, multicast, or unicast storm on a port. Storm control is applicable for physical interfaces and is used to restrict the unicast, broadcast and multicast ingress traffic on the Layer2 interfaces.

Storm control for unicast is a combination of known unicast and unknown unicast traffic. When storm control for unicast is configured, and it exceeds the configured value, the storm will hit each type of traffic through

the hardware policer. The following example describes how the unicast traffic is filtered, when the configured storm is 10%:

- Incoming traffic is unknown unicast 8% + known unicast 7%. Total of 15% storm is not filtered in hardware by the hardware policer.
- Incoming traffic is unknown unicast 11% + known unicast 7%. Total of 18% storm will hit unknown unicast traffic type, and the hardware policer will filter unknown traffic that exceeds 11%.
- Incoming traffic is unknown unicast 11% + known unicast 11%. Total of 22% storm will hit unknown unicast traffic and known unicast traffic, and the hardware policer will filter both unknown and unknown unicast traffic.

Configuring Storm Control

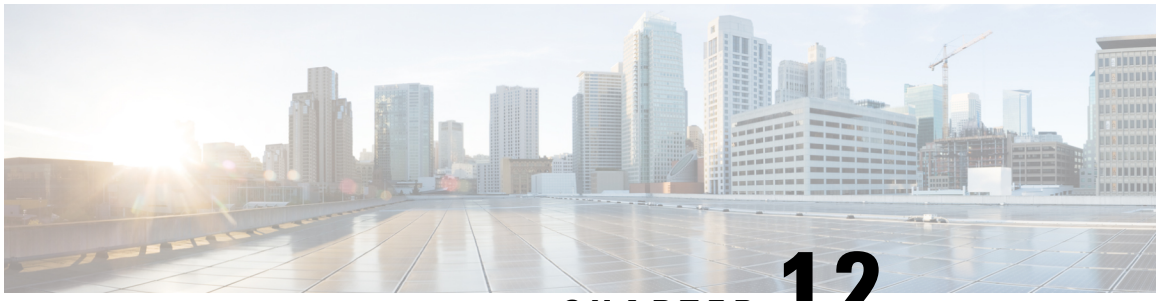
You can set the percentage of total available bandwidth that the controlled traffic can use.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Enters interface configuration mode.
Step 4	**storm-control** {**broadcast** | **multicast** | **unicast** | **unknown-unicast**} **level** {*level* [*level-low*] | **bps** *bps* [*bps-low*] | **pps** *pps* [*pps-low*]} **Example:**	Configure broadcast, multicast, unicast or unknown-unicast control. By default, storm control is disabled. • For *level*, specify the rate limit for broadcast, multicast, or unicast traffic as a percentage of the bandwidth. The port suppresses traffic when the rising threshold is reached. For optional *level_low*, specify the low level of the rate limit, as a percentage of the bandwidth. When action SNMP trap is enabled, and the traffic rate exceeds the level then drops below the *level_low*, the port will send out an SNMP trap.

	Command or Action	Purpose
		Note The optional *level-low* will enabled. The minimum acceptable le If level 0 is configured, it w • For bps *bps*, specify the rising threshold level for broadcast, multicast, or unicast traffic in bits per second (up to one decimal place). The port blocks traffic when the rising threshold is reached. The range is 0.0 to 10000000000.0. (Optional) For *bps-low*, specify the falling threshold level in bits per second (up to one decimal place). It can be less than or equal to the rising threshold level. The port forwards traffic when traffic drops below this level. The range is 0.0 to 10000000000.0. • For pps *pps*, specify the rising threshold level for broadcast, multicast, or unicast traffic in packets per second (up to one decimal place). The port blocks traffic when the rising threshold is reached. The range is 0.0 to 10000000000.0. (Optional) For *pps-low* , specifies the falling threshold level in packets per second (up to one decimal place). It can be less than or equal to the rising threshold level. The port forwards traffic when traffic drops below this level. The range is **0.0 to** 10000000000.0. **Note** Do not configure both **storm** interface. For BPS and PPS settings, y
Step 5	**storm-control action** {**shutdown** | **trap**} **Example:** `Router(config-if)# storm-control action trap`	If none of shutdown or SNMP trap is configured, by default the traffic will be suppressed when traffic exceeds the threshold specified by *level*.

	Command or Action	Purpose
		If *shutdown* is configured, the interface will enter err-disable when traffic exceeds the threshold specified by *level*. If *trap* is configured, the interface will send SNMP trap when traffic exceeds the threshold specified by level. And when traffic drops below the *level_low*, another SNMP trap will be sent out.
Step 6	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show storm-control** [*interface-id*] [**broadcast** | **multicast** | **unicast** | **unknown-unicast**]	Verify the storm control rate limit set on the interface for the specified traffic type.
Step 8	**exit** **Example:** `Router(config)# exit`	Returns the router to global configuration mode.



CHAPTER 12

Configuring MAC Address Notification

- MAC Address Notification, on page 157
- Configuring MAC Address Notification, on page 157

MAC Address Notification

This feature enables the user or administrator to keep track of the MAC addresses that are learned or removed on the Layer 2 switch while forwarding the Ethernet frames. This feature is required to keep a history of the MAC addresses that are learned and deleted from the router and generate notifications to the NMS periodically.

Whenever a new MAC address is learned or an old MAC address is removed, a SNMP notification is generated and sent to the NMS. A history table is also maintained for every hardware port, so that NMS can collect information by querying the MIB for the history table. This is done to make sure even when the notifications are not delivered to the NMS properly; the data is preserved on the router for the NMS to collect.

Note This feature will generate MAC notifications only for dynamic addresses. No notifications are generated for self, static, or multicast addresses.

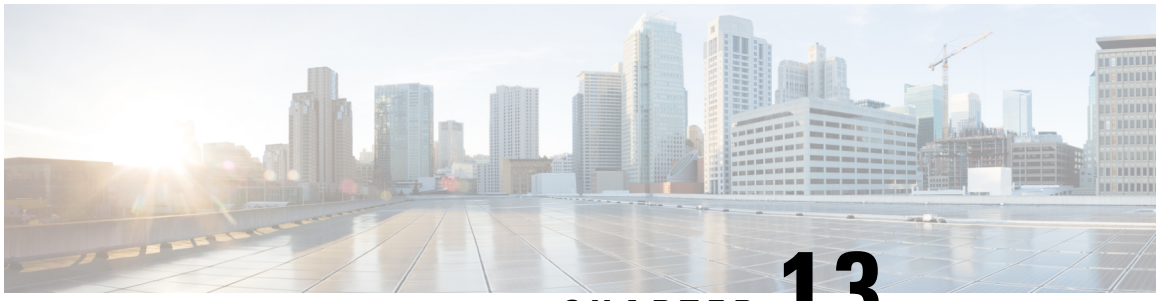
Configuring MAC Address Notification

Follow these steps to configure MAC address notification:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	[**no**] **mac-address-table notification change mac-move** [**interval** *value*] | [**historysize** *value*] | **mac-move**	Enable MAC notification feature. It is disabled by default.

	Command or Action	Purpose
		• **interval** *value*—Sets the notification trap interval in seconds. The switch will dispatch the notification trap only after this value is elapsed. Default is 1 second. **historysize** *value*—Configures the maximum number of entries in the MAC notification history table. The old table will be deleted and a new table will be created when this command is issued. Default size is 1. **mac-move**—Enables MAC move notification.
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Enters interface configuration mode, and specifies the Layer 2 interface on which to enable the SNMP MAC address notification trap.
Step 4	[**no**] **snmp trap mac-notification** {**added** | **removed**} **Example:** `Router(config-if)# snmp trap mac-notification change added`	After MAC notification is enabled globally, use this command to enable/disable MAC notification traps on a particular port. By default it is disabled. • **added**—Enables MAC notification trap when an address is added on this port. • **removed**—Enables MAC notification trap when an address is removed from this port.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 6	[**no**] **snmp-server enable traps mac-notification**	The actual notification traps will be sent only after this command is entered even if MAC notification is enabled globally and at the port. It is disabled by default.
Step 7	**show mac-address-table notification** [**interface** *interface-id*]	Verify if the feature is enabled or disabled, and display MAC notification interval and the history table.



CHAPTER 13

Configuring Q-in-Q and Layer 2 Protocol Tunneling

This chapter describes how to configure IEEE 802.1Q-in-Q VLAN tunnels and Layer 2 protocol tunneling on the Cisco IR8340 Routers.

- Information About Q-in-Q Tunnels, on page 159
- Information About Layer 2 Protocol Tunneling, on page 160
- Configuring VLAN Mapping for Selective Q-in-Q on a 802.1Q Tunnel Port, on page 160
- Enabling the Layer 2 Protocol Tunnel, on page 162
- Configuring Thresholds for Layer 2 Protocol Tunnel Ports, on page 163
- Verifying the Q-in-Q Configuration, on page 164
- VLAN Translation One-to-One Mapping, on page 164

Information About Q-in-Q Tunnels

A Q-in-Q VLAN tunnel enables a service provider to segregate the traffic of different customers in their infrastructure, while still giving the customer a full range of VLANs for their internal use by adding a second 802.1Q tag to an already tagged frame.

Business customers of service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and the traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit of 4096 of the 802.1Q specification.

Using the 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved and the traffic from different customers is segregated within the service-provider infrastructure even when they appear to be on the same VLAN. The 802.1Q tunneling expands the VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that is dedicated to tunneling. Each customer requires a separate VLAN, but that VLAN supports all of the customer's VLANs.

Customer traffic that is tagged in the normal way with appropriate VLAN IDs come from an 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge switch. The link between the customer device and the edge switch is an asymmetric link because one end is configured as an 802.1Q trunk

port and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer.

Packets that enter the tunnel port on the service-provider edge switch, which are already 802.1Q-tagged with the appropriate VLAN IDs, are encapsulated with another layer of an 802.1Q tag that contains a VLAN ID that is unique to the customer. The original 802.1Q tag from the customer is preserved in the encapsulated packet. Therefore, packets that enter the service-provider infrastructure are double-tagged.

The outer tag contains the customer's access VLAN ID (as assigned by the service provider), and the inner VLAN ID is the VLAN of the incoming traffic (as assigned by the customer).

Information About Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. The Spanning Tree Protocol (STP) must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. The Cisco Discovery Protocol (CDP) must be able to discover neighboring Cisco devices from local and remote sites, and the VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

When protocol tunneling is enabled, edge switches on the inbound side of the service-provider infrastructure encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider network. Core switches in the network do not process these packets, but forward them as normal packets. Bridge protocol data units (BPDUs) for CDP, STP, or VTP cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs.

If protocol tunneling is not enabled on 802.1Q tunneling ports, remote switches at the receiving end of the service-provider network do not receive the BPDUs and cannot properly run STP, CDP, 802.1X, and VTP. When protocol tunneling is enabled, Layer 2 protocols within each customer's network are totally separate from those running within the service-provider network. Customer switches on different sites that send traffic through the service- provider network with 802.1Q tunneling achieve complete knowledge of the customer's VLAN.



Note Layer 2 protocol tunneling works by tunneling BPDUs in the software. A large number of BPDUs that come into the supervisor will cause the CPU load to go up. You might need to make use of software rate limiters to reduce the load on the supervisor CPU. See Configuring Thresholds for Layer 2 Protocol Tunnel Ports, on page 163.

Configuring VLAN Mapping for Selective Q-in-Q on a 802.1Q Tunnel Port

To configure VLAN mapping for selective Q-in-Q on a 802.1Q tunnel port, complete the following steps.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	[**no**] **vlan dot1q tag native** **Example:** `Router(config)# vlan dot1q tag native`	Enable or disable native VLAN tagging on trunk port.
Step 4	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Enters interface configuration mode for the interface connected to the service provider network.
Step 5	[**no**] **switchport mode dot1q-tunnel** **Example:** `Router(config-if)# switchport mode dot1q-tunnel`	Configure the interface as an IEEE 802.1Q tunnel port.
Step 6	[**no**] **switchport access vlan** *vlan id* **Example:** `Router(config-if)# switchport access vlan 20`	Configure default VLAN used as S-VLAN on dot1q-tunnel ports.
Step 7	[**no**] **switchport vlan mapping default dot1q-tunnel** *outer vlan-id*	Configure VLAN mapping so that all packets entering the port are bundled into the specified S-VLAN: • *outer vlan-id*—Enter the outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094.
Step 8	[**no**] **switchport vlan mapping** *vlan-id* **dot1q-tunnel** *outer vlan-id*	Enters the VLAN IDs to be mapped: • *vlan-id*—The customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs.

	Command or Action	Purpose
		• *outer vlan-id*—Enter the outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits the configuration mode.

Example

What to do next

Use the **no switchport vlan mapping all** command to remove the VLAN mapping configuration.

Enabling the Layer 2 Protocol Tunnel

You can enable protocol tunneling on the 802.1Q tunnel port.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Enters interface configuration mode.

Step 4	[**no**] **l2protocol-tunnel** [**cdp** | **vtp** | **stp**]
	Example:
	`Router(config-if)# l2protocol-tunnel stp`
	`Router(config-if)# no l2protocol-tunnel cdp`
		Enable or disable Layer 2 protocol tunneling. Optionally, you can enable CDP, STP, or VTP tunneling.

Step 5	**exit** **Example:**	Exits the configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	

Configuring Thresholds for Layer 2 Protocol Tunnel Ports

You can specify the port drop and shutdown value for a Layer 2 protocol tunneling port.

When a drop threshold is enabled on a tunneling interface, the interface will drop any incoming PDU after the specified threshold for that protocol is reached. Similarly, when a shutdown threshold is enabled, the interface will be error-disabled when the threshold is exceeded. That effectively stops the interface from forwarding any packet. To enable the interface again, a user has to do 'shut' and 'no shut' to the interface. The unit of measure of both the thresholds is the number of packets per second. The valid range for configuration is between 1 to 4096. Both shutdown threshold and drop threshold can be enabled on a same interface for a same protocol. But shutdown threshold must be larger than drop threshold there.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Enters interface configuration mode.

Step 4	[**no**] **l2protocol-tunnel** {**drop-threshold** | **shutdown-threshold**} [**cdp** | **vtp** | **stp**] *packets-per-sec*
	Example:
	`Router(config-if)# l2protocol-tunnel shutdown-threshold stp 2000`
	`Router(config-if)# l2protocol-tunnel drop-threshold stp 2500`
	`Command rejected: protocol tunneling shutdown threshold must be greater than or equal to the drop threshold.`
	`Router(config-if)# l2protocol-tunnel drop-threshold stp 1500`
		drop-threshold—Specifies the maximum number of packets that can be processed on an interface before being dropped.
		shutdown-threshold—Specifies the maximum number of packets that can be processed on an interface. When the number of packets is exceeded, the port is put in error-disabled state.
		Optionally, you can specify CDP, STP, or VTP. Valid values for the packets are from 1 to 4096.
		Use the **no** form of the command to resets the threshold values to 0 and disable the shutdown threshold.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits the configuration mode.

Verifying the Q-in-Q Configuration

Command	Purpose
show dot1q-tunnel	Display all ports in dot1q-tunnel mode.
show vlan mapping [*interface-id*]	Display the VLAN mapping information for all interfaces or for the interface specified.

VLAN Translation One-to-One Mapping

VLAN translation provides the capability of carrying the customers traffic in single tagged packets across the service provider network. Since VLAN translation and selective QinQ are applied to a trunk port, the service provider gets the added benefit of being able to selectively drop or bundle all traffic that does not belong to a given set of C-VLANs and bridge accordingly in the trunk.

VLAN translation on trunk ports supports 1:1 C-VLAN to S-VLAN mapping. C-VLAN received on customer side trunk port is stripped and mapped S-VLAN is added.

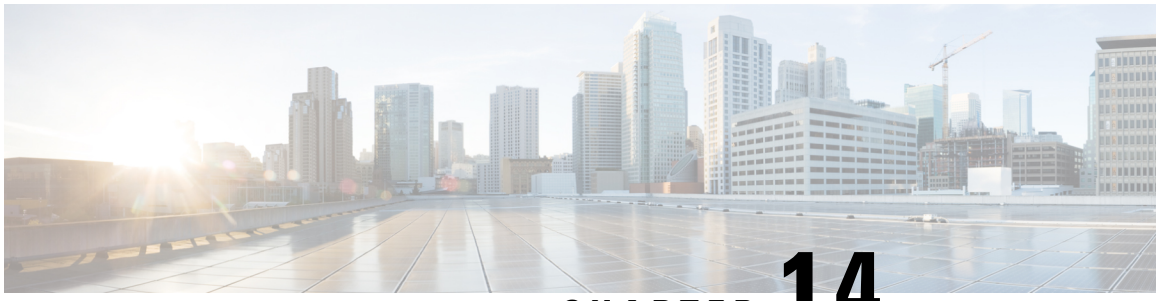
The SP provides L2VPN service to two different customers, Customer A and Customer B. The SP needs to keep the data as well as control traffic between the two customers separate from each other and also from the SP`s own control traffic. The SP network also needs to be transparent to the customer edge devices. Several mechanisms are available to keep the customer's VLAN ID space intact across the SP network.

IEEE802.1Q (Port-based QinQ) is one such mechanism wherein all the packets received on a *tunnel* port are tunneled through the SP network with the same outer tag. While such a mechanism is sufficient, it is rather restrictive and inflexible because it allows the SPs only port level granularity in providing the services.

The mechanisms described in the following sections and summarized in the following table give the SPs the ability to provide a more flexible and finer granular level of service. In all the following cases, the mappings of C-VLANs to S-VLANs at the ingress of the SP network and the correct mapping back from S-VLANs to C-VLANS at the egress of SP network relies on the proper configuration of the mappings and reverse mappings.

```
interface GigabitEthernet0/1/2
 switchport mode trunk
 switchport vlan mapping 10 100
 switchport vlan mapping 5 150

interface GigabitEthernet0/1/2
 switchport mode trunk
 switchport vlan mapping 10 dot1q-tunnel 100
 switchport vlan mapping 2-8 dot1q-tunnel 150
 switchport vlan mapping default dot1q-tunnel 300
```



CHAPTER 14

Layer 2 Tunneling Protocol Version 3

- Layer 2 Tunneling Protocol Version 3, on page 165
- Restrictions for Layer 2 Tunneling Protocol Version 3, on page 166
- Information About Layer 2 Tunneling Protocol Version 3, on page 168
- L2TPv3 Operation, on page 169
- L2TPv3 Features, on page 171
- How to Configure Layer 2 Tunneling Protocol Version 3, on page 182
- Configuration Examples for Layer 2 Tunneling Protocol Version 3, on page 206
- Additional References, on page 213
- Glossary, on page 214

Layer 2 Tunneling Protocol Version 3

The Layer 2 Tunneling Protocol Version 3 feature expands Cisco's support of Layer 2 VPNs. Layer 2 Tunneling Protocol Version 3 (L2TPv3) is an IETF l2tpext working group draft that provides several enhancements to L2TP to tunnel any Layer 2 payload over L2TP. Specifically, L2TPv3 defines the L2TP protocol for tunneling Layer 2 payloads over an IP core network by using Layer 2 VPNs.

Prerequisites for Layer 2 Tunneling Protocol Version 3

- Before you configure an xconnect attachment circuit for a provider edge (PE) device , the Cisco Express Forwarding (formerly known as CEF) feature must be enabled. To enable Cisco Express Forwarding on an interface, use the **ip cef** or **ip cef distributed** command.
- You must configure a loopback interface on the router for originating and terminating the L2TPv3 traffic. The loopback interface must have an IP address that is reachable from the remote PE device at the other end of an L2TPv3 control channel.
- 800 L2TPv3 sessions are supported on the Cisco 1000 Series Integrated Services in the below format:
- 800 L2TPv3 sessions ---- 800 loopbacks ---- 800 vlans ---- 800 SVIs

Note Recommended L2TPv3 sessions - 200

Restrictions for Layer 2 Tunneling Protocol Version 3

General L2TPv3 Restrictions

- Cisco Express Forwarding must be enabled for the L2TPv3 feature to function. The xconnect configuration mode is blocked until Cisco Express Forwarding is enabled. On distributed platforms, such as the Cisco 7500 series, if Cisco Express Forwarding is disabled while a session is established, the session is torn down. The session remains down until Cisco Express Forwarding is reenabled. To enable Cisco Express Forwarding, use the **ip cef** or **ip cef distributed** command.
- The number of sessions on PPP, High-Level Data Link Control (HDLC), Ethernet, or 802.1q VLAN ports is limited by the number of interface descriptor blocks (IDBs) that the router can support. For PPP, HDLC, Ethernet, and 802.1q VLAN circuit types, an IDB is required for each circuit.
- When L2TPv3 is used to tunnel Frame Relay D channel data-link connection identifiers (DLCIs), an IDB is not required for each circuit. As a result, the memory requirements are much lower. The scalability targets for the Engineering Field Test (EFT) program are 4000 L2TP session.
- To convert an interface with Any Transport over MPLS (AToM) xconnect to L2TPv3 xconnect, remove the AToM configuration from the interface and then configure L2TPv3. Some features may not work if L2TPv3 is configured before removing the AToM configuration.
- Frame Relay support includes only 10-bit DLCI addressing. The L2TPv3 feature does not support Frame Relay extended addressing.
- The interface keepalive feature is automatically disabled on the interface to which xconnect is applied, except for Frame Relay encapsulation, which is required for Local Management Interface (LMI).
- Static L2TPv3 sessions do not support Frame Relay LMI interworking.
- Static L2TPv3 sessions do not interoperate with Universal Tunnel Interface (UTI) using keepalives.
- Layer 2 fragmentation of IP packets and Intermediate System-to-Intermediate System (IS-IS) fragmentation through a static L2TPv3 session are not supported.
- Layer 3 fragmentation is not recommended because of performance degradation.
- The L2TPv3 Layer 2 (IP packet) fragmentation feature (see the Configuring the L2TPv3 Pseudowire task) is not supported when the customer edge (CE) router is running special Layer 2 options such as Layer 2 sequencing, compression, or encryption. Examples of these options are Frame Relay compression and fragmentation or PPP compression. In these scenarios, the IP payload is not in a format that is compatible with IP fragmentation.
- The Stateful Switchover (SSO), Route Processor Redundancy (RPR) and RPR+ components of the HA functions are supported only at the coexistence level. If you attempt a switchover using SSO, RPR, or RPR+, the tunnels will fail and then eventually recover after an undetermined time duration. This includes both IPv4 and IPv6 traffic.
- Interworking is not allowed when sequencing is enabled.
- Untagged packets (native VLAN) forwarding for xconnect that is configured on the dot1q subinterface is not supported.

- L2TPv3 xconnect is not supported on an EtherSwitch module. This limitation is also applicable to switch virtual interfaces (SVI) that are physically terminated on an EtherSwitch module interface.
- Only Ethernet, HDLC, Frame Relay and VLAN (802.1Q, QinQ, and QinAny) attachment circuits are supported; EVC is not supported.
- The IP local interface must be a loopback interface and the loopback interface cannot be in a VRF. Configuring any other interface with the "ip local interface" command results in a nonoperational setting.
- When utilizing the Cisco Cyber Vision (CCV) Sensor on the service CPU profile, WAN/LAN traffic performance decreases.

VLAN-Specific Restrictions

- A PE device is responsible only for static VLAN membership entries that are configured manually on the device. Dynamic VLAN membership entries, entry aging, and membership discovery are not supported.
- Implicit tagging for VLAN memberships operating on other layers, such as membership by MAC address, protocol type at Layer 2, or membership by IP subnet at Layer 3, is not supported.
- Point-to-multipoint and multipoint-to-point configurations are not supported. There is a 1:1 relationship between an attachment circuit and an L2TPv3 session.

IPv6 Protocol Demultiplexing for L2TPv3 Restrictions

- IPv6 protocol demultiplexing is supported only for Ethernet traffic.
- IPv6 protocol demultiplexing is supported over noninterworking sessions.

L2TPv3 Control Message Hashing Restrictions

- L2TPv3 control channel authentication configured using the **digest** command requires bidirectional configuration on the peer devices. A shared secret must be configured on the communicating nodes.
- For a compatibility matrix of all the L2TPv3 authentication methods, see the Valid Configuration Scenarios table in the IPv6 Protocol Demultiplexing section.

L2TPv3 Digest Secret Graceful Switchover Restrictions

- This feature works only with authentication passwords configured using the L2TPv3 Control Message Hashing feature. L2TPv3 control channel authentication passwords configured with the older, Challenge Handshake Authentication Protocol (CHAP)-like authentication system cannot be updated without tearing down L2TPv3 tunnels and sessions.
- In Cisco IOS Release 12.0(30)S, a maximum of two passwords can be configured simultaneously using the **digest secret** command.

For more information about the L2TPv3 Control Message Hashing feature, see the L2TPv3 Control Message Hashing section.

Quality of Service Restrictions in L2TPv3 Tunneling

Quality of service (QoS) policies configured with the modular QoS command-line interface (MQC) are supported in L2TPv3 tunnel sessions with the following restrictions: Protocol demultiplexing requires a combination of an IP address and the xconnect command configured on the interface. The interface is then treated as a regular L3. To apply QoS on the Layer 2 IPv6 traffic, you must classify the IPv6 traffic into a separate class before applying any feature(s) to it. The following match criteria are used to classify Layer 2 IPv6 traffic on a protocol demultiplexing interface:

The following match criterion is used to classify Layer 2 IPv6 traffic on a protocol demultiplexing interface:

```
class-map match-ipv6
     match protocol ipv6
```

In the absence of a class to handle Layer 2 IPv6 traffic, the service policy is not accepted on a protocol demultiplexing interface.

For detailed information about QoS configuration tasks and command syntax, refer to:

- *Cisco IOS Quality of Service Solutions Configuration Guide*
- *Cisco IOS Quality of Service Solutions Command Reference*

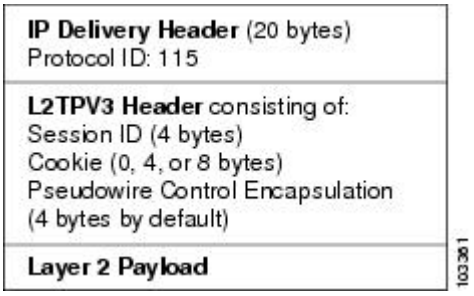
Information About Layer 2 Tunneling Protocol Version 3

L2TPv3 provides a method for delivering L2TP services over an IPv4 (non-UDP) backbone network. It encompasses the signaling protocol as well as the packet encapsulation specification.

L2TPv3 Header Description

The migration from UTI to L2TPv3 also requires the standardization of the UTI header. As a result, the L2TPv3 header has the new format shown in the figure below.

Figure 2: L2TPv3 Header Format



Each L2TPv3 packet contains an L2TPv3 header that includes a unique session ID representing one session and a variable cookie length. The L2TPv3 session ID and the Tunnel Cookie field length are assigned through the CLI. See the How to Configure Layer 2 Tunneling Protocol Version 3 section for more information on the CLI commands for L2TPv3.

Session ID

The L2TPv3 session ID identifies the session context on the decapsulating system. For dynamic sessions, the value of the session ID is selected to optimize the context identification efficiency of the decapsulating system. A decapsulation implementation may, therefore, elect to support a smaller session ID bit field. In this L2TPv3 implementation, an upper value for the L2TPv3 session ID was set at 023. The L2TPv3 session ID value 0 is reserved for use by the protocol. For static sessions, the session ID is manually configured.



Note The local session ID must be unique on the decapsulating system and is restricted to the least significant ten bits.

Session Cookie

The L2TPv3 header contains a control channel cookie field. The control channel cookie field has a variable length of 0, 4, or 8 bytes according to the cookie length supported by a given platform for packet decapsulation. The control channel cookie length can be configured manually for static sessions or determined dynamically for dynamic sessions.

The variable cookie length does not present a problem when the same platform is at both ends of an L2TPv3 control channel. However, when different platforms interoperate across an L2TPv3 control channel, both platforms need to encapsulate packets with a 4-byte cookie length.

Pseudowire Control Encapsulation

The L2TPv3 pseudowire control encapsulation consists of 32 bits (4 bytes) and contains information used to sequence L2TP packets (see the Sequencing section). For the purposes of sequencing, only the first bit and bits 8 to 31 are relevant. Bit 1 indicates whether the Sequence Number field, bits 8 to 31, contains a valid sequence number and is to be updated.

L2TPv3 Operation

L2TPv3 includes the following features:

- Xconnect for Layer 2 tunneling through a pseudowire over an IP network
- Layer 2 VPNs for PE-to-PE device service using xconnect that supports Ethernet and VLAN, including both static and dynamic (using the new L2TPv3 signaling) forwarded sessions

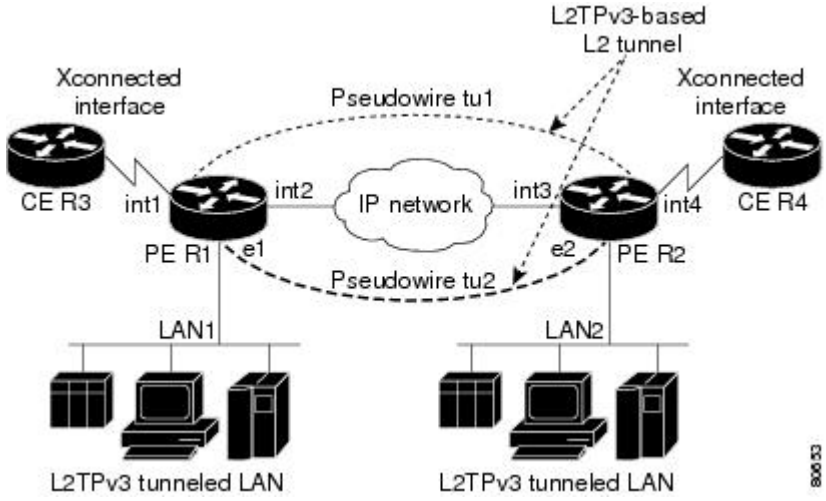
The initial Cisco IOS software supported only the following features:

- Layer 2 tunneling (as used in an L2TP access concentrator or LAC) to an attachment circuit, not Layer 3 tunneling
- L2TPv3 data encapsulation directly over IP (IP protocol number 115), not using the UDP
- Point-to-point sessions, not point-to-multipoint or multipoint-to-point sessions
- Sessions between the same Layer 2 protocols, such as Ethernet-to-Ethernet and VLAN-to-VLAN, but not VLAN-to-Ethernet

The attachment circuit is the physical interface or subinterface attached to the pseudowire.

The figure below shows how the L2TPv3 feature is used for setting up VPNs using Layer 2 tunneling over an IP network. All traffic between two customer network sites is encapsulated in IP packets carrying L2TP data messages and sent across an IP network. The backbone devices of the IP network treat the traffic as any other IP traffic and need not know anything about the customer networks.

Figure 3: L2TPv3 Operation



In the figure above, the PE devices R1 and R2 provide L2TPv3 services. The R1 and R2 devices communicate with each other using a pseudowire over the IP backbone network through a path comprising interfaces **int1** and **int2**, the IP network, and interfaces **int3** and **int4**.

Note When you configure SVI on the PE devices, the interfaces **int1** and **int4** act as LAN switching ports

The PE devices communicate with each other using pseudowires (tu1 and tu2) through a path comprising SVIs over an IP network, while the CE devices communicate through a pair of Xconnect Ethernet or VLAN interfaces using an L2TPv3 sessions.

The L2TPv3 session - tu1 is a pseudowire configured between interface **int1** on PE-R1 and interface **int4** on PE-R2. Any traffic arriving on interface **int1** on PE-R1 from CE-R3 is encapsulated and sent through the pseudowire, which is the control channel (tu1) to PE-R2, where the information is decapsulated and sent to CE-R4 from interface **int4** on P2-R2. When CE-R4 needs to send information to CE-R3, the traffic follows the same path, but, in reverse.

Note

- All packets received on interface int1 are forwarded to R4. R3 and R4 cannot detect the intervening network.
- For Ethernet interfaces, any packet received from LAN1 by R1 on Ethernet interface e1 is encapsulated directly in IP and sent through the pseudowire session tu2 to R2 interface e2, where it is sent on LAN2.
- A VLAN on an Ethernet interface can be mapped to an L2TPv3 session.

L2TPv3 Features

L2TPv3 provides xconnect support for Ethernet and VLAN using Static and Dynamic sessions.

Static L2TPv3 Sessions

Typically, the L2TP control plane is responsible for negotiating session parameters, such as the session ID or the cookie, to set up the session. However, some IP networks require sessions to be configured so that no signaling is required for session establishment. You can set up static L2TPv3 sessions for a PE device by configuring fixed values for the fields in the L2TP data header. A static L2TPv3 session allows the PE device to tunnel Layer 2 traffic as soon as the attachment circuit to which the session is bound comes up.

Static configuration allows sessions to be established without dynamically negotiating control connection parameters. This means that although sessions are displayed in the **show l2tun session** command output, no control channel information is displayed in the **show l2tun tunnel** command output.



Note In an L2TPv3 static session, you can still run the L2TP control channel to perform peer authentication and dead-peer detection. If the L2TP control channel cannot be established or is torn down because of a hello failure, the static session is also torn down.

If you use a static L2TPv3 session, you cannot perform circuit interworking, such as LMI, because there is no facility to exchange control messages. To perform circuit interworking, you must use a dynamic session.

Dynamic L2TPv3 Sessions

A dynamic L2TP session is established through the exchange of control messages containing attribute-value (AV) pairs. Each AV pair contains information about the nature of the Layer 2 link being forwarded, including the payload type and virtual circuit (VC) ID.

Multiple L2TP sessions, one for each forwarded Layer 2 circuit, can exist between a pair of PE devices and can be maintained by a single control channel. Session IDs and cookies are dynamically generated and exchanged as part of a dynamic session setup. Information such as sequencing configuration is also exchanged. Circuit state changes (UP/DOWN) are conveyed using the set link info (SLI) message.

Control Channel Parameters

The L2TP class configuration procedure creates a template of L2TP control channel parameters that can be inherited by different pseudowire classes. L2TP control channel parameters are used in control channel authentication, keepalive messages, and control channel negotiation. In an L2TPv3 session, the same L2TP class must be specified in the pseudowire configured on the PE device at each end of the control channel. Configuring L2TP control channel parameters is optional. However, the L2TP class must be configured before it is associated with a pseudowire class (see the Configuring the L2TPv3 Pseudowire task).

L2TPv3 Control Channel Authentication Parameters

Two methods of control channel message authentication are available: the L2TPv3 Control Message Hashing feature and CHAP-style L2TP control channel. The L2TPv3 Control Message Hashing feature introduces a more robust authentication method than the older, CHAP-style L2TP control channel method of authentication. You may choose to enable both the methods of authentication to ensure interoperability with peers that support only one of these methods of authentication, but this configuration will yield control of the authentication

method used on the peer PE device. Enabling both the methods of authentication should be considered as an interim solution to solve backward compatibility issues during software upgrades.

The principal difference between the two methods of authentication lies in the L2TPv3 Control Message Hashing feature using the entire message in the hash instead of computing the hash over selected contents of a received control message. In addition, instead of including the hash digest in only the start control channel replay (SCCRP) and start control channel connected (SCCCN) messages, it includes it in all messages.

Support for L2TP control channel authentication is maintained for backward compatibility. Either or both authentication methods can be enabled to allow interoperability with peers supporting only one of the authentication methods.

The table below shows a compatibility matrix for the different L2TPv3 authentication methods. PE1 is running the new authentication method. The possible authentication configurations for PE1 are shown in the first column. The other columns represent PE2 running software with different available authentication options. The tables cells in these columns indicate compatible configuration options for PE2. If any PE1/PE2 authentication configuration poses ambiguity about the authentication method used, the winning authentication method is indicated in bold. If both the old and new authentication methods are enabled on PE1 and PE2, both types of authentication occur.

Table 15: Compatibility Matrix for L2TPv3 Authentication Methods

PE1 Authentication Configuration	PE2 Supporting Old Authentication[1]	PE2 Supporting New Authentication[2]	PE2 Supporting Old and New Authentication[3]
None	None	None New integrity check	None New integrity check
Old authentication	Old authentication	—	Old authentication **Old authentication** and new authentication **Old authentication** and new integrity check
New authentication	—	New authentication	New authentication Old authentication and **new authentication**
New integrity check	None	None New integrity check	None New integrity check
Old and new authentication	Old authentication	New authentication	Old authentication New authentication **Old and new authentication** **Old authentication** and new integrity check

PE1 Authentication Configuration	PE2 Supporting Old Authentication[1]	PE2 Supporting New Authentication[2]	PE2 Supporting Old and New Authentication[3]
Old authentication and new integrity check	Old authentication	—	Old authentication **Old authentication** and new authentication **Old authentication** and new integrity check

1 Any PE software that supports only the old CHAP-like authentication system
2 Any PE software that supports only the new message digest authentication and integrity checking authentication system, but does not understand the old CHAP-like authentication system. This type of software may be implemented by other vendors based on the latest L2TPv3 draft.
3 Any PE software that supports both the old CHAP-like authentication and the new message digest authentication and integrity checking authentication system

Ethernet over L2TPv3

The Ethernet over L2TPv3 feature provides support for Ethernet-based Layer 2 payload tunneling over IP core networks using L2TPv3.

The Ethernet over L2TPv3 feature supports the following like-to-like switching modes:

- Ethernet port mode
- Ethernet VLAN mode
- Ethernet VLAN mode with VLAN rewrite
- Ethernet QinQ and QinAny mode

Note The QinQ over L2TPv3 support feature includes QinAny over L2TPv3, which has a fixed outer VLAN tag and a variable inner VLAN tag.

The Ethernet over L2TPv3 feature supports the following types of internetworking:

- Ethernet port to VLAN (routed)
- Ethernet port to VLAN (bridged)
- QinQ to Ethernet VLAN or Port Interworking (routed)
- QinQ to Ethernet VLAN or Port Interworking (bridged)

Note QinAny Interworking is not a valid configuration because the inner VLAN tag is undetermined.

GEC over L2TPv3

Gigabit EtherChannel (GEC) over Layer 2 Tunneling Protocol Version 3 (L2TPv3) provides support for GEC-based Layer 2 payload tunneling over IP core networks using L2TPv3. GEC also known as *port channel* is integrated with Ethernet and dot1q attachment circuits (ACs).

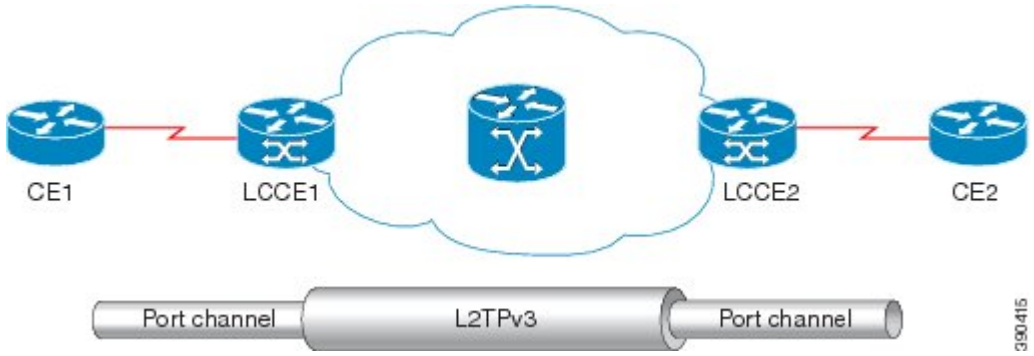
A port channel bundles physical links into a channel group to create a single logical link that provides the aggregate bandwidth of up to eight physical links. If a member port within a port channel fails, the traffic previously carried over the failed link switches to the remaining member ports within the port channel.

Interworking switching is supported in the following scenarios:

- The customer-edge-provider-edge (CE-PE) connecting interface on the local PE is a port-channel interface without dot1q encapsulation. The CE-PE connecting interface on the remote PE is a port-channel interface with dot1q encapsulation.
- The CE-PE connecting interface on the local PE is a port-channel interface with or without dot1q encapsulation. The CE-PE connecting interface on the remote PE is an Ethernet interface with or without dot1q encapsulation.

The figure below illustrates a port channel over IP core networks using L2TPv3. CE1 and CE2 are connected to L2TP Control Connection Endpoints (LCCE) and through port channels. The LCCE is connected to the IP core network using L2TPv3.

Figure 4: GEC over L2TPv3



Sequencing

Although the correct sequence of received Layer 2 frames is guaranteed by some Layer 2 technologies (by the nature of the link such as a serial line) or by the protocol itself, forwarded Layer 2 frames may be lost, duplicated, or reordered when they traverse a network as IP packets. If the Layer 2 protocol does not provide an explicit sequencing mechanism, you can configure L2TP to sequence its data packets according to the data channel sequencing mechanism described in the L2TPv3 IETF l2tpext working group draft.

A receiver of L2TP data packets mandates sequencing through the Sequencing Required AV pair when the session is being negotiated. A sender (or one that is manually configured to send sequenced packets) that receives this AV pair uses the Layer 2-specific pseudowire control encapsulation defined in L2TPv3.

You can configure L2TP to drop only out-of-order packets; you cannot configure L2TP to deliver the packets out-of-order. No reordering mechanism is available.

Interworking is not allowed when sequencing is enabled.

L2TPv3 Type of Service Marking

When Layer 2 traffic is tunneled across an IP network, information contained in the Type of Service (ToS) bits may be transferred to the L2TP-encapsulated IP packets in one of the following ways:

- If the tunneled Layer 2 frames themselves encapsulate IP packets, it may be desirable to simply copy the ToS bytes of the inner IP packets to the outer IP packet headers. This action is known as "ToS byte reflection."
- You can specify the ToS byte value used by all packets sent across the pseudowire. This is known as "Static ToS byte configuration".

For more details on how to configure ToS, see the Example Configuring a Negotiated L2TPv3 Session for Local HDLC Switching section.

Keepalive

The keepalive mechanism for L2TPv3 extends only to the endpoints of the tunneling protocol. L2TP has a reliable control message delivery mechanism that serves as the basis for the keepalive mechanism. The keepalive mechanism consists of an exchange of L2TP hello messages.

If a keepalive mechanism is required, the control plane is used, although it may not be used to bring up sessions. You can configure sessions manually.

In the case of static L2TPv3 sessions, a control channel between the two L2TP peers is negotiated through the exchange of start control channel request (SCCRQ), SCCRP, and SCCCN control messages. The control channel is responsible for maintaining only the keepalive mechanism through the exchange of hello messages.

The interval between hello messages is configurable per control channel. If one peer detects that the other peer has gone down through the keepalive mechanism, it sends a StopCCN control message and then notifies all the pseudowires to the peer about the event. This notification results in the teardown of both manually configured and dynamic sessions.

MTU Handling

It is important that you configure a Maximum Transmission Unit (MTU) appropriate for each L2TPv3 tunneled link. The configured MTU size ensures the following:

- The lengths of the tunneled Layer 2 frames fall below the MTU of the destination attachment circuit.
- The tunneled packets are not fragmented, which forces the receiving PE to reassemble them.

L2TPv3 handles the MTU as follows:

- The default behavior is to fragment packets that are larger than the session MTU.
- If you enable the **ip dfbit set** command in the pseudowire class, the default MTU behavior changes so that any packets that cannot fit within the tunnel MTU are dropped.
- If you enable the **ip pmtu** command in the pseudowire class, the L2TPv3 control channel participates in the path MTU (PMTU) discovery.

If you enable this feature, the following processing is performed:

- Internet Control Message Protocol (ICMP) unreachable messages sent back to the L2TPv3 device are deciphered and the tunnel MTU is updated accordingly. To receive ICMP unreachable messages for fragmentation errors, the Don't Fragment (DF) bit in the tunnel header is either set according to the DF bit value received from the CE device or set statically if the **ip dfbit set** option is enabled. The tunnel MTU is periodically reset to the default value based on a periodic timer.
- ICMP unreachable messages are sent back to the clients on the CE side. ICMP unreachable messages are sent to the CE whenever IP packets arrive on the CE-PE interface and have a packet size greater than the tunnel MTU. A Layer 2 header calculation is performed before the ICMP unreachable message is sent to the CE.

L2TPv3 Control Message Hashing

The L2TPv3 Control Message Hashing feature introduces a new and more secure authentication system that replaces the CHAP-like authentication system inherited from L2TPv2, which uses the Challenge and Challenge Response AV pairs in the SCCRQ, SCCRP, and SCCCN messages. The L2TPv3 Control Message Hashing feature incorporates an optional authentication or integrity check for all control messages.

The per-message authentication introduced by the L2TPv3 Control Message Hashing feature is designed to:

- Perform a mutual authentication between L2TP nodes.
- Check integrity of all control messages.
- Guard against control message spoofing and replay attacks that would otherwise be trivial to mount against the network.

The new authentication method uses the following:

- A computed, one-way hash over the header and body of the L2TP control message
- A preconfigured, shared secret that must be defined on the communicating L2TP nodes
- A local and remote random value exchanged using the Nonce AV pairs

Received control messages that lack any of the required security elements are dropped.

L2TPv3 control message integrity checking is a unidirectional mechanism that does not require the configuration of a shared secret. If integrity checking is enabled on the local PE device, control messages are sent with the message digest calculated without the shared secret or Nonce AV pairs and are verified by the remote PE device. If verification fails, the remote PE device drops the control message.

Enabling the L2TPv3 Control Message Hashing feature will impact performance during control channel and session establishment because additional digest calculation of the full message content is required for each sent and received control message. This is an expected trade-off for the additional security provided by this feature. In addition, network congestion may occur if the receive window size is too small. If the L2TPv3 Control Message Hashing feature is enabled, message digest validation must be enabled. Message digest validation deactivates the data path received sequence number update and restricts the minimum local receive window size to 35.

You may choose to configure control channel authentication or control message integrity checking. Control channel authentication requires participation by both peers and a shared secret must be configured on both devices. Control message integrity check is unidirectional and requires configuration on only one of the peers.

L2TPv3 Control Message Rate Limiting

The L2TPv3 Control Message Rate Limiting feature was introduced to counter the possibility of a denial-of-service (DoS) attack on a device running L2TPv3. The L2TPv3 Control Message Rate Limiting feature limits the rate at which SCCRQ control packets arriving at the PE that terminates the L2TPv3 tunnel can be processed. SCCRQ control packets initiate the process of bringing up the L2TPv3 tunnel and require a large amount of control plane resources of the PE device.

No configuration is required for the L2TPv3 Control Message Rate Limiting feature. This feature automatically runs in the background in supported releases.

L2TPv3 Digest Secret Graceful Switchover

Authentication of L2TPv3 control channel messages occurs using a password that is configured on all participating peer PE devices. Before the introduction of this feature, changing this password required removing of the old password from the configuration before adding the new password, causing an interruption in L2TPv3 services. The authentication password must be updated on all peer PE devices, which are often at different physical locations. It is difficult for all peer PE devices to be updated with the new password simultaneously to minimize interruptions in L2TPv3 services.

The L2TPv3 Digest Secret Graceful Switchover feature allows the password used to authenticate L2TPv3 control channel messages to be changed without tearing down the established L2TPv3 tunnels. This feature works only for authentication passwords configured with the L2TPv3 Control Message Hashing feature. Authentication passwords configured with the older, CHAP-like authentication system cannot be updated without tearing down L2TPv3 tunnels.

The L2TPv3 Digest Secret Graceful Switchover feature allows two control channel passwords to be configured simultaneously, so a new control channel password can be enabled without first removing the old password. Established tunnels are rapidly updated with the new password, but continue to use the old password until it is removed from the configuration. This allows authentication to continue normally with peer PE devices that have not yet been updated to use the new password. After all peer PE devices are configured with the new password, the old password can be removed from the configuration.

During the period when both a new and an old password are configured, authentication will occur only with the new password if the attempt to authenticate using the old password fails.

L2TPv3 Pseudowire

The pseudowire class configuration procedure creates a configuration template for the pseudowire. Use this template or class to configure session-level parameters for L2TPv3 sessions that are used to transport attachment circuit traffic over the pseudowire.

The pseudowire configuration specifies the characteristics of the L2TPv3 signaling mechanism, including the data encapsulation type, the control protocol, sequencing, Layer 3 fragmentation, payload-specific options, and IP properties. The setting that determines whether signaling is used to set up the pseudowire is also included.

If you specify the **encapsulation l2tpv3** command, you cannot remove it by using the **no encapsulation l2tpv3** command. You also cannot change the command setting by using the **encapsulation mpls** command. These methods result in the following error message:

```
Encapsulation changes are not allowed on an existing pw-class.
```

To remove the command, you must delete the pseudowire by using the **no pseudowire-class** command. To change the type of encapsulation, remove the pseudowire by using the **no pseudowire-class** command, reestablish the pseudowire, and specify the new encapsulation type.

Manual Clearing of L2TPv3 Tunnels

This feature lets you clear L2TPv3 tunnels manually. Before the introduction of this feature, there was no provision to clear a specific L2TPv3 tunnel manually. This functionality provides users more control over an L2TPv3 network.

L2TPv3 Tunnel Management

New and enhanced commands have been introduced to facilitate the management and diagnosis of problems with xconnect configurations. No specific configuration tasks are associated with these commands.

- **debug vpdn**--The output of this command includes authentication failure messages.
- **show l2tun session**--The **hostname** keyword allows the peer hostname to be displayed in the output.
- **show l2tun tunnel**--The **authentication** keyword allows the display of global information about L2TP control channel authentication AV pairs.
- **show xconnect**--The output of this command displays information about xconnect attachment circuits and pseudowires. This command also provides a sortable, single point of reference for information about all xconnect configurations.
- **xconnect logging pseudowire status**--This command enables syslog reporting of pseudowire status events.

For information about these Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the Cisco IOS Master Commands List, All Releases.

L2TPv3 Protocol Demultiplexing

The L2TPv3 Protocol Demultiplexing feature introduces the ability to provide native IPv6 support by utilizing a specialized IPv6 network to offload IPv6 traffic from the IPv4 network. The IPv6 traffic is tunneled to the IPv6 network transparently by using L2TPv3 pseudowires without affecting the configuration of the CE devices. IPv4 traffic is routed as usual within the IPv4 network, maintaining the existing performance and reliability of the IPv4 network.

The IPv4 PE devices must be configured to demultiplex the incoming IPv6 traffic from IPv4 traffic. The PE devices facing the IPv6 network do not require the IPv6 configuration. The configuration of the IPv6 network is beyond the scope of this document. For more information on configuring an IPv6 network, see the *IPv6 Configuration Guide*

L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations

The L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations feature lets you configure an Ethertype other than 0x8100 on Gigabit Ethernet interfaces with the QinQ or Dot1Q encapsulation. You can set the custom Ethertype to 0x9100, 0x9200, or 0x88A8. This allows interoperability in a multivendor Gigabit Ethernet environment.

HDLC over L2TPv3

HDLC for Layer 2 Data Encapsulation provides encapsulation of port-to-port Layer 2 traffic. All HDLC traffic including IPv4, IPv6, and non-IP packet, such as IS-IS, is tunneled over L2TPv3. HDLC does not support interworking mode.

Note L2TPv3 supports the IPv4 tunnel only for HDLC. The IPv4 tunnel supports IPv4 and IPv6 packets.

L2TPv3 Benefits

Simplifies Deployment of VPNs

L2TPv3 is an industry-standard Layer 2 tunneling protocol that ensures interoperability among vendors, thus increasing customer flexibility and service availability.

Omits the Need for MPLS

Service providers need not deploy Multiprotocol Label Switching (MPLS) in the core IP backbone to set up VPNs using L2TPv3 over the IP backbone, resulting in operational savings and increased revenue.

Supports Layer 2 Tunneling over IP for Any Payload

L2TPv3 provides enhancements to L2TP to support Layer 2 tunneling of any payload over an IP core network. L2TPv3 defines the base L2TP protocol as being separate from the Layer 2 payload that is tunneled.

Other Benefits

- Provides cookies for authentication
- Provides session state updates and multiple sessions
- Supports interworking (Ethernet-VLAN, Ethernet-QinQ, and VLAN-QinQ)

Supported L2TPv3 Payloads

Note Each L2TPv3 tunneled packet includes the entire Layer 2 frame of the payloads described in this section. If sequencing is required (see the Sequencing section), a Layer 2-specific sublayer (see the Pseudowire Control Encapsulation section) is included in the L2TPv3 header to provide the Sequence Number field.

Ethernet

An Ethernet frame arriving at a PE device is simply encapsulated in its entirety with an L2TP data header. At the other end, a received L2TP data packet is stripped of its L2TP data header. The payload, an Ethernet frame, is then forwarded to the appropriate attachment circuit.

Because the L2TPv3 tunneling protocol serves essentially as a bridge, it need not examine any part of an Ethernet frame. Any Ethernet frame received on an interface is tunneled, and any L2TP-tunneled Ethernet frame is forwarded out of the interface.

Note Because of the way in which L2TPv3 handles Ethernet frames, an Ethernet interface must be configured to promiscuous mode to capture all traffic received on the Ethernet segment attached to the device. All frames are tunneled through the L2TP pseudowire.

VLAN

L2TPv3 supports VLAN memberships in the following ways:

- Port-based, in which undated Ethernet frames are received
- VLAN-based, in which tagged Ethernet frames are received

In L2TPv3, Ethernet xconnect supports port-based VLAN membership and the reception of tagged Ethernet frames. A tagged Ethernet frame contains a tag header (defined in 802.1Q), which is 4 bytes long and consists of a 2-byte tag protocol identifier (TPID) field and a 2-byte tag control information (TCI) field. The TPID indicates that a TCI follows. The TCI is further broken down into the following three fields:

- User priority field
- Canonical format indicator (CFI)
- A 12-bit VLAN ID (VID)

For L2TPv3, an Ethernet subinterface configured to support VLAN switching may be bound to an xconnect service so that all Ethernet traffic, tagged with a VID specified on the subinterface, is tunneled to another PE. The VLAN Ethernet frames are forwarded in their entirety. The receiving PE may rewrite the VID of the tunneled traffic to another value before forwarding the traffic onto an attachment circuit.

To successfully rewrite VLANs, it may be necessary to disable the Spanning Tree Protocol (STP). This can be done on a per-VLAN basis by using the **no spanning-tree vlan** command.

Note Because of the way in which L2TPv3 handles VLAN packets, the Ethernet interface must be configured in promiscuous mode to capture all traffic received on the Ethernet segment attached to the device. All frames are tunneled through the L2TP pseudowire.

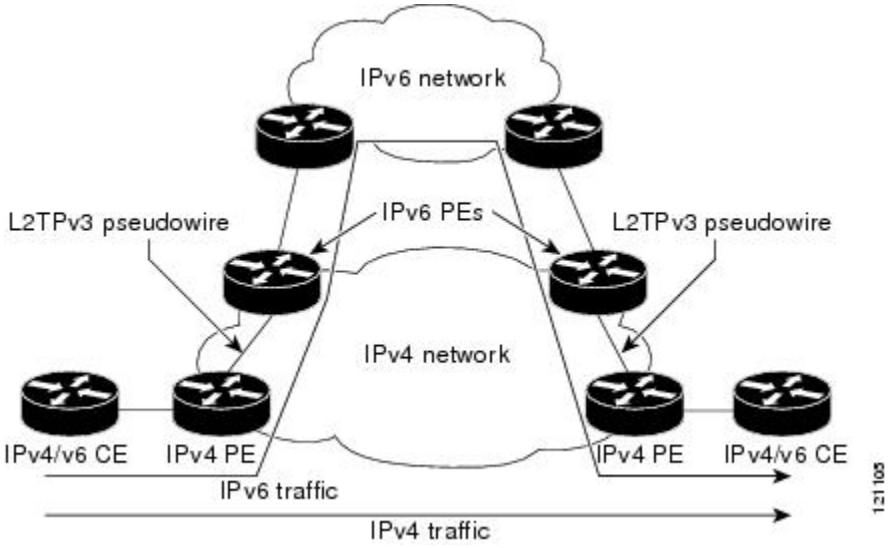
IPv6 Protocol Demultiplexing

Upgrading a service provider network to support IPv6 is a long and expensive process. As an interim solution, the Protocol Demultiplexing for L2TPv3 feature introduces the ability to provide native IPv6 support by setting up a specialized IPv6 network and offloading IPv6 traffic from the IPv4 network. IPv6 traffic is tunneled transparently to the IPv6 network using L2TPv3 pseudowires without affecting the configuration of the CE devices. IPv4 traffic is routed as usual within the IPv4 network, maintaining the existing performance and reliability of the IPv4 network.

The figure below shows a network deployment that offloads IPv6 traffic from the IPv4 network to a specialized IPv6 network. The PE devices demultiplex the IPv6 traffic from the IPv4 traffic. IPv6 traffic is routed to the

IPv6 network over an L2TPv3 pseudowire, while IPv4 traffic is routed normally. The IPv4 PE devices must be configured to demultiplex the incoming IPv6 traffic from the IPv4 traffic. The PE devices facing the IPv6 network do not require the IPv6 configuration.

Figure 5: Protocol Demultiplexing of IPv6 Traffic from IPv4 Traffic



If no IP address is configured, the protocol demultiplexing configuration is rejected. If an IP address is configured, the **xconnect** command configuration is rejected unless protocol demultiplexing is enabled in xconnect configuration mode before exiting that mode. If an IP address is configured with an **xconnect** command configuration and protocol demultiplexing is enabled, the IP address cannot be removed. To change or remove the configured IP address, the **xconnect** command configuration must first be disabled.

The table below shows the valid combinations of configurations.

Table 16: Valid Configuration Scenarios

Scenario	IP Address	xconnect Configuration	Protocol Demultiplexing Configuration
Routing	Yes	No	--
L2VPN	No	Yes	No
IPv6 Protocol Demultiplexing	Yes	Yes	Yes

Performance Impact of L2TPv3 on Cisco ASR 1000 Series Routers

L2TPv3 supports the following maximum number of attachment circuits and tunnels:

- First-generation Cisco ASR 1000 Series Route Processor (RP1) with Embedded Services Processor 10 (ESP10)
  - Attachment circuits for Ethernet: 8000 per system in a typical user environment. This includes 4000 per port and 8000 per SPA.
  - L2TPv3 tunnels: 1000 (in a typical user environment) and 2000 (maximum).

- Second-generation Cisco ASR 1000 Series Route Processor (RP2) with Embedded Services Processor 20 (ESP20)
  - Attachment circuits for Ethernet: 16,000 per system in a typical user environment. This includes 4000 per port and 8000 per SPA.
  - L2TPv3 tunnels: 2000 (in a typical user environment) and 4000 (maximum).

L2TPv3 adds tunnel encapsulation to TCP packets, which can cause fragmentation of big packets (packet size larger than the session MTU). Consider a scenario where a big TCP packet is followed by a small TCP packet (packet size smaller than the session MTU). After L2TPv3 encapsulation, the encapsulated big TCP packet will be fragmented, but the encapsulated small TCP packet will not be fragmented. On the Cisco ASR 1000 Series Routers, the fragmentation and reassembly of the big TCP packet requires an additional processor cycle. Because Cisco ASR 1000 Series Routers follow multithread processing, the small packet will need shorter processing time and may be forwarded ahead of the fragmented big packet. This process may result in packet sequence changes on the receiver's end.

As a workaround, you can enable the **ip pmtu** command to prevent the fragmentation of tunneled packets (see the MTU Handling section).

Layer 2 Protocol Tunneling and Forwarding

This feature introduces a new functionality for Layer 2 protocol tunneling on ISR platforms. Layer 2 protocol tunneling will tunnel more layer 2 protocols (mvrp/mmrp/elmi/link-oam/esmc/dtp) and forwards all 12 protocols (R4 R5 R6 R8 R9 RA RB RC RD RF stp vtp cdp pagp udld lacp dtp lldp ptppd mvrp mmrp elmi link-oam esmc).

How to Configure Layer 2 Tunneling Protocol Version 3

Configuring L2TP Control Channel Parameters

After you enter L2TP class configuration mode, you can configure L2TP control channel parameters in any order. If you have multiple authentication requirements, you can configure multiple sets of L2TP class control channel parameters with different L2TP class names. However, only one set of parameters can be applied to a connection between any pair of IP addresses.

Configuring L2TP Control Channel Timing Parameters

The following L2TP control channel timing parameters can be configured in L2TP class configuration mode:

- Packet size of the receive window used for the control channel
- Retransmission parameters used for control messages
- Timeout parameters used for the control channel

This task configures a set of timing control channel parameters in an L2TP class. All of the timing control channel parameter configurations are optional and may be configured in any order. If these parameters are not configured, default values are applied.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.
Step 4	**retransmit** {**initial retries** *initial-retries* | **retries** *retries* | **timeout** {**max** | **min**} *timeout*} **Example:** `Device(config-l2tp-class)# retransmit retries 10`	(Optional) Configures parameters that affect the retransmission of control packets. • **initial retries**—Specifies how many SCCRQs are resent before giving up on the session. Valid values for the *initial-retries* argument range from 1 to 1000. The default value is 2. • **retries**—Specifies how many retransmission cycles occur before determining that the peer PE device is not responding. Valid values for the *retries* argument range from 1 to 1000. The default value is 15. • **timeout** {**max** | **min**}—Specifies maximum and minimum retransmission intervals (in seconds) for resending control packets. Valid values for the *timeout* argument range from 1 to 8. The default maximum interval is 8. The default minimum interval is 1.
Step 5	**timeout setup** *seconds* **Example:** `Device(config-l2tp-class)# timeout setup 400`	(Optional) Configures the amount of time, in seconds, allowed to set up a control channel. • Valid values for the *seconds* argument range from 60 to 6000. The default value is 300.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-l2tp-class)# exit`	Exits L2TP class configuration mode.

Configuring L2TPv3 Control Channel Authentication Parameters

Configuring Authentication for the L2TP Control Channel

The L2TP control channel method of authentication is the older, CHAP-like authentication system inherited from L2TPv2.

The following L2TP control channel authentication parameters can be configured in L2TP class configuration mode:

- Authentication for the L2TP control channel
- Password used for L2TP control channel authentication
- Local hostname used for authenticating the control channel

This task configures a set of authentication control channel parameters in an L2TP class. All of the authentication control channel parameter configurations are optional and may be configured in any order. If these parameters are not configured, default values are applied.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.
Step 4	**authentication** **Example:** `Device(config-l2tp-class)# authentication`	(Optional) Enables authentication for the control channel between PE devices.

	Command or Action	Purpose

Step 5	**password** [**0** | **7**] *password*
	Example:
	`Device(config-l2tp-class)# password cisco`
		(Optional) Configures the password used for control channel authentication.
		• [**0** | **7**]—(Optional) Specifies the input format of the shared secret. The default value is **0**.
		  • **0**—Specifies that a plain-text secret is entered.
		  • **7**—Specifies that an encrypted secret is entered.
		• *password*—Defines the shared password between peer devices.

Step 6	**hostname** *name* **Example:** `Device(config-l2tp-class)# hostname yb2`	(Optional) Specifies a hostname used to identify the device during L2TP control channel authentication. • If you do not use this command, the default hostname of the device is used.
Step 7	**exit** **Example:** `Device(config-l2tp-class)# exit`	Exits L2TP class configuration mode.

Configuring L2TPv3 Control Message Hashing

This task configures L2TPv3 Control Message Hashing feature for an L2TP class.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.

	Command or Action	Purpose
Step 4	**digest** [**secret** [**0** | **7**] *password*] [**hash** {**md5** | **sha**}] **Example:** `Device(config-l2tp-class)# digest secret cisco hash sha`	(Optional) Enables L2TPv3 control channel authentication or integrity checking. • **secret**—(Optional) Enables L2TPv3 control channel authentication. **Note** If the **digest** command is issued • [**0** | **7**]—Specifies the input format of the shared secret. The default value is **0**. **0**—Specifies that a plain-text secret is entered. **7**—Specifies that an encrypted secret is entered. • *password*—Defines the shared secret between peer devices. The value entered for the *password* argument must be in the format that matches the input format specified by the [**0** | **7**] keyword option. • **hash** {**md5** | **sha**}—(Optional) Specifies the hash function to be used in per-message digest calculations. **md5**—Specifies HMAC-MD5 hashing. **sha**—Specifies HMAC-SHA-1 hashing. The default hash function is **md5**.
Step 5	**digest** check **Example:** `Device(config-l2tp-class)# digest check`	(Optional) Enables the validation of the message digest in received control messages. • Validation of the message digest is enabled by default. **Note** Validation of the message digest command. If authentication has be disabled to increase performa
Step 6	**hidden** **Example:** `Device(config-l2tp-class)# hidden`	(Optional) Enables AV pair hiding when sending control messages to an L2TPv3 peer. • AV pair hiding is disabled by default.

	Command or Action	Purpose
		• Only the hiding of the cookie AV pair is supported. • If a cookie is configured in L2TP class configuration mode (see the section *"Manually Configuring L2TPv3 Session Parameters"*), enabling AV pair hiding causes that cookie to be sent to the peer as a hidden AV pair using the password configured with the **digest secret** command. **Note** AV pair hiding is enabled on other authentication method
Step 7	**exit** **Example:** `Device(config-l2tp-class)# exit`	Exits L2TP class configuration mode.

Configuring L2TPv3 Digest Secret Graceful Switchover

Perform this task to make the transition from an old L2TPv3 control channel authentication password to a new L2TPv3 control channel authentication password without disrupting established L2TPv3 tunnels.

Before you begin

Before performing this task, you must enable control channel authentication as documented in the Configuring L2TPv3 Control Message Hashing task.

Note This task is not compatible with authentication passwords configured with the older, CHAP-like control channel authentication system.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**l2tp-class** *l2tp-class-name* **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode.
Step 4	**digest** [**secret** [**0** | **7**] *password*] [**hash** {**md5** | **sha**}] **Example:** `Device(config-l2tp-class)# digest secret cisco2 hash sha`	Configures a new password to be used in L2TPv3 control channel authentication. • A maximum of two passwords may be configured at any time. **Note** Authentication will now occur
Step 5	**end** **Example:** `Device(config-l2tp-class)# end`	Ends your configuration session by exiting to privileged EXEC mode.
Step 6	**show l2tun tunnel all** **Example:** `Device# show l2tun tunnel all`	(Optional) Displays the current state of Layer 2 tunnels and information about configured tunnels, including local and remote L2TP hostnames, aggregate packet counts, and control channel information. • Tunnels should be updated with the new control channel authentication password within a matter of seconds. If a tunnel does not update to show that two secrets are configured after several minutes have passed, the tunnel can be cleared manually and a defect report should be filed with the Cisco Technical Assistance Center (TAC). To clear an L2TPv3 tunnel manually, perform the task described in the sectionManually Clearing L2TPv3 Tunnels **Note** Issue this command to determi authentication. The output displ are configured.
Step 7	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 8	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.
Step 9	**no digest** [**secret** [**0** | **7**] *password* [**hash** {**md5** | **sha**}] **Example:** `Device(config-l2tp-class)# no digest secret cisco hash sha`	Removes the old password used in L2TPv3 control channel authentication. **Note** Do not remove the old pass
Step 10	**end** **Example:** `Device(config-l2tp-class)# end`	Ends your configuration session by exiting to privileged EXEC mode.
Step 11	**show l2tun tunnel all** **Example:** `Device# show l2tun tunnel all`	(Optional) Displays the current state of Layer 2 tunnels and information about configured tunnels, including local and remote L2TP hostnames, aggregate packet counts, and control channel information. • Tunnels should no longer be using the old control channel authentication password. If a tunnel does not update to show that only one secret is configured after several minutes have passed, that tunnel can be cleared manually and a defect report should be filed with TAC. To clear an L2TPv3 tunnel manually, perform the task described in the section Manually Clearing L2TPv3 Tunnels **Note** Issue this command to ensu authentication. The output d is configured.

Configuring L2TP Control Channel Maintenance Parameters

The L2TP hello packet keepalive interval control channel maintenance parameter can be configured in L2TP class configuration mode.

This task configures the interval used for hello messages in an L2TP class. This control channel parameter configuration is optional. If this parameter is not configured, the default value is applied.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.
Step 4	**hello** *interval* **Example:** `Device(config-l2tp-class)# hello 100`	(Optional) Specifies the exchange interval (in seconds) used between L2TP hello packets. • Valid values for the *interval* argument range from 0 to 1000. The default value is 60.
Step 5	**exit** **Example:** `Device(config-l2tp-class)# exit`	Exits L2TP class configuration mode.

Configuring the L2TPv3 Pseudowire

Perform this task to configure the L2TPv3 pseudowire.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class etherpw`	Enters pseudowire class configuration mode and optionally specifies the name of the L2TP pseudowire class.
Step 4	**encapsulation l2tpv3** **Example:** `Device(config-pw)# encapsulation l2tpv3`	Specifies that L2TPv3 is used as the data encapsulation method to tunnel IP traffic.
Step 5	**protocol** {**l2tpv3** | **none**} [*l2tp-class-name*] **Example:** `Device(config-pw)# protocol l2tpv3 class1`	(Optional) Specifies the L2TPv3 signaling protocol to be used to manage the pseudowires created with the control channel parameters in the specified L2TP class (see the section Configuring L2TP Control Channel Parameters). • If the *l2tp-class-name* argument is not specified, the default values for L2TP control channel parameters are used. The default **protocol** option is **l2tpv3**. • If you do not want to use signaling in the L2TPv3 sessions created with this pseudowire class, enter **protocol none**.
Step 6	**ip local interface** *interface-name* **Example:** `Device(config-pw)# ip local interface e0/0`	Specifies the PE device interface whose IP address is to be used as the source IP address for sending tunneled packets. • The same or a different local interface name can be used for each of the pseudowire classes configured between a pair of PE devices. **Note** This command must be confi method.
Step 7	**ip pmtu** **Example:** `Device(config-pw)# ip pmtu`	(Optional) Enables the discovery of the PMTU for tunneled traffic and helps fragmentation. • This command enables the processing of ICMP unreachable messages that indicate fragmentation errors in the backbone network that carries L2TPv3 session traffic. Also, this command enables MTU checking for IP packets sent into the session and that have the DF bit set. Any

	Command or Action	Purpose
		IP packet larger than the MTU is dropped and an ICMP unreachable message is sent. MTU discovery is disabled by default. **Note** The **ip pmtu** command is not s 5. • This command must be enabled in the pseudowire class configuration to enable fragmentation of IP packets before the data enters the pseudowire. **Note** To enable fragmentation of IP p also enter the **ip dfbit set** comm obtained more rapidly. **Note** When the **ip pmtu** command is If no IP header is found inside t
Step 8	**ip tos** {**value** *value* | **reflect**} **Example:** `Device(config-pw)# ip tos reflect`	(Optional) Configures the value of the ToS byte in IP headers of tunneled packets, or reflects the ToS byte value from the inner IP header. • Valid values for the *value* argument range from 0 to 255. The default ToS byte value is 0.
Step 9	**ip dfbit set** **Example:** `Device(config-pw)# ip dfbit set`	(Optional) Configures the value of the DF bit in the outer headers of tunneled packets. • Use this command if (for performance reasons) you do not want reassembly of tunneled packets on the peer PE device. • This command is disabled by default.
Step 10	**ip ttl** *value* **Example:** `Device(config-pw)# ip ttl 100`	(Optional) Configures the value of the time to live (TTL) byte in the IP headers of tunneled packets.

	Command or Action	Purpose
		• Valid values for the *value* argument range from 1 to 255. The default TTL byte value is 255.
Step 11	**ip protocol** {**l2tp** | *protocol-number*} **Example:** `Device(config-pw)# ip protocol l2tp`	(Optional) Configures the IP protocol to be used for tunneling packets.
Step 12	**sequencing** {**transmit** | **receive** | **both**} **Example:** `Device(config-pw)# sequencing both`	(Optional) Specifies the direction in which sequencing of data packets in a pseudowire is enabled: • **transmit**—Updates the Sequence Number field in the headers of data packets sent over the pseudowire according to the data encapsulation method that is used. • **receive**—Keeps the Sequence Number field in the headers of data packets received over the pseudowire. Out-of-order packets are dropped. • **both**—Enables both the **transmit** and **receive** options.
Step 13	**exit** **Example:** `Device(config-pw)# exit`	Exits pseudowire class configuration mode.

Configuring the Xconnect Attachment Circuit

The virtual circuit identifier that you configure creates the binding between a pseudowire configured on a PE device and an attachment circuit in a CE device. The virtual circuit identifier configured on the PE device at one end of the L2TPv3 control channel must also be configured on the peer PE device at the other end.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type slot / port* **Example:** `Device(config)# interface ethernet 0/0`	Specifies the interface by type (for example, Ethernet), slot, and port number, and enters interface configuration mode.
Step 4	**xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Device(config-if)# xconnect 10.0.3.201 123 pw-class vlan-xconnect`	Specifies the IP address of the peer PE device and the 32-bit virtual circuit identifier shared between the PE at each end of the control channel. • The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device. • At least one of the following pseudowire class parameters must be configured for the *pseudowire-parameters* argument: **encapsulation** {**l2tpv3** [**manual**] | **mpls**}—Specifies the tunneling method used to encapsulate data in the pseudowire: **l2tpv3**—L2TPv3 is the tunneling method to be used. **manual**—(Optional) No signaling is to be used in the L2TPv3 control channel. This command places the device in xconnect configuration mode for the manual configuration of L2TPv3 parameters for the attachment circuit. **mpls**—MPLS is the tunneling method to be used. • **pw-class** {*pw-class-name*}—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. • The optional **encapsulation** parameter specifies the method of pseudowire tunneling used: L2TPv3 or MPLS. Enter **manual** if you do not want signaling to be used in the L2TPv3 control channel. The **encapsulation l2tpv3 manual** keyword combination enters xconnect configuration submode. See the section "*Manually Configuring L2TPv3 Session Parameters*" for the other L2TPv3 commands that you must enter to complete the configuration of the L2TPv3 control channel. If you do not enter an **encapsulation** value, the

	Command or Action	Purpose
		encapsulation method entered with the **password** command in the Configuring the Xconnect Attachment Circuit task is used. • The optional **pw-class** parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. Specify the pseudowire-class option if you need to configure more advanced options. **Note** You must configure either th **Note** If you select L2TPv3 as your • The optional **sequencing** parameter specifies whether sequencing is required for packets that are received, sent, or both received and sent.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configure L2TPv3 on a Switched Virtual Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**pseudowire-class** *pw-class-name* **Example:** `Device(config)# pseudowire-class pc`	Enters pseudowire class configuration mode and specifies the name of the L2TP pseudowire class.
Step 4	**encapsulation l2tpv3** **Example:** `Device(config-pw)# encapsulation l2tpv3`	Specifies L2TPv3, which is used as the data encapsulation method to tunnel IP traffic.
Step 5	**protocol** {**l2tpv3** | **none**} *l2tp-class-name* **Example:** `Device(config-pw)# protocol l2tpv3 class1 pc`	(Optional) Specifies the L2TPv3 signaling protocol to be used to manage the pseudowires created with the control channel parameters in the specified L2TP class (see the section Configuring L2TP Control Channel Parameters). If the l2tp-class-name argument is not specified, the default values for L2TP control channel parameters are used. The default **protocol** option is **l2tpv3**. If you do not want to use signaling in the L2TPv3 sessions createdwith this pseudowire class, enter **protocol none**.
Step 6	**xconnect** *ip address vc-id* **encapsulation l2tpv3 pw-class** *pw-class-name* **Example:** `Device(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 pw-class pc`	Specifies the IP address of the peer PE device and the 32-bit virtual circuit identifier shared between the PE at each end of the control channel, and enters xconnect configuration mode. • The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device. • The **encapsulation l2tpv3** parameter specifies that L2TPv3 is to be used as the pseudowire tunneling method. • The mandatory **pw-class** and *pw-class-name* keyword and argument combination specifies the pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken.
Step 7	**ip local interface** *interface-name* **loopback** **Example:** `Device(config-pw)# ip local interface ge0/0/0 loopback0`	Creates a loopback interface and enters interface configuration mode. Specifies the PE device interface whose IP address is to be used as the source IP address for sending tunneled packets. The same or a different local interface name can be used for each ofthe pseudowire classes configured between a pair of PE devices.

	Command or Action	Purpose
		Note This command must be config method.
Step 8	**ip address** *ip address* **Example:** `Device(config-pw)# ip address 10.1.0.1 255.255.255.255`	Assigns an IP address to the interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Manually Configuring L2TPv3 Session Parameters

When you bind an attachment circuit to an L2TPv3 pseudowire for the xconnect service by using the **xconnect l2tpv3 manual** command (see the section "Configuring the Xconnect Attachment Circuit") because you do not want signaling, you must configure L2TP-specific parameters to complete the L2TPv3 control channel configuration.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** `Device(config)# interface ethernet 0/0`	Specifies the interface by type (for example, Ethernet), slot, and port number, and enters interface configuration mode.
Step 4	**xconnect** *peer-ip-address vc-id* **encapsulation l2tpv3 manual pw-class** *pw-class-name* **Example:** `Device(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class vlan-xconnect`	Specifies the IP address of the peer PE device and the 32-bit virtual circuit identifier shared between the PE at each end of the control channel, and enters xconnect configuration mode. • The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device.

	Command or Action	Purpose
		• The **encapsulation l2tpv3 manual** parameter specifies that L2TPv3 is to be used as the pseudowire tunneling method. • The mandatory **pw-class** *pw-class-name* keyword and argument combination specifies the pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken.
Step 5	**l2tp id** *local-session-id remote-session-id* **Example:** `Device(config-if-xconn)# l2tp id 222 111`	Configures the identifiers for the local L2TPv3 session and for the remote L2TPv3 session on the peer PE device. • This command is required to complete the attachment circuit configuration and a static L2TPv3 session configuration.
Step 6	**l2tp cookie local** *size low-value* [*high-value*] **Example:** `Device(config-if-xconn)# l2tp cookie local 4 54321`	(Optional) Specifies the value that the peer PE must include in the cookie field of incoming (received) L2TP packets. • The size of the cookie field can be 4 or 8 bytes. If you do not enter this command, no cookie value is included in the header of L2TP packets. • If you configure the cookie length in incoming packets as 8 bytes, you must specify a 4-byte high value and a 4-byte low value.
Step 7	**l2tp cookie remote** *size low-value* [*high-value*] **Example:** `Device(config-if-xconn)# l2tp cookie remote 4 12345`	(Optional) Specifies the value that the device includes in the cookie field of outgoing (sent) L2TP packets. • The size of the cookie field can be 4 or 8 bytes. If you do not enter this command, no cookie value is included in the header of L2TP packets. • If you configure the cookie length in outgoing packets as 8 bytes, you must specify a 4-byte high value and a 4-byte low value.
Step 8	**l2tp hello** *l2tp-class-name* **Example:** `Device(config-if-xconn)# l2tp hello l2tp-defaults`	(Optional) Specifies the L2TP class name to be used (see the section Configuring L2TP Control Channel Parameters) for control channel configuration parameters, including

	Command or Action	Purpose
		the interval to use between hello keepalive messages.  **Note** This command assumes tha control channel is to be use default, no hello messages a
Step 9	**exit** **Example:** `Device(config-if-xconn)# exit`	Exits xconnect configuration mode.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configuring Protocol Demultiplexing for L2TPv3

Configuring Protocol Demultiplexing for Ethernet Interfaces

Perform this task to configure the Protocol Demultiplexing feature on an Ethernet interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** `Device(config)# interface ethernet 0/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.16.128.4`	Sets a primary or secondary IP address for an interface.
Step 5	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*	Specifies the IP address of the peer PE device and the 32-bit VCI shared between the PE at

	Command or Action	Purpose
	Example: `Device(config-if)# xconnect 10.0.3.201 888 pw-class demux`	each end of the control channel, and enters xconnect configuration mode. • The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device. • **pw-class** *pw-class-name*—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class** parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. **Note** The L2TPv3 session can also be *Session Parameters*" for informa
Step 6	**match protocol ipv6** **Example:** `Device(config-if-xconn)# match protocol ipv6`	Enables protocol demultiplexing of IPv6 traffic.
Step 7	**exit** **Example:** `Device(config-if-xconn)# exit`	Exits xconnect configuration mode.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configuring an L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations

The L2TPv3 Custom Ethertype for dot1q and QinQ Encapsulations feature lets you configure an Ethertype other than 0x8100 on Gigabit Ethernet interfaces with QinQ or dot1Q encapsulations. You can set the custom Ethertype to 0x9100, 0x9200, or 0x88A8. To define the Ethertype field type, you use the **dot1q tunneling ethertype** command.

Perform this task to set a custom Ethertype.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**dot1q tunneling ethertype** {**0x88A8** | **0x9100** | **0x9200**} **Example:** `Device(config-if)# dot1q tunneling ethertype 0x9100`	Defines the Ethertype field type used by peer devices when implementing Q-in-Q VLAN tagging.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configuring GEC over L2TPv3

Perform this task to configure Gigabit EtherChannel (GEC) over Layer 2 Tunneling Protocol Version 3 (L2TPv3).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface Loopback0** **Example:** `Device(config)# interface Loopback0`	Creates a loopback interface and enters interface configuration mode.
Step 4	**ip address** *ip-address* **Example:**	Assigns an IP address to the interface.

	Command or Action	Purpose
	`Device(config-if)# ip address 10.1.0.1 255.255.255.255`	
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class l2tpv3`	Enters pseudowire class configuration mode and optionally specifies the name of the Layer 2 Tunneling Protocol (L2TP) pseudowire class.
Step 7	**encapsulation l2tpv3** **Example:** `Device(config-pw)# encapsulation l2tpv3`	Specifies that L2TPv3 is used as the data encapsulation method to tunnel IP traffic.
Step 8	**ip local interface** *interface-name* **Example:** `Device(config-pw)# ip local interface loopback0`	Specifies the provider edge (PE) interface whose IP address is to be used as the source IP address for sending tunneled packets. • Use the same local interface name for all pseudowire classes that are configured between a pair of PE devices. **Note** This command must be configur method.
Step 9	**exit** **Example:** `Device(config-pw)# exit`	Exits pseudowire class configuration mode and enters global configuration mode.
Step 10	**interface port-channel** *channel-number* **Example:** `Device# interface port-channel 1`	Defines a port channel and enters interface configuration mode.
Step 11	**xconnect** *peer-ip-address* **encapsulation l2tpv3 pw-class** *pw-class-name* **Example:** `Device(config-subif)# xconnect 10.0.3.201 1234 encapsulation l2tpv3 pw-class l2tpv3`	Specifies the IP address of the peer PE device and the 32-bit virtual circuit identifier (VCI) shared between the PE at each end of the control channel. • The combination of the peer device ID and the VCI must be unique. • **pw-class** *pw-class-name*—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class**

	Command or Action	Purpose
		parameter binds **xconnect** to a specific pseudowire class. The pseudowire class then serves as a template for all attachment circuits bound to it.
Step 12	**exit** **Example:** `Device(config-subif)# exit`	Exits subinterface configuration mode and enters global configuration mode.
Step 13	**interface gigabitethernet** *interface-type-number* **Example:** `Device(config)# interface gigabitEthernet 0/0/0`	Enters interface configuration mode.
Step 14	**channel-group** *channel-group-number* **Example:** `Device(config-if)# channel-group 1`	Add the interface to an EtherChannel group.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring GEC with Dot1Q

Perform this task to configure Gigabit EtherChannel (GEC) with VLAN over Layer 2 Tunneling Protocol Version 3 (L2TPv3).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface port-channel** *interface-number* **Example:** `Device(config)# interface port-channel 1.1`	Defines a port channel and enters subinterface configuration mode.

	Command or Action	Purpose
Step 4	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100`	Specifies that dot1q is used as the data encapsulation method to tunnel IP traffic.
Step 5	**xconnect** *peer-ip-address* **encapsulation l2tpv3 pw-class** *pw-class-name* **Example:** `Device(config-subif)# xconnect 10.0.3.201 1234 encapsulation l2tpv3 pw-class l2tpv3`	Specifies the IP address of the peer provider edge (PE) device and the 32-bit virtual circuit identifier (VCI) that is shared between the PE device at each end of the control channel. • The combination of the peer device ID and the VCI must be unique. • **pw-class** *pw-class-name*—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class** parameter binds **xconnect** to a specific pseudowire class. The pseudowire class then serves as a template for all attachment circuits bound to it.
Step 6	**end** **Example:** `Device# end`	Exits subinterface configuration mode and returns to privileged EXEC mode .

Configuring GEC with QinQ

Perform this task to configure Gigabit EtherChannel (GEC) with queue-in-queue (QinQ) over Layer 2 Tunneling Protocol Version 3 (L2TPv3).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface port-channel** *interface-number* **Example:** `Device(config)# interface port-channel 1.1`	Defines the subinterface as a port channel and enters subinterface configuration mode.

	Command or Action	Purpose
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** *second-vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100 second-dot1q 200`	Specifies that QinQ is used as the data encapsulation method to tunnel IP traffic.
Step 5	**xconnect** *peer-ip-address* **encapsulation l2tpv3 pw-class** *pw-class-name* **Example:** `Device(config-subif)# xconnect 10.0.3.202 1234 encapsulation l2tpv3 pw-class l2tpv3`	Specifies the IP address of the peer provider edge (PE) device and the 32-bit virtual circuit identifier (VCI) that is shared between the PE device at each end of the control channel. • The combination of the peer device ID and the VCI must be unique. • **pw-class** *pw-class-name*—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class** parameter binds **xconnect** to a specific pseudowire class. The pseudowire class then serves as a template for all attachment circuits bound to it.
Step 6	**end** **Example:** `Device# end`	Exits subinterface configuration mode and returns to privileged EXEC mode.

Manually Clearing L2TPv3 Tunnels

Perform this task to manually clear a specific L2TPv3 tunnel and all the sessions in that tunnel.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear l2tun** {**l2tp-class** *l2tp-class-name* | **tunnel id** *tunnel-id* | **local ip** *ip-address* | **remote ip** *ip-address* | **all**} **Example:** `Device# clear l2tun tunnel id 56789`	Clears the specified L2TPv3 tunnel. (This command is not available if there are no L2TPv3 tunnel sessions configured.) • **l2tp-class** *l2tp-class-name*—All L2TPv3 tunnels with the specified L2TP class name are torn down. • **tunnel id** *tunnel-id*—The L2TPv3 tunnel with the specified tunnel ID are torn down.

	Command or Action	Purpose
		• **local ip** *ip-address*—All L2TPv3 tunnels with the specified local IP address are torn down. • **remote ip** *ip-address*—All L2TPv3 tunnels with the specified remote IP address are torn down. • **all**—All L2TPv3 tunnels are torn down.

Configuration Examples for Layer 2 Tunneling Protocol Version 3

Note The IP addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Example: Configuring an L2TPv3 Session for an Xconnect Ethernet Interface

L2TPv3 is the only encapsulation method that supports a manually provisioned session setup. This example shows how to configure a static session configuration in which all control channel parameters are set up in advance. There is no control plane used and no negotiation phase to set up the control channel. The PE device starts sending tunneled traffic as soon as the Ethernet interface (int e0/0) comes up. The virtual circuit identifier, 123, is not used. The PE sends L2TP data packets with session ID 111 and cookie 12345. In turn, the PE expects to receive L2TP data packets with session ID 222 and cookie 54321.

```
l2tp-class l2tp-defaults
 retransmit initial retries 30
 cookie-size 8
pseudowire-class ether-pw
 encapsulation l2tpv3
 protocol none
 ip local interface Loopback0
interface Ethernet 0/0
 xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class ether-pw
 l2tp id 222 111
 l2tp cookie local 4 54321
 l2tp cookie remote 4 12345
 l2tp hello l2tp-defaults
```

Example: Configuring a Negotiated L2TPv3 Session for an Xconnect VLAN Subinterface

The following is a sample configuration of a dynamic L2TPv3 session for a VLAN xconnect interface. In this example, only VLAN traffic with a VLAN ID of 5 is tunneled. In the other direction, the L2TPv3 session identified by a virtual circuit identifier of 123 receives forwarded frames whose VLAN ID fields are rewritten to contain the value 5. L2TPv3 is used as both the control plane protocol and the data encapsulation.

```
l2tp-class class1
 authentication
 password secret
pseudowire-class vlan-xconnect
 encapsulation l2tpv3
 protocol l2tpv3 class1
 ip local interface Loopback0
interface Ethernet0/0.1
 encapsulation dot1q 5
 xconnect 10.0.3.201 123 pw-class vlan-xconnect
```

Example: Configure a Static L2TPv3 Session for a SVI

Configure an SVI for various components of an L2TPv3 session:

```
pseudowire-class pc
 encapsulation l2tpv3
 ip local interface Loopback0
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
interface GigabitEthernet0/0/0
 ip address 12.0.0.1 255.255.255.252
interface GigabitEthernet0/1/0
 switchport access vlan 30
 switchport mode access
interface Vlan30
 xconnect 2.2.2.2 4294967295 encapsulation l2tpv3 pw-class pc
```

Example: Configuring a Negotiated L2TPv3 Session for Local HDLC Switching

The following is a sample configuration of a dynamic L2TPv3 session for local HDLC switching. In this example, note that it is necessary to configure two different IP addresses at the endpoints of the L2TPv3 pseudowire because the virtual circuit identifier must be unique for a given IP address.

```
interface loopback 1
 ip address 10.0.0.1 255.255.255.255
interface loopback 2
 ip address 10.0.0.2 255.255.255.255
pseudowire-class loopback1
 encapsulation l2tpv3
 ip local interface loopback1
pseudowire-class loopback2
 encapsulation l2tpv3
 ip local interface loopback2
interface s0/0
 encapsulation hdlc
 xconnect 10.0.0.1 100 pw-class loopback2
interface s0/1
 encapsulation hdlc
 xconnect 10.0.0.2 100 pw-class loopback1
```

Example: Verifying an L2TPv3 Session

To display information about current L2TPv3 sessions on a device, use the **show l2tun session brief** command.

```
Device# show l2tun session brief
L2TP Session Information Total tunnels 1 sessions 1
LocID      TunID      Peer-address    State     Username, Intf/
                                      sess/cir  Vcid, Circuit
2391726297 2382731778 6.6.6.6         est,UP    100, Gi0/2/0
```

To display detailed information about current L2TPv3 sessions on a device, use the **show l2tun session all** command.

```
Device#show l2tun session all
L2TP Session Information Total tunnels 1 sessions 1
Session id 2391726297 is up, logical session id 36272, tunnel id 2382731778
  Remote session id is 193836624, remote tunnel id 2280318174
  Locally initiated session
  Unique ID is 12
Session Layer 2 circuit, type is Ethernet, name is GigabitEthernet0/2/0
  Session vcid is 100
  Circuit state is UP
    Local circuit state is UP
    Remote circuit state is UP
Call serial number is 98300002
Remote tunnel name is l2tp-asr-2
  Internet address is 6.6.6.6
Local tunnel name is l2tp-asr-1
  Internet address is 3.3.3.3
IP protocol 115
  Session is L2TP signaled
  Session state is established, time since change 00:05:25
    94 Packets sent, 58 received
    9690 Bytes sent, 5642 received
  Last clearing of counters never
  Counters, ignoring last clear:
    94 Packets sent, 58 received
    9690 Bytes sent, 5642 received
    Receive packets dropped:
      out-of-order:             0
      other:                    0
      total:                    0
Send packets dropped:
      exceeded session MTU:     0
      other:                    0
      total:                    0
  DF bit off, ToS reflect disabled, ToS value 0, TTL value 255
  Sending UDP checksums are disabled
  Received UDP checksums are verified
  No session cookie information available
  FS cached header information:
    encap size = 24 bytes
    45000014 00000000 ff73a965 03030303
    06060606 0b8db650
  Sequencing is off
  Conditional debugging is disabled
  SSM switch id is 4101, SSM segment id is 12294
```

Example: Verify a Static L2TPv3 Session for a Switched Virtual Interface

```
show xconnect interface Vlan30 detail
Legend:       XC ST=Xconnect State   S1=Segment1 State    S2=Segment2 State
UP=Up         DN=Down                AD=Admin Down        IA=Inactive
SB=Standby    HS=Hot Standby         RV=Recovering        NH=No Hardware

XC ST     Segment 1                     S1 Segment 2                          S2
------+-------------------------------+--+--------------------------------+--
UP pri   ac Vl30:30(Eth VLAN)            UP l2tp 2.2.2.2:4294967295           UP
         Interworking: vlan                   Session ID: 2947605650
                                              Tunnel ID: 3954331565
                                              Peer name: Clarinet-4451
                                              Protocol State: UP
                                              Remote Circuit State: UP
                                              pw-class: pc
```

Example: Verifying an L2TP Control Channel

The L2TP control channel is used to negotiate capabilities, monitor the health of the peer PE device, and set up various components of an L2TPv3 session.

To display information about L2TP control channels to other L2TP-enabled devices for all L2TP sessions on the device, use the **show l2tun tunnel** command.

```
Device# show l2tun tunnel
L2TP Tunnel Information Total tunnels 1 sessions 1
LocTunID   RemTunID   Remote Name   State  Remote Address  Sessn L2TP Class/
                                                          Count VPDN Group
2382731778 2280318174 l2tp-asr-2    est    6.6.6.6         1     l2tp_default_cl
```

To display detailed information about L2TP control channels to other L2TP-enabled devices for all L2TP sessions on the device, use the **show l2tun tunnel all** command.

```
Device# show l2tun tunnel all
L2TP Tunnel Information Total tunnels 1 sessions 1
Tunnel id 2382731778 is up, remote id is 2280318174, 1 active sessions
  Locally initiated tunnel
  Tunnel state is established, time since change 00:02:59
  Tunnel transport is IP  (115)
  Remote tunnel name is l2tp-asr-2
    Internet Address 6.6.6.6, port 0
  Local tunnel name is l2tp-asr-1
    Internet Address 3.3.3.3, port 0
  L2TP class for tunnel is l2tp_default_class
  Counters, taking last clear into account:
    54 packets sent, 35 received
    5676 bytes sent, 3442 received
  Last clearing of counters never
  Counters, ignoring last clear:
    54 packets sent, 35 received
    5676 bytes sent, 3442 received
Control Ns 5, Nr 4
  Local RWS 1024 (default), Remote RWS 1024
  Control channel Congestion Control is disabled
  Tunnel PMTU checking disabled
  Retransmission time 1, max 1 seconds
  Unsent queuesize 0, max 0
  Resend queuesize 0, max 2
  Total resends 0, ZLB ACKs sent 2
  Total out-of-order dropped pkts 0
  Total out-of-order reorder pkts 0
  Total peer authentication failures 0
  Current no session pak queue check 0 of 5
  Retransmit time distribution: 0 0 0 0 0 0 0 0 0 0
  Control message authentication is disabled
```

Example: Configuring L2TPv3 Control Channel Authentication

The following example shows how to configure CHAP-style authentication of the L2TPv3 control channel:

```
l2tp-class class0
 authentication
 password cisco
```

The following example shows how to configure control channel authentication using the L2TPv3 Control Message Hashing feature:

```
l2tp-class class1
 digest secret cisco hash sha
 hidden
```

The following example shows how to configure control channel integrity checking and how to disable validation of the message digest using the L2TPv3 Control Message Hashing feature:

```
l2tp-class class2
 digest hash sha
 no digest check
```

The following example shows how to disable the validation of the message digest using the L2TPv3 Control Message Hashing feature:

```
l2tp-class class3
 no digest check
```

Example: Configuring L2TPv3 Digest Secret Graceful Switchover

The following example shows how to use the L2TPv3 Digest Secret Graceful Switchover feature to change the L2TP control channel authentication password for the L2TP class named class1. This example assumes that you already have an old password configured for the L2TP class named class1.

```
Device(config)#l2tp-class class1
Device(config-l2tp-class)#digest secret cisco2 hash sha
!
! Verify that all peer PE devices have been updated to use the new password before
! removing the old password.
!
Device(config-l2tp-class)#no digest secret cisco hash sha
```

Example: Verifying L2TPv3 Digest Secret Graceful Switchover

The following **show l2tun tunnel all** command output shows information about the L2TPv3 Digest Secret Graceful Switchover feature:

```
Device#show l2tun tunnel all
! The output below displays control channel password information for a tunnel which has
! been updated with the new control channel authentication password.
!
Tunnel id 12345 is up, remote id is 54321, 1 active sessions
Control message authentication is on, 2 secrets configured
Last message authenticated with first digest secret
!
! The output below displays control channel password information for a tunnel which has
! only a single control channel authentication password configured.
!
Tunnel id 23456 is up, remote id is 65432, 1 active sessions
!
Control message authentication is on, 1 secrets configured
Last message authenticated with first digest secret
!
! The output below displays control channel password information for a tunnel which is
! communicating with a peer that has only the new control channel authentication password
! configured.
!
Tunnel id 56789 is up, remote id is 98765, 1 active sessions
!
Control message authentication is on, 2 secrets configured
Last message authenticated with second digest secret
```

Example: Configuring a Pseudowire Class for Fragmentation of IP Packets

The following is a sample configuration of a pseudowire class that will allow IP traffic generated from the CE device to be fragmented before entering the pseudowire:

```
pseudowire class class1
 encapsulation l2tpv3
 ip local interface Loopback0
 ip pmtu
 ip dfbit set
```

Example: Configuring Protocol Demultiplexing for L2TPv3

The following example shows how to configure the L2TPv3 Protocol Demultiplexing feature on IPv4 PE devices. The PE devices facing the IPv6 network do not require the IPv6 configuration.

```
interface ethernet 0/1
 ip address 172.16.128.4
 xconnect 10.0.3.201 888 pw-class demux
  match protocol ipv6
```

Example: Manually Clearing an L2TPv3 Tunnel

The following example demonstrates how to manually clear a specific L2TPv3 tunnel using the tunnel ID:

```
clear l2tun tunnel 65432
```

Example: Configuring an L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations

The following example shows how to configure an Ethertype other than 0x8100 on Gigabit Ethernet interfaces with QinQ or dot1Q encapsulations. In this example, the Ethertype field is set to 0x9100 on Gigabit Ethernet interface 1/0/0.

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/0
Device(config-if)# dot1q tunneling ethertype 0x9100
```

Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport

Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport on Dynamic Mode

The following example shows how to configure xconnect on a serial interface with HDLC encapsulation on a dynamic mode. The dynamic mode uses L2TPv3 signaling in control channel to set up the L2TPv3 tunnel.

```
pseudowire-class 774
 encapsulation l2tpv3
 protocol l2tpv3
 ip local interface GigabitEthernet0/0/1.774
!
interface Serial0/2/0:0
 no ip address
 xconnect 4.4.4.4 200 pw-class 774
```

Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport on Static Mode

The following example shows how to configure xconnect on a serial interface with HDLC encapsulation on a static mode. The static mode is used to disable signaling in the L2TPv3 control channel. Since signaling is disabled, you must specify the manual option in xconnect and configure L2TP-specific parameters to complete the L2TPv3 control channel configuration.

```
pseudowire-class pe1-ether-pw
 encapsulation l2tpv3
 protocol none
```

```
 ip local interface Loopback1
!
interface Serial0/2/0:0
 no ip address
 xconnect 2.2.2.2 50 encapsulation l2tpv3 manual pw-class pe1-ether-pw
  l2tp id 111 111
  l2tp cookie local 4 54321
  l2tp cookie remote 4 12345
```

Example: Configuring GEC over L2TPv3

The following is a sample configuration of Gigabit EtherChannel (GEC) over Layer 2 Tunneling Protocol Version 3 (L2TPv3):

```
Device#enable
Device# configure terminal
Device(config)# interface Loopback0
Device(config-if)# ip address 10.1.0.1 255.255.255.255
Device(config-if)# exit
Device(config)# pseudowire-class l2tpv3
Device(config-pw)# encapsulation l2tpv3
Device(config-pw)# ip local interface loopback0
Device(config-if)# exit
Device(config)# interface port-channel 1
Device(config-if)# xconnect 1.1.1.1 1234 encapsulation l2tpv3 pw-class l2tpv3
Device(config-if)# exit
Device(config)# interface g0/0/0
Device(config-if)# channel-group 1
Device(config-if)# end
```

Example: Configuring GEC with Dot1q over L2TPv3

The following is a sample configuration of a Gigabit EtherChannel (GEC) with dot1q over Layer 2 Tunneling Protocol Version 3 (L2TPv3):

```
Device#enable
Device#configure terminal
Device(config)#interface port-channel 1
Device(config-if)#interface port-channel 1.1
Device(config-subif)#encapsulation dot1q 100
Device(config-subif)#xconnect 10.0.0.2 1234 encapsulation l2tpv3 pw-class l2tpv3
Device(config-subif)#end
```

Example: Configuring GEC with QinQ over L2TPv3

The following is a sample configuration of a Gigabit EtherChannel (GEC) with queue-in-queue (QinQ) over Layer 2 Tunneling Protocol Version 3 (L2TPv3):

```
Device#enable
Device#configure terminal
Device(config)#interface port-channel 1
Device(config-if)#interface port-channel 1.1
Device(config-subif)#encapsulation dot1q 100 second-dot1q 200
Device(config-subif)#xconnect 10.0.0.3 1234 encapsulation l2tpv3 pw-class l2tpv3
Device(config-subif)#end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
WAN commands: complete command syntax, command mode, defaults, usage guidelines and examples	Wide-Area Networking Command Reference
Layer 2 Tunnel Protocol Version 3	*Layer 2 Tunneling Protocol Version 3*
Any Transport over MPLS	*Any Transport over MPLS*
Cisco 12000 series routers hardware support	*Cross-Platform Release Notes for Cisco IOS Release 12.0S*
Cisco 7600 series routers hardware support	*Cross-Platform Release Notes for Cisco IOS Release 12.2SR*
Cisco 3270 series routers hardware support	*Release Notes for Cisco IOS Software Release 12.2SE*

Standards and RFCs

Standard/RFC	Title
draft-ietf-l2tpext-l2tp-base-03.txt	*Layer Two Tunneling Protocol (Version 3) 'L2TPv3'*
draft-martini-l2circuit-trans-mpls-09.txt	*Transport of Layer 2 Frames Over MPLS*
draft-ietf-pwe3-frame-relay-03.txt.	*Encapsulation Methods for Transport of Frame Relay over MPLS Networks*
draft-martini-l2circuit-encap-mpls-04.txt.	*Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*
draft-ietf-pwe3-ethernet-encap-08.txt.	*Encapsulation Methods for Transport of Ethernet over MPLS Networks*
draft-ietf-pwe3-hdlc-ppp-encap-mpls-03.txt.	*Encapsulation Methods for Transport of PPP/HDLC over MPLS Networks*
draft-ietf-ppvpn-l2vpn-00.txt.	*An Architecture for L2VPNs*

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: *http://www.cisco.com/go/mibs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Glossary

AV pairs—Attribute-value pairs.

CEF—Cisco Express Forwarding. The Layer 3 IP switching technology that optimizes network performance and scalability for networks with large and dynamic traffic patterns.

data-link control layer—Layer 2 in the SNA architectural model. Responsible for the transmission of data over a particular physical link. Corresponds approximately to the data link layer of the OSI model.

DCE—Data circuit-terminating equipment (ITU-T expansion). Devices and connections of a communications network that comprise the network end of the user-to-network interface.

DF bit—Don't Fragment bit. The bit in the IP header that can be set to indicate that the packet should not be fragmented.

DTE—Data terminal equipment. The device at the user end of a user-network interface that serves as a data source, destination, or both.

HDLC—High-Level Data Link Control. A generic link-level communications protocol developed by the ISO. HDLC manages synchronous, code-transparent, serial information transfer over a link connection.

ICMP—Internet Control Message Protocol. A network protocol that handles network errors and error messages.

IDB— Interface descriptor block.

IS-IS—Intermediate System-to-Intermediate System. The OSI link-state hierarchical routing protocol based on DECnet Phase V routing, whereby ISs (devices) exchange routing information based on a single metric to determine network topology.

L2TP—An extension to PPP that merges features of two tunneling protocols: Layer 2 Forwarding (L2F) from Cisco Systems and Point-to-Point Tunneling Protocol (PPTP) from Microsoft. L2TP is an IETF standard endorsed by Cisco Systems and other networking industry leaders.

L2TPv3—The draft version of L2TP that enhances functionality in RFC 2661 (L2TP).

LMI—Local Management Interface.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the devices in the network where to forward packets based on preestablished IP routing information.

MQC—Modular quality of service CLI.

MTU—Maximum Transmission Unit. The maximum packet size, in bytes, that a particular interface can handle.

PMTU—Path MTU.

PVC—Permanent virtual circuit. A virtual circuit that is permanently established. A Frame Relay logical link, whose endpoints and class of service are defined by network management. Analogous to an X.25 permanent virtual circuit, a PVC consists of the originating Frame Relay network element address, originating data-link control identifier, terminating Frame Relay network element address, and termination data-link control identifier. Originating refers to the access interface from which the PVC is initiated. Terminating refers to the access interface at which the PVC stops. Many data network customers require a PVC between two points. PVCs save the bandwidth associated with circuit establishment and tear down in situations where certain virtual circuits must exist all the time. Data terminating equipment with a need for continuous communication uses PVCs.

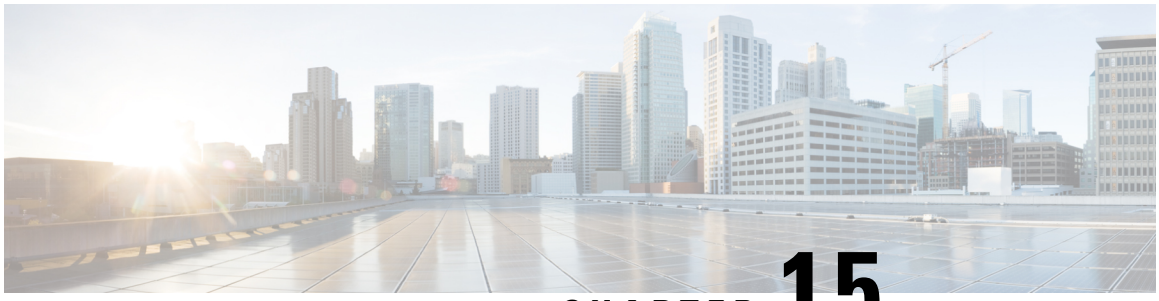
PW—Pseudowire.

SNMP—Simple Network Management Protocol. The network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices and manage configurations, statistics collection, performance, and security.

tunneling—Architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme.

UNI—User-Network Interface.

VPDN—Virtual private dialup network. A network that allows separate and autonomous protocol domains to share common access infrastructure, including modems, access servers, and ISDN devices. A VPDN enables users to configure secure networks that take advantage of ISPs that tunnel remote access traffic through the ISP cloud.



CHAPTER 15

Power over Ethernet

- Power over Ethernet, on page 217
- Device Detection and Power Allocation, on page 218
- Configuring Power Over Ethernet, on page 218
- **Configuring Universal PoE**, on page 219
- PoE Debug Commands, on page 220

Power over Ethernet

Power over Ethernet (PoE) is typically used to power up devices such as Access points, IP Cameras and IP Phones connected to the device's Ethernet ports.

The Cisco IR8340 routers support standard Power over Ethernet (PoE), Power over Ethernet Plus (PoE+), Cisco Enhanced Power over Ethernet (EPoE), and Cisco Universal Power over Ethernet (UPoE) on all copper ports. The total PoE available power is 120 W to be shared by the four LAN ports.

The following features are supported:

- UPoE—Supported on LAN port 1 and 2 with a maximum of 60 watts of power on each port
- PoE+—Supported on all PoE ports (LAN port 1-4) with a maximum of 30 watts of power on each port
- PoE—Supported on all PoE ports (LAN port 1-4) with a maximum of 15 watts of power on each port
- Support both Cisco PD and IEEE802.3af/IEEE802.3 on all ports, with DC power disconnected.
- Per port power consumption measurement.
- Ability to specify max power consumption on every port.
- PoE power policing—Comprises the following two modes, which determines the action to take on the interface after a port shuts down because of an inline-power policing violation:
  - **Logging**—An error message is logged to the console and the interface restarts; the device powers up.
  - **Errdisable** (default)—In addition to logging an error message to the console, the interface is placed in an errdisable state so that the device attached to the port does not receive inline-power until you restart the port or configure an errdisable autorecovery mechanism.
- Static power allocation on ports.

- Load Shedding upon PSU removal or failure.

Device Detection and Power Allocation

The router will detect a Cisco Pre-standard or an IEEE-compliant PD when the PoE is enabled and the connected device is not being powered by an AC adapter.

After device detection, the router will determine the power requirements based on power classification class. Depending on the available power in the power budget, the router determines if a port can be powered. The router initially allocates this power when it detects and powers the device. Power negotiation using CDP/LLDP protocols happens thereafter. Supported protocols for power negotiation are CDP for Cisco PD, and LLDP for non-Cisco PDs. Maximum power budget for 1 WAN port at any time is 15.4 W. On reload the PoE ports are powered down until the unit reboots.

Configuring Power Over Ethernet

Each copper port on the router can auto detect one of following connected devices, and supply power to them properly:

- An IEEE 802.3af and IEEE 802.3at compliant power device
- Cisco EPOE and UPOE power device

To configure power over ethernet, use these commands:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet0/1/0`	Specifies the physical port to be configured, and enters interface configuration mode.
Step 3	**power inline** [**auto**| **max** *max-wattage*] **never** **Example:** `Router(config-if)# power inline auto`	Configures the PoE mode on the port. The keywords have these meanings: • Auto—Enables powered-device detection. If enough power is available, automatically allocates power to the PoE port after device detection. This is the default setting. • Max max-wattage—Limits the power allowed on the port. The range for PoE+ ports is 4000 to 30000 mW. The range for

	Command or Action	Purpose
		Cisco UPOE ports is 4000 to 60000 mW. If no value is specified, the maximum is allowed. • Never—Disables device detection, and disable power to the port. **Note** If a port has a Cisco powere **never** command to configu port into the error-disabled
Step 4	**end** **Example:** `router(config-if)# end`	Returns to privileged EXEC mode.

What to do next

Use the following commands to check the PoE port status:

```
Router#show power inline
Available:120.0(w)  Used:70.0(w)  Remaining:50.0(w)
Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------- ----- ----
Gi0/1/0   auto   on         30.0    Ieee PD             4     60.0
Gi0/1/1   auto   on         30.0    Ieee PD             4     60.0
Gi0/1/2   auto   on         3.7     IP Phone 7811       1     30.0
Gi0/1/3   auto   on         6.3     IP Phone 7962       2     30.0
--------- ------ ---------- ---------- ---------- ------ -----
Totals:          4    on    70.0

Router#show power inline gigabitEthernet 0/1/0
Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------- ----- ----
Gi0/1/0   auto   on         60.0    IR8340-K9           4     60.0
Router#
```

Configuring Universal PoE

Cisco UPOE can provide a maximum of 60Watts power over both signal and spare pairs of RJ45 cable. UPOE capable switch port can enable spare pair and supply power to it through CDP or LLDP negotiations with UPOE power device automatically.

If end-point power device is capable to consume power on both signal and spare pairs but without corresponding CDP/LLDP negotiation mechanism available, following configurations can be used to manually force four-pair on specific port.

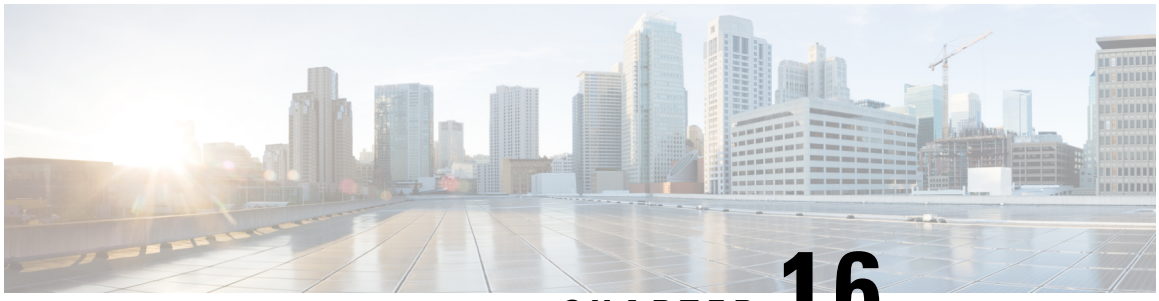
Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet0/1/0`	Specifies the physical port to be configured, and enters interface configuration mode.
Step 3	**power inline four-pair forced** **Example:** `Router(config-if)# power inline four-pair forced`	Forces power enabling on both signal and spare pairs from a switch port.
Step 4	**end** **Example:** `router(config-if)# end`	Returns to privileged EXEC mode.

PoE Debug Commands

The following table shows the PoE debug commands:

Command	Description
debug ilpower controller	Display PoE controller debug messages.
debug ilpower event	Display PoE event debug messages.
debug ilpower port	Display PoE port management debug messages.
debug ilpower powerman	Display PoE power management debug messages.
debug ilpower cdp	Display PoE CDP debug messages.
debug ilpower registries	Display PoE registries debug messages.
debug ilpower scp	Display PoE scp debug messages.



CHAPTER 16

Configuring the T1/E1 Network Interface Module

The Cisco T1/E1 Network Interface Modules (NIM) are inserted into the NIM slot on the router to provide T1, fractional T1, E1, and fractional E1 support for data applications.

- Information About T1/E1 Network Interface Module, on page 221
- Configuring T1/E1 Network Interface Module, on page 221

Information About T1/E1 Network Interface Module

The IR8340 router has two Network Interface Module (NIM) slots, 0/2 and 0/3. The T1/E1 Network Interface Module IRM-NIM-2T1E1 can be installed in these two slots. It is a 2-port channelized data module and supports 24/31 channel groups for T1/E1 per port. Each T1/E1 module has two ports, P0 and P1. Each port is linked to a controller in configuration as below:

- If the module is in slot 0/2, it has two controllers 0/2/0 and 0/2/1.
- If the Module is in slot 0/3, it has two controllers 0/3/0 and 0/3/1.

Use RJ-48 cables to connect the T1/E1 modules between two devices.

Configuring T1/E1 Network Interface Module

Configuring the Card Type

The T1/E1 network interface module will not be operational until a card type is configured.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**card type** {**t1**|**e1**} *slot subslot* **Example:** `Router(config)# card type t1 0 2`	Specifies card type as T1 or E1 for the network interface module. In this example, the T1/E1 module is connected on 0/2 slot.

Changing the Card Type

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**no card type** {**t1**|**e1**} *slot subslot* **Example:** `Router(config)# no card type t1 0 2`	(Optional) Removes the previous configuration.
Step 4	**card type** {**t1**|**e1**} *slot subslot* **Example:** `Router(config)# card type e1 0 2`	Specifies T1 or E1 connectivity for the network interface module.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits the card configuration mode and returns to global configuration mode.
Step 6	**write** **Example:** `Router(config)# write`	Rebuilds the router configuration.
Step 7	**reload** **Example:** `Router(config)# reload`	Reloads router so that changes can take effect. After this command executes, the router goes into the ROM monitor (rommon) mode.

	Command or Action	Purpose
Step 8	**boot** **Example:** `Router(rommon)# boot`	Boots the router with the configuration for the newly selected card type.

Configuring the T1/E1 Network Interface Module for Data Support

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**controller** {**t1**|**e1**} *slot/subslot/port* **Example:** `Router(config)# controller t1 0/2/0`	Enters controller configuration mode for the network interface module. • Valid values for slot is 0, subslot is 2 or 3, and port is 0 or 1.

Step 4	Do one of the following: **framing** {**sf**|**esf**} or **framing** {**crc4**|**no-crc4**}
	Example:
	`Router(config-controller)# framing esf`
	`Router(config-controller)# framing crc4`
		In T1 configurations, specifies super frame (**sf**) or extended super frame (**esf**) as the frame type for data lines. Default is **esf** .
		In E1 configurations, specifies cyclic redundancy check 4 (**crc4**) or **no-crc4** as the frame type for data lines. Default is **crc4** .

Step 5	Do one of the following: **linecode** {**ami**|**b8zs**} or **linecode** {**ami**|**hdb3**} **Example:** `Router(config-controller)# linecode b8zs`	In T1 configurations, specifies alternate mark inversion (AMI) or bipolar 8-zero substitution (b8zs) as the linecode. Default is **b8zs**. **Note** When using linecode AMI, groups created do not conta threshold specified by stand
Step 6	**fdl** {**att**|**ansi**|**both**} **Example:** `Router(config-controller)# fdl both`	T1 only. Sets the facility data link (fdl) exchange standard for T1 interfaces using esf framing. You can select the ATT standard (ATT TR54016), the ANSI standard (ANSI

	Command or Action	Purpose
		T1.403), or both standards. Default is **ansi** . To disable fdl, enter the **no fdl** command.
Step 7	**clock source** {**internal** | **line** [**primary** | **secondary**] | **network**} **Example:** `Router(config-controller)# clock source network`	Specifies the clock source. The options are as follows: • internal—Sets the controller framer as the clock master. The **clock source i nternal** command is only applicable with the **channel-group** command and the **pri-group** (for data) command. **Note** The **pri-group** command is su the keyword **voice-dsp** . • line—Specifies the phase-locked loop (PLL) on a port. When both a primary port and a secondary port are configured and the primary port fails, the PLL switches over to the secondary. When the PLL on the primary port becomes active again, the PLL automatically switches to the primary port. • network—Sets the controller to sync to the TDMSW clock for both TDM voice and data support. This configures the far end of the T1/E1 line as the clock line. Default is line.
Step 8	**line-termination** {**75-ohm**|**120-ohm**} **Example:** `Router(config-controller)# line-termination 75-ohm`	E1 only. Sets the line termination on an E1 controller. • **75-ohm** specifies 75-ohm unbalanced termination. • **120-ohm** specifies 120-ohm balanced termination.
Step 9	**loopback** {**diagnostic** | **local** {**payload** | **line** } | **remote** {**iboc** | **esf** {**payload** | **line** }}} **Example:** `Router(config-controller)# loopback remote esf line`	Sets the loopback method for testing the interface. Options are: • **diagnostic** —Loops the transmit signal back to receive. • **local** —Puts the interface into local loopback mode at the **payload** or **line** level.

	Command or Action	Purpose
		• **remote** —Puts the interface into remote loopback mode through an inband bit oriented code (iboc) or, for T1 only, **remote esf** , which uses fdl codes to set **payload** or **line** levels.
Step 10	Do one of the following: **cablelength long** *db-loss-value* or **cablelength short** *length* **Example:** `Router(config-controller)# cablelength short 110`	T1 only.The **cablelength long** command attenuates the pulse from the transmitter using pulse equalization and line build-out. This command applies to cables longer than 660 feet. Loss values are: • 0db • -7.5db • -15db • -22.5db Default attenuation is 0db. The **cablelength short** command sets transmission attenuation for cable lengths of 660 feet or less. When you use the **cablelength short** command, specify the length as follows: • **110** for cable lengths from 0 to 110 feet • **220** for cable lengths from 111 to 220 feet • **330** for cable lengths from 221 to 330 feet • **440** for cable lengths from 331 to 440 feet • **550** for cable lengths from 441 to 550 feet • **660** for cable lengths from 551 to 660 feet There is no default cable length.
Step 11	**channel group** *channel-group-number* {**timeslots** *range* [**speed** *kbps*] | **unframed**} **Example:** `Router(config-controller)# channel group 1 timeslots 1-4`	Configures the serial WAN on a T1 or E1 interface by specifying channels and their timeslots. For T1, values are as follows: • channel-group-number is from 0 to 23. • **timeslots** range is from 1 to 24. • Default value of **speed** for T1 is 64 kbps. Configuration of **speed** is optional. For E1, values are as follows:

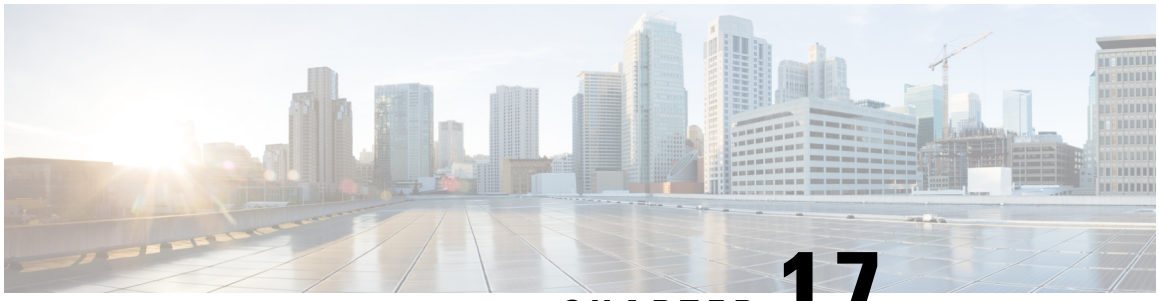
	Command or Action	Purpose
		• channel-group-number is from 0 to 30. • **timeslots** range is from 1 to 31. • Default value of **speed** for E1 is 64 kbps. Configuration of **speed** is optional. • **unframed** (E1 only) specifies that all 31 timeslots are to be used for data and that none are to be used for framing signals.
Step 12	**national reserve** *N sa4 sa5 sa6 sa7 sa8* **Example:** `Router(config-controller)# national reserve 0 1 1 1 1 0`	E1 only. Sets the six required national bits in E1 in the G.751 frame. Default is 1 1 1 1 1 1.
Step 13	**crc-threshold** *value* **Example:** `Router(config-controller)# crc-threshold 500`	T1 only. Defines a severely errored second by specifying the number of CRC errors that must occur in one second to reach the severely errored second state. Default is 320.
Step 14	**yellow** {**generation**|**detection**} **Example:** `Router(config-controller)# no yellow detection`	Enables generation and detection of yellow alarms. Default condition is that **generation** and **detection** of yellow alarms are enabled. Use the **no** form of the command to disable yellow alarm detection.
Step 15	**bert pattern** *pattern* **interval** *time* **Example:** `Router(config-controller)# bert pattern 2^11 interval 1440`	(Optional) Activates the BERT with the chosen test pattern for a specified duration. Configure BERT patterns on the T1/E1 network interface modules as follows: • When the linecode is AMI, use patterns 2^11, 2^15, or 2^20-QRSS. • When the linecode is b8zs or hdb3, use patterns 2^11, 2^15, 2^20-QRSS, or 2^20-O.153. • The **interval** time is from 1 to 14,400 minutes.

Example of T1/E1 Network Interface Module Configuration

The following example shows the configuration of the router with the T1/E1 NIM installed and configured for data.

```
card type t1 0 2
controller T1 0/2/0
framing esf
linecode b8zs
```

```
cablelength long 0db
channel-group 0 timeslots 1
channel-group 1 timeslots 2
channel-group 2 timeslots 3
interface Serial0/2/0:0
 ip address 1.1.1.1 255.255.255.0
!
interface Serial0/2/0:1
 ip address 2.2.2.1 255.255.255.0
!
interface Serial0/2/0:2
 ip address 3.3.3.1 255.255.255.0
```



CHAPTER 17

Configuring the Serial Network Interface Module

- About the Asynchronous/Synchronous Serial Network Interface Module, on page 229
- Configuring the Serial Interface to Sync or Async Mode, on page 229
- Configuring Synchronous Serial Ports, on page 230
- Configuring Asynchronous Serial Ports, on page 237

About the Asynchronous/Synchronous Serial Network Interface Module

The 8-port Asynchronous/Synchronous Network Interface Module (NIM) IRM-NIM-RS232 provides asynchronous/synchronous serial connections supporting EIA-RS232 for the Cisco IR8340 Router.

The IR8340 router has two NIM slots, 0/2 and 0/3. The serial NIMs can be installed in these two slots.

Each RS-232 Serial Module has 8 serial interfaces. The interface numers are:

- serial 0/2/0 - serial 0/2/7—If the serial module is in slot 0/2.
- serial 0/3/0 - serial 0/3/7—If the serial module is in slot 0/3.

The following features are supported:

- Supports DCE and DTE.
- Each serial port can be configured as either Asynchronous or Synchronous mode.
- A maximum speed of 256 kbps is supported for RS232 Synchronous port.
- A maximum speed of 230.4 kbps is supported for Asynchronous port.

Configuring the Serial Interface to Sync or Async Mode

By using the Physical Layer mode option, you can configure the serial interface to Synchronous or Asynchronous mode as needed. By default it will be in Synchronous mode. Use the following commands to configure the serial interface:

interface serial *slot/subslot/port*

physical-layer {**async**|**sync**}

```
Router(config)#interface Serial0/3/0
Router(config-if)#physical-layer ?
  async  Configure asynchronous physical layer on serial interface
  sync   Configure synchronous physical layer on serial interface

Router(config-if)#
```

Configuring Synchronous Serial Ports

After the serial port mode is configured as synchronous mode, connect the DCE type serial cable to one side and DTE cable on the other side. The serial link will come up.

Checking DCE and DTE Cable Type

After the serial port mode is configured as Sync, connect the DCE type serial cable to one side and DTE cable on the other side. The serial link will come up. Use the following command to check cable type:

```
Router1#show controllers Serial 0/3/0

Serial0/3/0 - (IRM-NIM-RS232) is administratively down
 Encapsulation : HDLC
 Cable type: RS-232 DCE
 mtu 1500, max_buffer_size 1524, max_pak_size 1656 enc 132
 loopback: Off,  crc: 16, invert_data: Off
 nrzi: Off, idle char: Flag
 dce_terminal_timing_enable: Off ignore_dtr: Off
 tx_clockrate: 64000bps, rx_clockrate: 64000bps, serial_clock_index: 0
 serial_restartdelay:60000,  serial_restartdelay_def:60000

     RTS up, CTS down, DTR up, DCD down, DSR down

Router1#

Router2#show controllers Serial 0/3/1

Serial0/3/1 - (IRM-NIM-RS232) is administratively down
 Encapsulation : HDLC
 Cable type: RS-232 DTE
 mtu 1500, max_buffer_size 1524, max_pak_size 1656 enc 132
 loopback: Off,  crc: 16, invert_data: Off
 nrzi: Off, idle char: Flag
 tx_invert_clk: Off, ignore_dcd: Off
 tx_clockrate: 31998bps, rx_clockrate: 31998bps, rx_clock_threshold: 0
 serial_restartdelay:60000,  serial_restartdelay_def:60000

     RTS down, CTS down, DTR down, DCD down, DSR down

Router2#
```

The **show interfaces serial0/3/0 controller** command can also be used to check the cable type.

Specifying Synchronous Serial Encapsulation

The synchronous serial interfaces support the following serial encapsulation methods:

- Frame-Relay

- PPP (chap and pap)
- HDLC (default)

Use the following command to configure synchronous serial encapsulation. To remove the configuration, use the no form of the command.

[no] encapsulation {frame-relay|ppp|hdlc}

DCE and DTE Configuration for HDLC Encapsulation

The following examples show the basic DCE and DTE configuration for HDLC encapsulation.

```
U1 configuration (DCE Side)
======================
U1#show running-config interface Serial0/3/1
Building configuration...
Current configuration : 99 bytes
!
interface Serial0/3/1
ip address 31.31.31.1 255.255.255.0
no keepalive
clock rate 256000
end

U1#show interfaces Serial0/3/1
Serial0/3/1 is up, line protocol is up
Hardware is IRM-NIM-RS232
Internet address is 31.31.31.1/24
MTU 1500 bytes, BW 256 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 2/255
Encapsulation HDLC, loopback not set
Keepalive not set
Last input 00:00:31, output 00:00:09, output hang never

U1#ping 31.31.31.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 31.31.31.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/8 ms

U2 configuration (DTE Side)
======================
U2#show running-config interface Serial0/3/1
Building configuration...
Current configuration : 80 bytes
!
interface Serial0/3/1
ip address 31.31.31.2 255.255.255.0
no keepalive
end

U2#show interfaces Serial0/3/1
Serial0/3/1 is up, line protocol is up
Hardware is IRM-NIM-RS232
Internet address is 31.31.31.2/24
MTU 1500 bytes, BW 2000 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive not set
Last input 00:00:33, output 00:00:10, output hang never

U2#ping 31.31.31.1
```

```
*Jan 26 15:29:49.206 IST: %LINK-3-UPDOWN: Interface Serial0/3/6, changed state to down
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 31.31.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/8/9 ms
```

DCE and DTE Configuration for PPP (CHAP) Encapsulation

The following examples show the basic DCE and DTE configuration for PPP (CHAP) encapsulation.

```
U1 configuration (DCE side)
================================
U1#show running-config interface Serial0/3/2
Building configuration...
Current configuration : 118 bytes
!
interface Serial0/3/2
ip address 32.32.32.1 255.255.255.0
encapsulation ppp
no keepalive
clock rate 256000
end
U1#show interfaces Serial0/3/2
Serial0/3/2 is up, line protocol is up
Hardware is IRM-NIM-RS232
Internet address is 32.32.32.1/24
MTU 1500 bytes, BW 256 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Open
Open: IPCP, CDPCP, loopback not set
Keepalive not set
Last input 00:00:13, output 00:00:13, output hang never

===configuration for ppp-chap authentication=======
U1(config)#username u1 secret userpassword
U1#show running-config interface Serial0/3/1
Building configuration...
Current configuration : 143 bytes
!
interface Serial0/3/1
ip address 31.31.31.2 255.255.255.0
encapsulation ppp
no keepalive
ppp authentication chap
clock rate 256000
end

U1#ping 32.32.32.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 32.32.32.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/8 ms

U2 configuration (DTE Side)
=====================
U2#show running-config interface Serial0/3/2
Building configuration...
Current configuration : 99 bytes
!
interface Serial0/3/2
ip address 32.32.32.2 255.255.255.0
encapsulation ppp
no keepalive
end
```

```
U2#show interfaces Serial0/3/2
Serial0/3/2 is up, line protocol is up
Hardware is IRM-NIM-RS232
Internet address is 32.32.32.2/24
MTU 1500 bytes, BW 2000 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Open
Open: IPCP, CDPCP, loopback not set
Keepalive not set
Last input 00:00:08, output 00:00:15, output hang never

===configuration for ppp-chap authentication=======
U2(config)#username U1 secret userpassword
U2#show running-config interface Serial0/3/1
Building configuration...
Current configuration : 124 bytes
!
interface Serial0/3/1
ip address 31.31.31.2 255.255.255.0
encapsulation ppp
no keepalive
ppp authentication chap
end

U2#ping 32.32.32.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 32.32.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/9 ms
```

DCE and DTE Configuration for PPP (PAP) Encapsulation

The following examples show the basic DCE and DTE configuration for PPP (PAP) encapsulation.

```
U1 configuration (DCE side)
================================
U1#show running-config interface Serial0/3/2
Building configuration...
Current configuration : 118 bytes
!
interface Serial0/3/2
ip address 32.32.32.1 255.255.255.0
encapsulation ppp
no keepalive
clock rate 256000
end
U1#show interfaces Serial0/3/2
Serial0/3/2 is up, line protocol is up
Hardware is IRM-NIM-RS232
Internet address is 32.32.32.1/24
MTU 1500 bytes, BW 256 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Open
Open: IPCP, CDPCP, loopback not set
Keepalive not set
Last input 00:00:13, output 00:00:13, output hang never

===configuration for ppp-pap authentication=======
U1(config)#username u1 secret userpassword
U1#show running-config interface Serial0/3/1
Building configuration...
Current configuration : 181 bytes
!
```

```
interface Serial0/3/1
ip address 31.31.31.1 255.255.255.0
encapsulation ppp
ppp authentication pap
ppp pap sent-username Sumatra-1 password 0 userpassword
clock rate 256000
end
U1#

U1#ping 32.32.32.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 32.32.32.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/8 ms

U2 configuration (DTE Side)
======================
U2#show running-config interface Serial0/3/2
Building configuration...
Current configuration : 99 bytes
!
interface Serial0/3/2
ip address 32.32.32.2 255.255.255.0
encapsulation ppp
no keepalive
end

U2#show interfaces Serial0/3/2
Serial0/3/2 is up, line protocol is up
Hardware is IRM-NIM-RS232
Internet address is 32.32.32.2/24
MTU 1500 bytes, BW 2000 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Open
Open: IPCP, CDPCP, loopback not set
Keepalive not set
Last input 00:00:08, output 00:00:15, output hang never

===configuration for ppp-pap authentication=======
U2(config)#username U1 secret userpassword
U2#show running-config interface Serial0/3/1
Building configuration...
Current configuration : 176 bytes
!
interface Serial0/3/1
ip address 31.31.31.2 255.255.255.0
encapsulation ppp
no keepalive
ppp authentication pap
ppp pap sent-username Sumatra-2 password 0 userpassword
end
U2

U2#ping 32.32.32.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 32.32.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/9 ms
```

DCE and DTE Configuration for Frame Relay Encapsulation

The following examples show the basic DCE and DTE configuration for Frame Relay encapsulation.

```
U1 configuration (DCE side)
================================
U1#show running-config interface Serial0/3/3
Building configuration...
Current configuration : 158 bytes
!
interface Serial0/3/3
ip address 33.33.33.1 255.255.255.0
encapsulation frame-relay
no keepalive
clock rate 256000
frame-relay interface-dlci 100
end
U1#show interfaces Serial0/3/3
Serial0/3/3 is up, line protocol is up
Hardware is IRM-NIM-RS232
Internet address is 33.33.33.1/24
MTU 1500 bytes, BW 256 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation FRAME-RELAY, loopback not set
Keepalive not set
FR SVC disabled, LAPF state down
Broadcast queue 0/64, broadcasts sent/dropped 1/0, interface broadcasts 0
Last input 00:00:16, output 00:00:16, output hang never

U1#ping 33.33.33.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 33.33.33.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/9 ms

U2 configuration (DTE Side)
======================
U2#show running-config interface Serial0/3/3
Building configuration...
Current configuration : 139 bytes
!
interface Serial0/3/3
ip address 33.33.33.2 255.255.255.0
encapsulation frame-relay
no keepalive
frame-relay interface-dlci 100
end
U2#show interfaces Serial0/3/3
Serial0/3/3 is up, line protocol is up
Hardware is IRM-NIM-RS232
Internet address is 33.33.33.2/24
MTU 1500 bytes, BW 2000 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation FRAME-RELAY, loopback not set
Keepalive not set
FR SVC disabled, LAPF state down
Broadcast queue 0/64, broadcasts sent/dropped 1/0, interface broadcasts 0
Last input 00:05:05, output 00:05:05, output hang never

U2#ping 33.33.33.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 33.33.33.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/8 ms
```

Serial Synchronous Show and Debug Commands

This section provides show and debug commands of the serial synchronous interfaces.

1. Connect to the NIM console using Telnet or Secure Shell (SSH) and open a session to the device using the **hw-module session** *slot/subslot* command in privileged EXEC mode.

   To exit the session, press **Ctrl-a** and **Ctrl-q** from your keyboard.

2. **show platform hardware subslot** *slot-number* **module device ge-stats**
3. **show platform hardware subslot** *slot-number* **module host-if status**
4. **show platform hardware subslot** *slot-number* **module host-if statistics**
5. **show platform hardware subslot** *slot-number* **module device ?**
   - help: the current information
   - scc-info: module serial information
   - scc-stats: module serial stats
   - scc-stats-p<x>: module serial port stats
   - ge-info: module ge information
   - ge-stats: module ge stats
   - fpga: module fpga information
   - fpga-p<x>: module fpga port information
   - fc-p<x>: module port flow control information
   - mempool: module mempool information
6. **debug hw-module subslot** *slot-number* **?**
   - commands: Control plane configuration and commands
   - errors: Error handling and race conditions
   - events: Control plane event notifications
   - ha: SPA Specific HA
   - interrupts: Interrupt handling
   - obfl: SPA Specific OBFL (on-board failure logging)
   - oir: SPA OIR information
   - periodic: Periodic processing (for example, plugin_one_sec)
7. **show monitor event-trace spa all**
8. Use **show diag subslot** *slot-number* **eeprom detail** to check NIM eeprom contents

Configuring Asynchronous Serial Ports

Configure the serial interface to Asynchronous mode by using these commands:

```
Router(config)#interface Serial0/3/0
Router(config-if)#physical-layer async
```

Specifying Asynchronous Serial Encapsulation

The asynchronous serial interfaces support the following serial encapsulation methods:

- Block Serial tunneling (BSTUN)
- Raw socket TCP (Raw-TCP)
- Raw socket UDP (Raw-UDP)
- SCADA

Use the following command to configure asynchronous serial encapsulation. To remove the configuration, use the no form of the command.

[no] encapsulation {**bstun**|**raw-tcp**|**raw-udp**|**scada**}

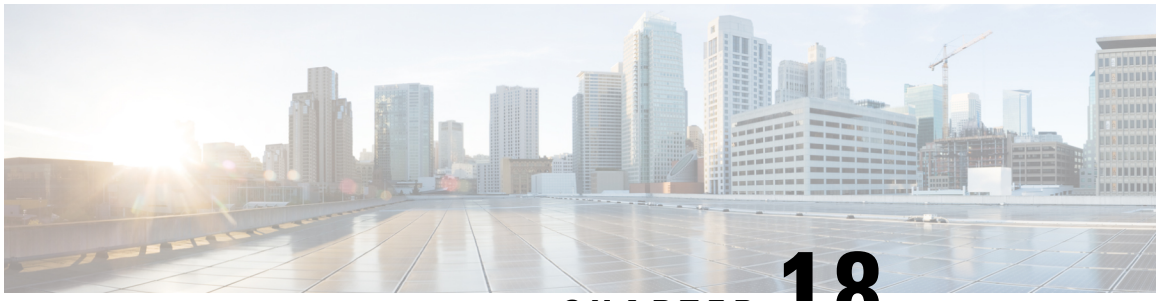
Example

```
Router(config)#interface Serial0/3/0
Router(config-if)#physical-layer async
Router(config-if)#encapsulation raw-tcp
Router(config-if)#end
Router#show running-config interface Serial0/3/0
Building configuration...

Current configuration : 89 bytes
!
interface Serial0/3/0
 physical-layer async
 no ip address
 encapsulation raw-tcp
end

Router#
```

Encapsulation methods are set according to the type of protocol or application you configure in the Cisco IOS software.



CHAPTER 18

Cellular pluggable interface modules

The Cisco 4G LTE-Advanced Configuration chapter has been replaced by a new standalone guide called Cellular Pluggable Interface Module Configuration Guide . This guide contains updated information on all aspects of using the Cisco Cellular PIM.

Important The Pluggable Module is not hot swappable. The router must be reloaded after a new module is installed.

- Support for P-5GS6-R16SA-GL pluggable module, on page 239
- Galileo support on the LTE pluggable modules, on page 239
- Radio signal parameters and GPS coordinates of cellular telemetry monitoring using syslog messages, on page 241
- GPS NMEA unique identifier , on page 242
- Configure cellular modem GPS NMEA unique identifier, on page 244

Support for P-5GS6-R16SA-GL pluggable module

From Cisco IOS-XE Release 17.18.2, the P-5GS6-R16SA-GL Pluggable Module is supported on ESR6300 routers.

Support for the P-5GS6-R16SA-GL Pluggable Module works the same way on the ESR6300 routers as it does on the other IoT routers. For details, see 5G Sub-6 GHz Pluggable Interface Module and Cellular Pluggable Interface Module Configuration Guide .

Galileo support on the LTE pluggable modules

The Cisco IOS XE 17.11.1a release introduces support for the Galileo constellation on LTE pluggable modules.

GPS was the only GNSS constellation supported in earlier releases. The Cisco IOS XE 17.11.1a release introduces support for Galileo constellation.

Note You can enable only one constellation at a time.

Configuration command examples:

```
config# controller cellular<slot/port>
(config-controller)# <no> lte gps constellation <gps | galileo | gnss >
```

Example:

```
(config-controller)# lte gps constellation ?
galileo  select Galileo as active constellation
gps      select GPS as active constellation
gnss      select multiple GNSS as active constellation
```

Note The default setting is GPS mode.

The galileo and gnss options in this CLI configure Galileo and simultaneous GNSS modes, such as GPS and Galileo.

If you disable the GPS configuration, make sure that no constellation is configured. This maintains consistency with GPS mode configuration.

Example:

```
config#


controller Cellular 0/4/0
(config-controller)# no lte gps constellation gps
```

Show commands example:

Note The example shows the current GNSS constellation as Galileo.

Example:

```
config#


show cellular 0/4/0 gps detail
GPS Feature =  enabled
GPS Mode Configured =  standalone
Current Constellation Configured =  galileo
GPS Port Selected =  Dedicated GPS port
GPS Status =  GPS acquiring
```

Any changes made to the configuration will require the router to be rebooted.

More information is available in the Cellular Pluggable Interface Module Configuration Guide .

Radio signal parameters and GPS coordinates of cellular telemetry monitoring using syslog messages

Cellular telemetry is a feature that enables real-time monitoring and analysis of cellular connection performance by collecting Radio Frequency (RF) parameters and Global Positioning System (GPS) coordinates from the cellular network at a fixed interval of 60 seconds.

- Monitors key radio signal parameters such as RSSI, RSRP, RSRQ, PCI, and SNR.
- Collects GPS coordinates for location tracking.
- Facilitates troubleshooting, network optimization, and ensures reliable connectivity.

Cellular Telemetry Reference Information

Cellular telemetry parameters that can be monitored include:

- Received Signal Strength Indicator (RSSI)
- Reference Signal Received Power (RSRP)
- Reference Signal Received Quality (RSRQ)
- Physical Cell Identity (PCI)
- Signal to Noise Ratio (SNR)
- Global Positioning System (GPS) coordinates

The Cisco Catalyst WAN Manager does not support cellular telemetry.

Enable cellular telemetry

To enable the cellular telemetry feature in the contoller cellular interface 0/x/0 using the CLI.

Before you begin

- Insert the cellular Pluggable Interface Module (PIM) into the IR device.
- Enable GPS in the controller for GPS coordinates to be displayed in syslogs.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Router#configure terminal
```

Step 2 Enter the cellular configuration mode.

Example:

```
Router (config)#controller cellular 0/4/0
```

Step 3 Enable the RF parameters and GPS coordinates.

Example:

```
Router (config-controller)#lte modem serviceability signal-parameters
Router (config-controller)#end
```

Disable cellular telemetry

To disable the cellular telemetry feature, use the no form of the **lte modem serviceability signal-parameters** command, as shown in the example:

```
Router#Configure terminal


Router(config)#controller cellular 0/4/0
Router(config-controller)#no lte modem serviceability signal-parameters
Router(config-controller)#end
```

Monitor cellular telemetry

You can monitor RF parameters and GPS coordinates for cellular interfaces by using the **show logging** command or check the console output.

The following example displays RF parameters and GPS coordinates at every one minute interval:

```
Router#show logging
*Sep  3 17:08:42.081: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/4/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 27.4 dB
Latitude =  12 Deg 56 Min 8.9260 Sec North
Longitude =  77 Deg 41 Min 44.1641 Sec East

*Sep  3 17:09:42.080: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/4/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 29.0 dB
Latitude =  12 Deg 56 Min 8.8989 Sec North
Longitude =  77 Deg 41 Min 44.1570 Sec East
```

The following example displays only RF parameters and not GPS coordinates since GPS is not enabled in the controller:

```
Router#show logging
*Sep  3 17:08:42.081: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/4/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 27.4 dB

*Sep  3 17:09:42.080: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/4/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 29.0 dB
```

GPS NMEA unique identifier

Starting with Cisco IOS-XE Release 26.1.1, Cisco routers can send a Unique Identifier (UID) to identify the source device when streaming NMEA GPS sentences over UDP on the Cisco IR1101 and Cisco IR1835

routers. UID streaming is supported from the Dead Reckoning (DR) module and from cellular modems that support GNSS. The UID uniquely identifies each router in GPS data processing systems.

Table 17: Feature History Table

Feature name	Release information	Feature description
Send a UID to identify the unique source when sending GPS coordinates data	Release 26.1.1	This feature allows you to configure Cisco routers to send a Unique Identifier (UID) while streaming NMEA GPS sentences over UDP. This UID streaming is supported by both the Dead Reckoning (DR) module and cellular modems with GNSS capabilities, enabling each router to be uniquely identified in GPS data processing systems.

Cisco proprietary NMEA sentence format

When you configure a UID, a Cisco proprietary NMEA sentence is sent alongside the standard NMEA sentence in the same UDP packet. The format is:

```
$PCSCU,<UID>,<Source>,<Router Time>,<NMEA Sentence Time>,<NMEA Sentence Checksum>,,,,,,*95
```

- `CSC` indicates Cisco
- `U` indicates this sentence provides a unique identifier
- `<UID>` is the serial number, hostname, or custom ID
- `<Source>` indicates the origin of the NMEA stream:
  - `0`: cellular modem0
  - `1`: cellular modem1
  - `2`: DR module

For example, for cellular modem1, if you use the serial number as the UID, the Cisco proprietary NMEA sentence appears as:

```
$PCSCU,FCW2447P0EU,1,213431.100,213431.000,75,,,,,,*95
```

In this NMEA sentence

- `CSC` indicates Cisco
- `U` indicates that this sentence provides a unique identifier
- `FCW2447P0EU` indicates serial number as the UID
- `1` indicates source as cellular modem1
- `1213431.100` indicates the router time
- `213431.000` indicates the NMEA sentence time

- `75` indicates the NMEA sentence checksum

NMEA sentence selection

By default, all NMEA sentences from the modem or DR module are sent to the configured UDP destination. To reduce UDP traffic, you can select specific NMEA sentence types to send using a hex bitmask. The table presents each NMEA sentence type and its corresponding HEX value.

Table 18: NMEA sentence type and HEX value

HEX value	NMEA sentence type
0x01	GGA
0x02	GSA
0x04	GSV
0x08	RMC
0x10	VTG
0x20	GNS

Configure cellular modem GPS NMEA unique identifier

Follow these steps to configure cellular modem GPS NMEA UID.

Procedure

Step 1 Use the **configure terminal** command to enter the configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **controller cellular** *slot* command to enter the controller cellular configuration mode.

Example:

```
Router(config)# controller cellular 0/x/0
```

Step 3 Use the **lte gps nmea uid** {**custom** *unique-identifier* **| hostname | serial-number**} command to enable the GPS NMEA UID.

UID can be any of these:

- **custom**: user-configured string of upto 50 characters.
- **hostname**: host name of the router.
- **serial-number**: serial number of the router.

When using **serial-number** or **hostname**, the UID is automatically extracted and added to the proprietary NMEA sentence. This allows uniform CLI configuration across routers.

When using a **custom ID**, each router must be configured with a *unique-identifier* value to ensure distinct identification.

Example:

```
Router(config-controller)# lte gps nmea uid custom 1428
```

Example:

```
Router(config-controller)# lte gps nmea uid hostname
```

Example:

```
Router(config-controller)# lte gps nmea uid serial-number
```

Step 4 Use the **lte gps nmea ip udp** *source-address destination-address destination-port* **stream** *stream-id* command to specify the source and destination IP address.

Example:

```
Router(config-controller)# lte gps nmea ip udp 10.195.79.179 171.70.55.77 14013 stream 1
```

Step 5 Use the **lte gps nmea filter** *hex-value* command to select the sentence types.

hex-value: HEX bitmask for NMEA message filtering (0x01:GGA, 0x02:GSA, 0x04:GSV, 0x08:RMC, 0x010:VTG, 0x20:GNS)

This example shows the HEX value if you want to send sentence type as only RMC (0x08) and GNS (0x20):

Example:

```
Router(config-controller)# lte gps nmea filter 28
```

This example shows the HEX value if you want to send sentence type as only GGA (0x01) and GSA (0x02):

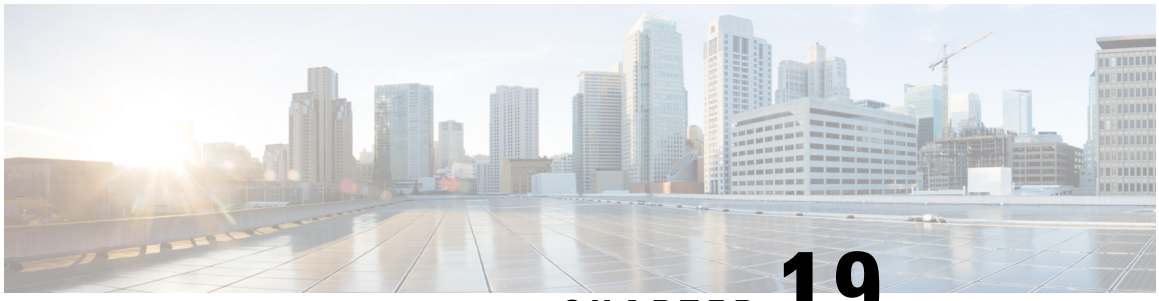
Example:

```
Router(config-controller)# lte gps nmea filter 03
```

Step 6 Use the **end** command to exit the controller configuration mode and return to the privileged EXEC mode.

Example:

```
Router(config-controller)# end
```



CHAPTER 19

Information About SCADA

- SCADA Overview, on page 247
- Role of the IR8340, on page 247
- Key Terms, on page 248
- Protocol Translation Application, on page 248
- Prerequisites, on page 249
- Guidelines and Limitations, on page 250
- Default Settings, on page 250
- Configuring Protocol Translation, on page 250
- Configuring the T101 Protocol Stack, on page 252
- Configuring the T104 Protocol Stack, on page 254
- Configuration Example, on page 258
- Configuring the DNP3 Protocol Stacks, on page 259
- Starting and Stopping the Protocol Translation Engine, on page 263
- Verifying Configuration, on page 264
- Debug Commands, on page 265

SCADA Overview

SCADA refers to a control and management system employed in industries such as water management, electric power, and manufacturing. A SCADA system collects data from various types of equipment within the system and forwards that information back to a Control Center for analysis. Generally, individuals located at the Control Center monitor the activity on the SCADA system and intervene when necessary.

The Remote Terminal Unit (RTU) acts as the primary control system within a SCADA system. RTUs are configured to control specific functions within the SCADA system, which can be modified as necessary through a user interface.

Change the serial interface mode as async and then configure the interface.

Role of the IR8340

In the network, the Control Center always serves as the master in the network when communicating with the IR8340. The IR8340 serves as a proxy master station for the Control Center when it communicates with the RTU.

The IR8340 provides protocol translation to serve as a SCADA gateway to do the following:

- Receive data from RTUs and relay configuration commands from the Control Center to RTUs.
- Receive configuration commands from the Control Center and relay RTU data to the Control Center.
- Terminate incoming requests from the Control Center, when an RTU is offline.

The IR8340 performs Protocol Translation for the following protocols:

- IEC 60870 T101 to/from IEC 60870 T104
- DNP3 serial to DNP3 IP

Key Terms

The following terms are relevant when you configure the T101 and T104 protocol stacks on the IR8340:

- Channel—A channel is configured on each IR8340 serial port interface to provide a connection to a single RTU for each IP connection to a remote Control Center. Each connection transports a single T101 (RTU) or T104 (Control Center) protocol stack.
- Link Address—Refers to the device or station address.
- Link Mode (Balanced and Unbalanced)—Refers to the modes of data transfer.
  - An Unbalanced setting refers to a data transfer initiated from the master.
  - A Balanced setting can refer to either a master or slave initiated data transfer.
- Sector—Refers to a single RTU within a remote site.
- Sessions—Represents a single connection to a remote site.

The following terms are relevant when you configure the DNP3 protocol stacks on the on the IR8340:

- Channel—A channel is configured on the IR8340 serial port interface to provide a connection to a single RTU for each IP connection to a remote Control Center. Each connection transports a single DNP3 serial (RTU) or DNP3 IP (Control Center) protocol stack.
- Link Address—Refers to the device or station address.
- Sessions—Represents a single connection to a remote site.

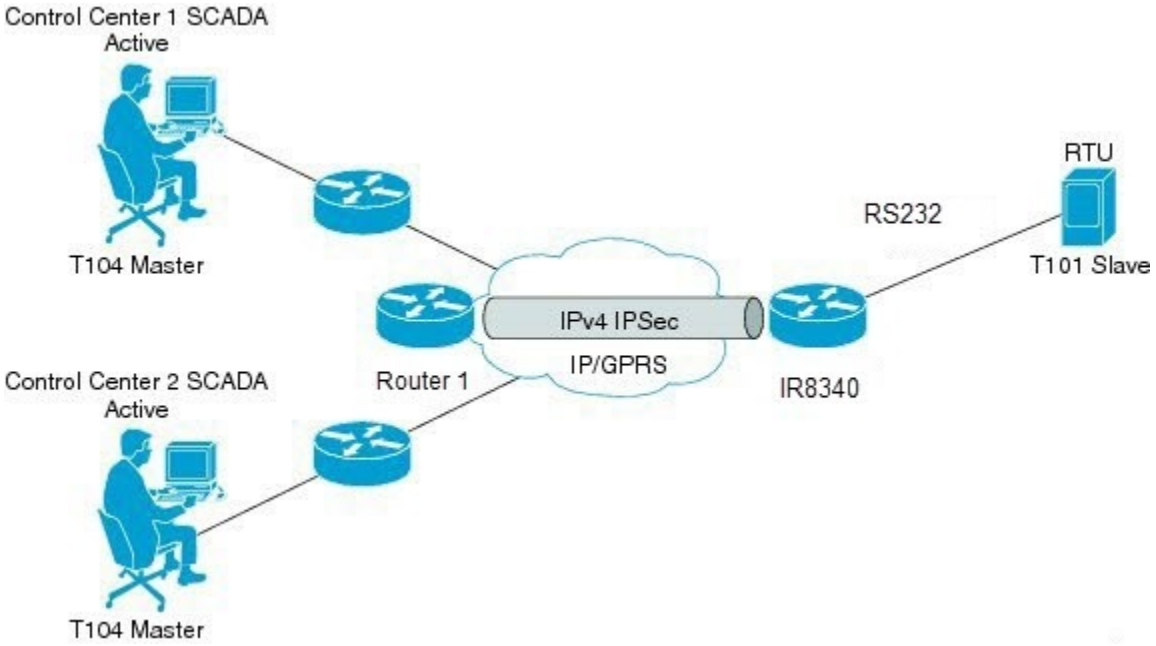
Protocol Translation Application

In the following figure, IR8340 (installed within a secondary substation of the Utility Network) employs Protocol Translation to provide secure, end-to-end connectivity between Control Centers and RTUs within a SCADA System.

The IR8340 connects to the RTU (slave) through a RS232 connection. To protect the traffic when forwarded over public infrastructures (for example, cellular), the IR8340 forwards SCADA data from the RTU to the Control Center in the SCADA system through an IPSec tunnel (FlexVPN site-to-site or hub and spoke). The IPSec tunnel protects all traffic between the IR8340 and the Head-end aggregation router. SCADA traffic can

be inspected through an IPS device positioned in the path of the SCADA traffic before it is forwarded to the proper Control Center.

Figure 6: Routers Within a SCADA System



Prerequisites

RTUs must be configured and operating in the network.

For each RTU that connects to the IR8340, you will need the following information for T101/T104:

- Channel information
  - Channel name
  - Connection type: serial
  - Link transmission procedure setting: unbalanced or balanced
  - Address field of the link (number expressed in octets)
- Session information
  - Session name
  - Size of common address of Application Service Data Unit (ASDU) (number expressed in octets)
  - Cause of transmission (COT) size (number expressed in octets)
  - Information object address (IOA) size (number expressed in octets)
- Sector information
  - Sector name

- ASDU address, (number expressed in octets)

For each RTU that connects to the IR8340, you will need the following information for DNP3:

- Channel information
  - Channel name
  - Connection type: serial
  - Link address
- Session information
  - Session name

Guidelines and Limitations

- Each channel supports only one session.
- Each sessions supports only one sector.
- The object types 8, 17, 18, 19, 20, 38, 39, and 40 are not supported for IEC protocol translation.

Default Settings

DNP3 Parameters	Default
Unsolicited Response (DNP3-serial)	Not Enabled
Send Unsolicited Message (DNP3-IP)	Enabled

Configuring Protocol Translation

Before making any configuration changes to a IR8340 operating with Protocol Translation, please review the section on Starting and Stopping the Protocol Translation Engine, on page 263.

Enabling the IR8340 Serial Port and SCADA Encapsulation

Before you can enable and configure Protocol Translation on the IR8340, you must first enable the serial port on the IR8340 and enable SCADA encapsulation on that port.

Before you begin

Determine availability of serial port on the IR8340.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters the global configuration mode.
Step 2	**interface serial** *slot/port/interface*	Enters the interface command mode for the serial slot/port/interface. *slot* –value of 0 *port* –value of 2 or 3 *interface* –value of 0
Step 3	**physical-layer async**	Configure the serial interface to Asynchronous mode.
Step 4	**no shutdown**	Brings up the port, administratively.
Step 5	**encapsulation scada**	Enables encapsulation on the serial port for protocol translation and other SCADA protocols.

EXAMPLE

This example shows how to enable serial port 0/2/0 and how to enable encapsulation on that interface to support SCADA protocols.

```
router# configure terminal
router(config)# interface serial 0/2/0
router(config-if)# physical-layer async
router(config-if)# no shutdown
router(config-if)# encapsulation scada
```

Configuring T101 and T104 Protocol Stacks

You can configure T101 and T104 protocol stacks, which allow end-to-end communication between Control Centers (T104) and RTUs (T101) within a SCADA system.

- Configuring the T101 Protocol Stack, on page 252
- Configuring the T104 Protocol Stack, on page 254
- Starting and Stopping the Protocol Translation Engine, on page 263

Prerequisites

Ensure that you have gathered all the required configuration information.

Enable the serial port and SCADA encapsulation.

Configuring the T101 Protocol Stack

Configure the channel, session, and sector parameters for the T101 protocol stack.

Before you begin

Ensure that you have gathered all the required configuration information. (See Prerequisites, on page 249.)

Enable the serial port and SCADA encapsulation. (See Enabling the IR8340 Serial Port and SCADA Encapsulation, on page 250.)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**scada-gw protocol t101**	Enters the configuration mode for the T101 protocol.
Step 3	**channel** *channel_name*	Enters the channel configuration mode for the T101 protocol. *channel_name* –Identifies the channel on which the serial port of the IR8340 communicates to the RTU. **Note** When the entered channel nam Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**link-mode** {**balanced** | **unbalanced**}	Configures the link-mode as either balanced or unbalanced. unbalanced–Refers to a data transfer initiated from the master. balanced–Refers to either a master or slave data transfer.
Step 5	**link-addr-size** {**none** | **one** | **two**}	Defines the link address size in octets.
Step 6	**bind-to-interface serial** *slot/port/interface*	Defines the IR8340 serial interface on which the system sends its T101 protocol traffic. *slot* –Value of 0 *port* –Value of 2

	Command or Action	Purpose
		interface –Value of 0
Step 7	**exit**	Ends configuration of the channel and exits the channel configuration mode. Saves all settings.
Step 8	**session** *session_name*	Enters the session configuration mode and assigns a name to the session.
Step 9	**attach-to-channel** *channel_name*	Attaches the session to the channel. Enter the same channel name that you entered in Step 3 . *channel_name* –Identifies the channel.
Step 10	**common-addr-size** {**one** | **two** | **three**}	Defines the common address size in octets.
Step 11	**cot size** {**one** | **two** | **three**}	Defines the cause of transmission such as spontaneous or cyclic data schemes in octets.
Step 12	**info-obj-addr-size** {**one** | **two** | **three**}	Defines the information object element address size in octets.
Step 13	**link-addr-size** {**one** | **two** | **three**}	Defines the link address size in octets.
Step 14	**link-addr** *link_address*	Refers to the link address of the RTU.  **Note** The link address entered he *link_address* –Range of 0-65535.
Step 15	**exit**	Exits the session configuration mode.
Step 16	**sector** *sector_name*	Enters the sector configuration mode and assigns a name to the sector for the RTU. *sector_name* –Identifies the sector.
Step 17	**attach-to-session** *session_name*	Attaches the RTU sector to the session. Enter the same session name that you entered in Step 9 . *session_name*- Identifies the session.
Step 18	**asdu-addr** *asdu_address*	Refers to the ASDU structure address of the RTU.
Step 19	**exit**	Exits the sector configuration mode.
Step 20	**exit**	Exits the protocol configuration mode.

EXAMPLE

This example shows how to configure the parameters for the T101 protocol stack.

```
router# configure terminal
router(config)# scada-gw protocol t101
router(config-t101)# channel rtu_channel

router(config-t101-channel)# link-mode unbalanced
router(config-t101-channel)# link-addr-size one
router(config-t101-channel)# bind-to-interface serial 0/2/0
router(config-t101-channel)# exit
router(config-t101)# session rtu_session
router(config-t101-session)# attach-to-channel rtu_channel
router(config-t101-session)# common-addr-size two
router(config-t101-session)# cot-size one
router(config-t101-session)# info-obj-addr-size two
router(config-t101-session)# link-addr 3
router(config-t101-session)# exit
router(config-t101)# sector rtu_sector
router(config-t101-sector)# attach-to-session rtu_session
router(config-t101-sector)# asdu-addr 3
router(config-t101-sector)# exit
router(config-t101)# exit
router(config)#
```

Configuring the T104 Protocol Stack

Follow the steps below for each Control Center that you want to connect to over a T104 protocol.

Before you begin

Ensure that you have gathered all the required configuration information. (See Prerequisites, on page 249.)

Enable the serial port and SCADA encapsulation. (See Enabling the IR8340 Serial Port and SCADA Encapsulation, on page 250.)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**scada-gw protocol t104**	Enters the configuration mode for the T104 protocol.
Step 3	**channel** *channel_name*	Enters the channel configuration mode for the T104 protocol. *channel_name* –Identifies the channel on which the router communicates with the Control Center.

	Command or Action	Purpose
		Note When the entered channel n Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**k-value** *value*	Sets the maximum number of outstanding Application Protocol Data Units (APDUs) for the channel. **Note** An APDU incorporates the *value* –Range of values from 1 to 32767. Default value is 12 APDUs.
Step 5	**w-value** *value*	Sets the maximum number of APDUs for the channel. *value* –Range of values from 1 to 32767. Default value is 8 APDUs.
Step 6	**t0-timeout** *value*	Defines the t0-timeout value for connection establishment of the T104 channel.
Step 7	**t1-timeout** *value*	Defines the t1-timeout value for send or test APDUs on the T104 channel.
Step 8	**t2-timeout** *value*	Defines the t2-timeout value for acknowledgements when the router receives no data message. **Note** The t2 value must always b
Step 9	**t3-timeout** *value*	Defines the t3-timeout value for sending s-frames in case of a long idle state on the T104 channel. **Note** The t3 value must always b

	Command or Action	Purpose
Step 10	**tcp-connection {0|1} local-port** {*port_number* | **default} remote-ip** {*A.B.C.D* | *A.B.C.D/LEN* | **any} [vrf** *WORD***]**	In a configuration where there are redundant Control Centers, sets the connection value for the secondary Control Center as defined on the primary Control Center. *port-number* –value between 2000 and 65535. default–value of 2404. *A.B.C.D* –single host. *A.B.C.D/nn* –subnet A.B.C.D/LEN. any–any remote hosts 0.0.0.0/0. WORD–VRF name.
Step 11	**exit**	Exits the channel configuration mode.
Step 12	**session** *session_name*	Enters the session configuration mode and assigns a name to the session. s*ession_name* –Use the same name that you assigned to the channel in Step 3.
Step 13	**attach-to-channel** *channel_name*	Defines the name of the channel that transports the session traffic.
Step 14	**exit**	Exits the session configuration mode.
Step 15	**sector** *sector_name*	Enters the sector configuration mode and assigns a name to the sector for the Control Center.
Step 16	**attach-to-session** *session_name*	Attaches the Control Center sector to the channel. *session_name* –Use the same name that you assigned to the channel in Step 3.
Step 17	**asdu-addr** *asdu_address*	Refers to the ASDU structure address. Value entered here must match the ASDU value on the RTU. *asdu_address –asdu_address* –Value of 1 or 2.
Step 18	**map-to-sector** *sector_name*	Maps the Control Center (T104) sector to the RTU (T101) sector.
Step 19	Return to Step 1.	Repeat all steps in this section for each Control Center active in the network.

EXAMPLE

This example shows how to configure the parameters for the T104 protocol stack on *Control Center 1* and *Control Center 2,* both of which are configured as *masters* , and how to map the T104 sector to the T101 sector.

To configure Control Center 1 (*cc_master1*), enter the following commands.

```
router# configure terminal
router(config)# scada-gw protocol t104
router(config-t104)# channel cc_master1
router(config-t104-channel)# k-value 12
router(config-t104-channel)# w-value 8
router(config-t104-channel)# t0-timeout 30
router(config-t104-channel)# t1-timeout 15
router(config-t104-channel)# t2-timeout 10
router(config-t104-channel)# t3-timeout 30
router(config-t104-channel)# tcp-connection 0 local-port 2050 remote-ip 209.165.200.225
router(config-t104-channel)# tcp-connection 1 local-port 2051 remote-ip 209.165.201.25
router(config-t104-channel)# exit
router(config-t104)# session cc_master1
router(config-t104-session)# attach-to-channel cc_master1

router(config-t104-session)# exit
router(config-t104)# sector cc_master1-sector
router(config-t104-sector)# attach-to-session cc_master1
router(config-t104-sector)# asdu-adr 3
router(config-t104-sector)# map-to-sector rtu_sector
router(config-t104)# exit
router(config)#
```

To configure Control Center 2 (*cc_master2*), enter the following commands.

```
router(config)# scada-gw protocol t104
router(config-t104)# channel cc_master2
router(config-t104-channel)# k-value 12
router(config-t104-channel)# w-value 8
router(config-t104-channel)# t0-timeout 30
router(config-t104-channel)# t1-timeout 15
router(config-t104-channel)# t2-timeout 10
router(config-t104-channel)# t3-timeout 30
router(config-t104-channel)# tcp-connection 0 local-port 2060 remote-ip 209.165.201.237
router(config-t104-channel)# tcp-connection 1 local-port 2061 remote-ip 209.165.200.27
router(config-t104-channel)# exit
router(config-t104)# session
 cc_master2
router(config-t104-session)# attach-to-channel cc_master2

router(config-t104-session)# exit
router(config-t104)# sector cc_master2-sector
router(config-t104-sector)# attach-to-session cc_master2
router(config-t104-sector)# asdu-adr 3
router(config-t104-sector)# map-to-sector rtu_sector
router(config-t104-sector)# exit
router(config-t104)# exit
router(config)#
```

Configuration Example

The following example shows how to configure the serial port interface for T101 connection, configure T101 and T104 protocol stacks, and starts the Protocol Translation Engine on the IR8340.

```
router# configure terminal
router(config)# interface serial 0/2/0
router (config-if)# no shutdown
router (config-if)# encapsulation scada
router (config-if)# exit
router(config)# scada-gw protocol t101
router(config-t101)# channel rtu_channel

router(config-t101-channel)# link-mode unbalanced
router(config-t101-channel)# link-addr-size one
router(config-t101-channel)# bind-to-interface serial 0/2/0
router(config-t101-channel)# exit
router(config-t101)# session rtu_session
router(config-t101-session)# attach-to-channel rtu_channel
router(config-t101-session)# common-addr-size two
router(config-t101-session)# cot-size one
router(config-t101-session)# info-obj-addr-size two
router(config-t101-session)# link-addr 3
router(config-t101-session)# exit
router(config-t101)# sector rtu_sector
router(config-t101-sector)# attach-to-session rtu_session
router(config-t101-sector)# asdu-addr 3
router(config-t101-sector)# exit
router(config-t101)# exit
router(config)# scada-gw protocol t104
router(config-t104)# channel cc_master1
router(config-t104-channel)# k-value 12
router(config-t104-channel)# w-value 8
router(config-t104-channel)# t0-timeout 30
router(config-t104-channel)# t1-timeout 15
router(config-t104-channel)# t2-timeout 10
router(config-t104-channel)# t3-timeout 30
router(config-t104-channel)# tcp-connection 0 local-port 2050 remote-ip any
router(config-t104-channel)# tcp-connection 1 local-port 2051 remote-ip any
router(config-t104-channel)# exit
router(config-t104)# session cc_master1
router(config-t104-session)# attach-to-channel cc_master1

router(config-t104-session)# exit
router(config-t104)# sector cc_master1-sector
router(config-t104-sector)# attach-to-session cc_master1
router(config-t104-sector)# asdu-adr 3
router(config-t104-sector)# map-to-sector rtu_sector
router(config-t104)# exit

router(config-t104)# session cc_master2
router(config-t104-session)# attach-to-channel cc_master2

router(config-t104-session)# exit
router(config-t104)# sector cc_master2-sector
router(config-t104-sector)# attach-to-session cc_master2
router(config-t104-sector)# asdu-adr 3
router(config-t104-sector)# map-to-sector rtu_sector
router(config-t104-sector)# exit
```

```
router(config-t104)# exit
router(config)# scada-gw enable
```

This example configures end-to-end communication between Control Centers and RTUs within a SCADA system using the DNP3 protocol stacks and starts the Protocol Translation Engine on the IR8340:

```
router# configure terminal
router(config)# interface serial 0/2/0
router (config-if)# no shutdown
router (config-if)# encapsulation scada
router (config-if)# exit
router(config)# scada-gw protocol dnp3-serial
router(config-dnp3s)# channel rtu_channel
router(config-dnp3s-channel)# bind-to-interface serial 0/2/0
router(config-dnp3s-channel)# link-addr source 3
router(config-dnp3s-channel)# unsolicited-response enable
router(config-dnp3s-channel)# exit
router(config-dnp3s)# session rtu_session
router(config-dnp3s-session)# attach-to-channel rtu_channel
router(config-dnp3s-session)# link-addr dest 3
router(config-dnp3s-session)# exit
router(config-dnp3s)# exit
router(config)# scada-gw protocol dnp3-ip
router(config-dnp3n)# channel cc_channel
router(config-dnp3n-channel)# link-addr dest 3
router(config-dnp3n-channel)# tcp-connection local-port default remote-ip any
router(config-dnp3n-channel)# exit
router(config-dnp3n)# session cc_session
router(config-dnp3n-session)# attach-to-channel cc_channel
router(config-dnp3n-session)# link-addr source 3
router(config-dnp3n-session)# map-to-session rtu_session
router(config-dnp3n)# exit
router(config)# exit
router(config)# scada-gw enable
```

Note IOA addresses obtained from T101 side are sent to T104 side without any modification by the SCADA Gateway.

Configuring the DNP3 Protocol Stacks

You can configure the DNP3 serial and DNP3 IP protocol stacks, which allow end-to-end communication between Control Centers and RTUs within a SCADA system.

Configuring DNP3 Serial

Configure the channel and session parameters for the DNP serial communication with an RTU.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**scada-gw protocol dnp3-serial**	Enters configuration mode for the DNP3 serial protocol.
Step 3	**channel** *channel_name*	Enters channel configuration mode for the DNP3 serial protocol. *channel_name* –Identifies the channel on which the router serial port communicates to the RTU. Note: When the entered channel name does not already exist, the router creates a new channel Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**bind-to-interface serial 0/2/0**	Defines the router async interface on which the system sends its DNP3 protocol traffic.
Step 5	**link-addr source** *source_address*	Refers to the link address of the master. *source_address* –Range of values from 1 to 65535.
Step 6	**unsolicited-response enable**	(Optional) Allows unsolicited responses. Entering the **no** form of this command disables unsolicited responses. The default is disabled.
Step 7	**exit**	Ends configuration of the channel and exits channel configuration mode. Saves all settings.
Step 8	**session** *session_name*	Enters session configuration mode and assigns a name to the session. Note: When the entered session name does not already exist, the router creates a new session. Entering the **no** form of this command deletes an existing session.
Step 9	**attach-to-channel** *channel_name*	Attaches the session to the channel. Note: Enter the same channel name that you entered in Step 3 above *channel_name* –Identifies the channel.
Step 10	**link-addr dest** *destination_address*	Refers to the link address of the slave. *destination_address* –Range of values from 1 to 65535.

	Command or Action	Purpose
Step 11	**exit**	Exits session configuration mode.
Step 12	**exit**	Exits protocol configuration mode.

EXAMPLE

This example shows how to configure the parameters for the DPN3-serial protocol stack:

```
router# configure terminal
router(config)# scada-gw protocol dnp3-serial
router(config-dnp3s)# channel rtu_channel
router(config-dnp3s-channel)# bind-to-interface serial 0/2/0
router(config-dnp3s-channel)# link-addr source 3
router(config-dnp3s-channel)# unsolicited-response enable
router(config-dnp3s-channel)# exit
router(config-dnp3s)# session rtu_session
router(config-dnp3s-session)# attach-to-channel rtu_channel
router(config-dnp3s-session)# link-addr dest 3
router(config-dnp3s-session)# exit
router(config-dnp3s)# exit
router(config)#
```

Configuring DNP3 IP

Follow the steps below for the Control Center that you want to connect to over DNP3 IP. For redundancy, you can create multiple connections that share the same session configuration under the same session.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**scada-gw protocol dnp3-ip**	Enters configuration mode for the DNP-IP protocol.
Step 3	**channel** *channel_name*	Enters channel configuration mode for the DNP-IP protocol. *channel_name* –Identifies the channel on which the router communicates with the Control Center. Note: When the entered channel name does not already exist, the router creates a new channel. Entering the **no** form of this command deletes an existing channel. However, all sessions must be deleted before you can delete a channel.
Step 4	**link-addr dest** *destination_address*	Refers to the link address of the master.

	Command or Action	Purpose
		destination_address –Range of values from 1 to 65535.
Step 5	**send-unsolicited-msg enable**	(Optional) Allow unsolicited messages. The default is enabled.
Step 6	**tcp-connection local-port** [**default** | *local_port*] **remote-ip** [**any** | *remote_ip* | *remote_subnet*]	Configures the local port number and remote IP address for the TCP connection: • **default**–20000. • *local_port* –Range of values from 2000 to 65535. • any–Any remote hosts 0.0.0.0/0 • *remote_ip* –Single host: A.B.C.D • *remote_subnet* –Subnet: A.B.C.D/LEN If remote_subnet is specified, when two channels have the same local ports, the remote subnets cannot overlap each other. Note: Every <local-port, remote-ip> must be unique per channel. If remote_subnet is specified, when two channels have the same local ports, the remote subnets cannot overlap each other.
Step 7	**exit**	Exits channel configuration mode.
Step 8	**session** *session_name*	Enters session configuration mode and assigns a name to the session. Note: When the entered session name does not already exist, the router creates a new session. Entering the **no** form of this command deletes an existing session.
Step 9	**attach-to-channel** *channel_name*	Attaches the session to the channel. Enter the same channel name that you entered in Step 3. *channel_name* –Identifies the channel.
Step 10	**link-addr source** *source_address*	Refers to the link address of the slave. *source_address* –Value of 1-65535.
Step 11	**map-to-session** *session_name*	Maps the dnp3-ip session to an existing dnp3-serial session.

	Command or Action	Purpose
		Note: One dnp3-ip session can be mapped to only one dnp3-serial session.
Step 12	**exit**	Exits session configuration mode.
Step 13	**exit**	Exits protocol configuration mode.

EXAMPLE

This example shows how to configure the DNP3 IP parameters:

```
router# configure terminal
router(config)# scada-gw protocol dnp3-ip
router(config-dnp3n)# channel cc_channel
router(config-dnp3n-channel)# link-addr dest 3
router(config-dnp3n-channel)# tcp-connection local-port default remote-ip any
router(config-dnp3n-channel)# exit
router(config-dnp3n)# session cc_session
router(config-dnp3n-session)# attach-to-channel cc_channel
router(config-dnp3n-session)# link-addr source 4
router(config-dnp3n-session)# map-to-session rtu_session
router(config-dnp3n)# exit
router(config)# exit
```

Starting and Stopping the Protocol Translation Engine

You must start the Protocol Translation Engine to use Protocol Translation on the IR8340.

Starting–After enabling SCADA encapsulation on the IR8340 serial port and configuring the T101 and T104 protocols on the IR8340, you can start the Protocol Translation Engine.

Stopping–Before you can make any configuration changes to Protocol Translation on the IR8340 with an active Protocol Translation Engine, you must stop the engine.

Before you begin

Before **starting** the Protocol Translation Engine on the router for the **first time**, make sure you complete the following items:

Enabling the IR8340 Serial Port and SCADA Encapsulation, on page 250

Configuring T101 and T104 Protocol Stacks, on page 251

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**[no] scada-gw enable**	Starts (**scada-gw enable**) or stops (**no scada-gw enable**) the Protocol Translation Engine on the IR8340.

EXAMPLE

To start the protocol translation engine on the router, enter the following commands:

```
router# configure terminal
router(config)# scada-gw enable
```

To stop the protocol translation engine on the router, enter the following commands:

```
router# configure terminal
router(config)# no
 scada-gw enable
```

Verifying Configuration

Command	Purpose
show running-config	Shows the configuration of the router including active features and their settings.
show scada database	Displays details on the SCADA database.
show scada statistics	Shows statistics for the SCADA gateway, including the number of messages sent and received, timeouts, and errors.
show scada tcp	Displays TCP connections associated with the SCADA gateway.

This example shows the output from the show scada tcp and show scada statistics commands:

```
Router#show scada tcp
DNP3 network channel [test]: 4 max simultaneous connections
conn: local-ip: 3.3.3.21 local-port 20000 remote-ip 3.3.3.15 data-socket
1
Total:
1 current client connections
0 total closed connections

Router#show scada statistics
DNP3 network Channel [test]:
5 messages sent, 2 messages received
0 timeouts, 0 aborts, 0 rejections
2 protocol errors, 2 link errors, 0 address errors
DNP3 serial Channel [test]:
152 messages sent, 152 messages received
1 timeouts, 0 aborts, 0 rejections
0 protocol errors, 0 link errors, 0 address errors

Router#show scada database
------------ Scada Gateway Database ----------------------
Configuration Root
   State -- 0x3
   MaxEvents -- 0x258
   Retry -- 0x3
   DegradedFrequency -- 0x2710
   Timeout -- 0x2710
   PointsAvailableToAlloc -- 0x470
   From Master
      IEC 104
```

```
Line(2) status:DOWN
   Link(2) status:DOWN InUse(0)
   Request(-99):NOT running
      function(1)id(1)Triggered(0)MustRun(0)Attempt(4)diff(3604)Freq(100)
      RTU(101) status:DOWN InUse(0) Online(0)
         Object(Single Input)
            DI(100)
               Value -- 0x0
               Quality -- 0x4
            DI(101)
               Value -- 0x0
               Quality -- 0x4
            DI(102)
               Value -- 0x0
               Quality -- 0x4
```

Debug Commands

This section lists some debug commands that are helpful when troubleshooting.

- SCADA device to trace debug commands:

```
#debug scada device ?
  network   Network device to trace
  none      No device to trace
  serial    Serial device to trace

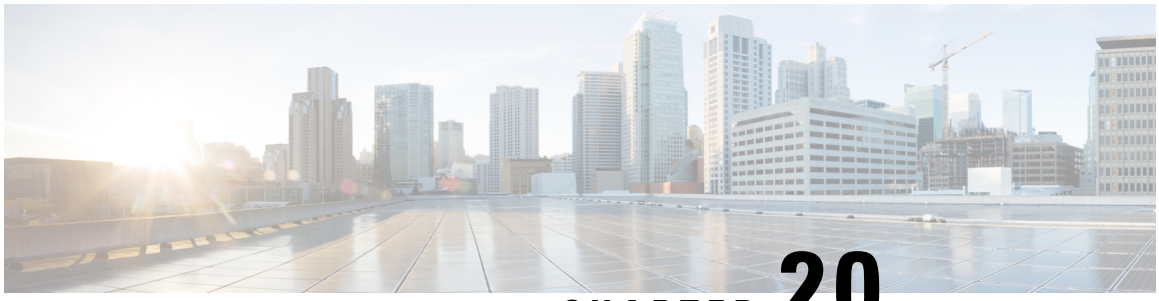
#debug scada device network ?
  <0-65535>   Address of device to trace

#debug scada device none ?
  <cr>  <cr>

#debug scada device serial ?
  <0-65535>   Address of device to trace
```

- SCADA function level debug commands:

```
#debug scada function ?
  analog          analog ouput trace
  clock           clock trace
  config          config trace
  control         control trace
  datalink        datalink trace
  driver-event    Driver event debug
  driver-packet   Driver Packet debug
  error           error trace
  event           event trace
  file            file trace
  freeze          freeze trace
  general         general debug
  physical        physical trace
  poll            poll trace
  protocol        protocol trace
  request         request trace
  stack           stack trace
  status          status trace
  tcp-event       TCP event debug
  timer           timer trace
  transport       transport trace
  umode           umode trace
```



CHAPTER 20

Raw Socket Transport

Raw Socket Transport transports streams of characters from one serial interface to another over an IP network for utility applications.

This document describes Raw Socket Transport for the IR8340 and provides a reference section describing the Raw Socket Transport commands.

This document includes the following sections:

- Information About Raw Socket Transport, on page 267
- Prerequisites, on page 270
- Guidelines and Limitations, on page 270
- Default Settings, on page 270
- Configuring Raw Socket Transport, on page 270
- Verifying Configuration, on page 276
- Configuration Example, on page 277
- Show Line Details for Configuring Raw-TCP/UDP, on page 280
- Raw-Socket Show and Debug Commands, on page 281

Information About Raw Socket Transport

Raw Socket is a method for transporting serial data through an IP network. The feature can be used to transport Supervisory Control and Data Acquisition (SCADA) data from Remote Terminal Units (RTUs). This method is an alternative to the Block Serial Tunnel (BSTUN) protocol.

Raw Socket Transport supports TCP or UDP as the transport protocol. An interface can be configured to use either protocol but not both at the same time. TCP transport is suitable for applications such as control applications that require acknowledged and sequenced delivery of data. For latency-sensitive applications such as line SEL relays, UDP transport provides faster transport of serial data than TCP.

Raw Socket Transport supports the following for the asynchronous serial interface:

- TCP as the transport protocol, with built-in auto TCP connection retry mechanism.
- Up to 32 TCP sessions and 32 UDP sessions.
- Interface configuration as a server, client, or a combination of both.
- One server interface, but multiple clients.

- VRF-awareness, which enables the router to send Raw Socket Transport traffic to a server host connected through a Virtual Private Network (VPN) Virtual Routing and Forwarding (VRF) interface.

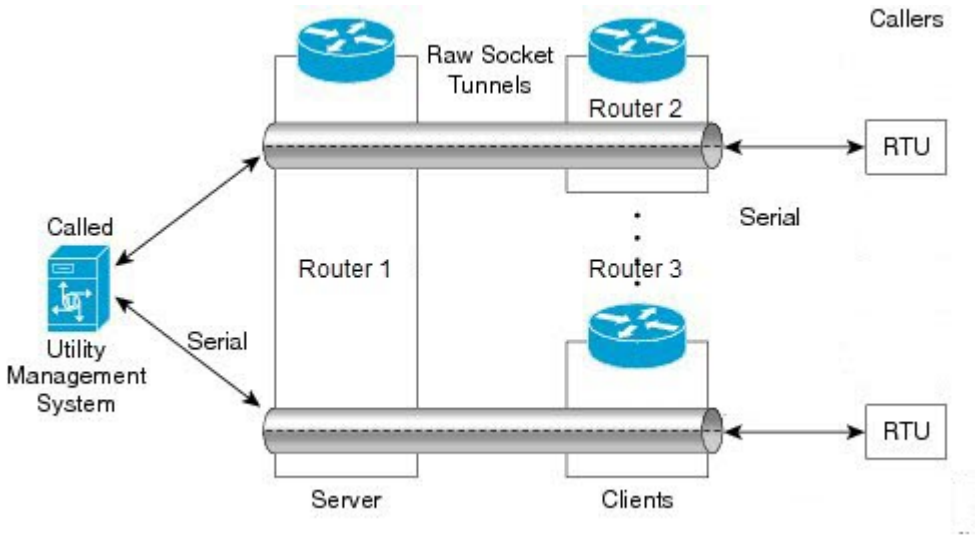
This section includes the following topics:

TCP Transport

TCP Raw Socket transport uses a client-server model. At most one server and multiple clients can be configured on a single asynchronous serial line. In client mode, the IR8340 can initiate up to 32 TCP sessions to Raw Socket servers, which can be other IR8340 routers or third-party devices.

The following figure shows a sample Raw Socket TCP configuration. In this example, serial data is transferred between RTUs and a utility management system across an IP network that includes several IR8340 routers. One IR8340 router (Router 1) acts as a Raw Socket server, listening for TCP connection requests from the other IR8340 routers (Router 2 and Router 3), which are configured as Raw Socket clients.

A Raw Socket client receives streams of serial data from the RTUs and accumulates this data in its buffer, then places the data into packets, based on user-specified packetization criteria. The Raw Socket client initiates a TCP connection with the Raw Socket server and sends the packetized data across the IP network to the Raw Socket server, which retrieves the serial data from the packets and sends it to the serial interface, and on to the utility management system.



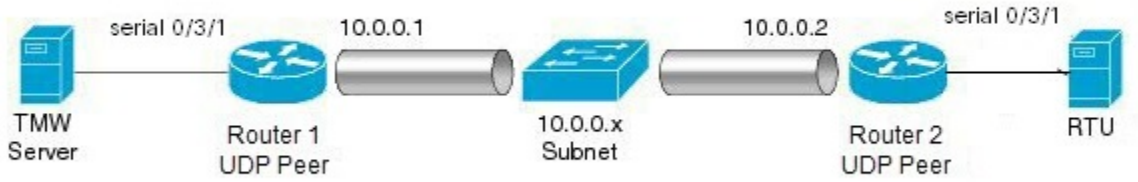
Note When you configure the serial link interface on the router as a server, the interface's peer is the serial link interface on the client router and vice versa.

UDP Transport

UDP transport uses a peer-to-peer model. Multiple UDP connections can be configured on an asynchronous serial line. IR8340 can support up to 32 UDP sessions.

The following figure shows a sample Raw Socket UDP configuration. In this example, serial data is transferred between RTUs and a utility management system across an IP network that includes two routers that are configured as Raw Socket UDP peers.

In this example, the Raw Socket UDP peer receives streams of serial data from the RTUs and accumulates this data in its buffer, then places the data into packets, based on user-specified packetization criteria. The Raw Socket UDP peer sends the packetized data across the IP network to the Raw Socket peer at the other end, which retrieves the serial data from the packets and sends it to the serial interface, and on to the utility management system.



Serial Data Processing

When the default serial protocol, Asynchronous Communication Protocol, is used, the streams of serial data received by a Raw Socket peer can be packetized based on the following criteria:

- **Packet length**–You can specify a packet length that triggers the IR8340 to transmit the serial data to the peer. Once the IR8340 collects this much data in its buffer, it packetizes the accumulated data and forwards it to the Raw Socket peer.
- **Packet-timer value**–The packet timer specifies the amount of time the IR8340 waits to receive the next character in a stream. If a character is not received by the time the packet timer expires, the data the IR8340 has accumulated in its buffer is packetized and forwarded to the Raw Socket peer.
- **Special character**–You can specify a character that will trigger the IR8340 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. When the special character (for example, a CR/LF) is received, the IR8340 packetizes the accumulated data and sends it to the Raw Socket peer.

See Configuring Common Raw Socket Line Options, on page 271 for information about configuring the processing options.

VRF-Aware Raw Socket

The VRF-aware Raw Socket Transport feature enables you to isolate Raw Socket traffic using a VRF for efficient management and control of serial data. After configuring a VRF, you can associate the serial interface configured for Raw Socket Transport with the VRF. See Raw Socket VRF, on page 279 for a configuration example.

Prerequisites

Determine how you want Raw Socket traffic transported in your network, including the network devices and interfaces to use, how the router packetizes the serial data, and whether to use VRF.

Guidelines and Limitations

The guidelines and limitations are as given:

- Typically, UDP traffic is blocked by firewalls in the network. If the network has such firewalls, make sure to configure pinholes to allow the raw socket UDP traffic.
- The **access-class <acl> in** command is unsupported for serial raw-socket line configurations on IR platforms using IOS-XE.
- When an access list is applied to a raw-socket-enabled serial interface, traffic is processed according to the configured permit and deny entries.

Default Settings

Feature	Default Setting
Raw Socket Transport	Disabled.
Packet length	No packet length is configured.
Serial Protocol	Asynchronous Communication Protocol
Packet timeout	15 ms.
Special character	No special character is configured.
Raw Socket mode	Best-effort mode is off, not supported on the IR8340.
TCP idle timeout	5 minutes.

Configuring Raw Socket Transport

This section includes the following topics:

Enabling Raw Socket Transport on the Serial Interface

To enable Raw Socket Transport on the IR8340 router, you must first enable an asynchronous serial port and enable Raw Socket TCP or UDP encapsulation for that port.

Before you begin

Determine availability of the serial port on the IR8340.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface serial** *slot/subslot/port*	Enters the interface command mode for the serial interface.
Step 3	**physical-layer async**	Configure the serial interface to Asynchronous mode.
Step 4	**no ip address**	Disables IP processing on the interface.
Step 5	Do one of the following: • **encapsulation raw-tcp** • **encapsulation raw-udp**	Enables Raw Socket TCP encapsulation or UDP encapsulation for the serial port.

Example

```
router# configure terminal
router(config)# interface serial 0/3/2
router(config)# physical-layer async
router(config-if)# no ip address
router(config-if)# encapsulation raw-tcp
router(config-if)# exit
```

Configuring Common Raw Socket Line Options

You can configure options common to all connections on a line. The common options apply to both TCP and UDP.

Before you begin

Enable Raw Socket Transport as described in Enabling Raw Socket Transport on the Serial Interface, on page 270.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**line 0**/*slot* /*port*	Enters line command mode for the serial slot/port.

	Command or Action	Purpose
Step 3	**raw-socket packet-length** *length*	Specifies the packet size that triggers the IR8340 to transmit the data to the peer. When the IR8340 accumulates this much data in its buffer, it packetizes the data and forwards it to the Raw Socket peer. *length*— 2 to 1400 bytes. By default, the packet-length trigger is disabled.
Step 4	**raw-socket packet-timer** *timeout*	Specifies the maximum time in milliseconds the IR8340 waits to receive the next character in a stream. If a character is not received by the time the packet-timer expires, the accumulated data is packetized and forwarded to the Raw Socket peer. *timeout* —3 to 1000 ms. The default is 15 ms.
Step 5	**raw-socket spec-char** *ascii_char*	Specifies a character that will trigger the IR8340 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. *ascii_char*— 0 to 255. By default, the special character trigger is disabled.

Example

```
router# configure terminal
router(config)# line 0/3/2
router(config-line)# raw-socket packet-length 32
router(config-line)# raw-socket packet-timer 500
router(config-line)# raw-socket special-char 3
router(config-line)# parity even
router(config-line)# stopbits 1
router(config-line)# speed 9600
```

What to do next

Use the **no** form of these commands to return to the default values.

Configuring Raw Socket TCP

After enabling Raw Socket TCP encapsulation, you configure the TCP server and/or clients.

Configuring the Raw Socket TCP Server

Before you begin

Enable a serial port and Raw Socket TCP encapsulation for that port, as described in Enabling Raw Socket Transport on the Serial Interface, on page 270.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**line 0**/*slot* /*port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket tcp server** *port* [*ip_address*]	Starts the Raw Socket Transport TCP server for an asynchronous line interface. In Raw Socket server mode, the IR8340 listens for incoming connection requests from Raw Socket clients. *port* –Port number the server listens on. *ip_address* –(Optional) Local IP address on which the server listens for connection requests.
Step 4	**raw-socket tcp idle-timeout** *session_timeout*	Sets the Raw Socket Transport TCP session timeout for the asynchronous line interface. If no data is transferred between the client and server over this interval, then the TCP session closes. The client then automatically attempts to reestablish the TCP session with the server. This timeout setting applies to all Raw Socket Transport TCP sessions under this particular line. *session_timeout* –Currently configured session idle timeout in minutes. The default is 5 minutes.

Example

```
router# configure terminal

router(config)# line 0/3/2
router(config-line)# raw-socket tcp server 4000 10.0.0.1
router(config-line)# raw-socket tcp idle-timeout 10
router(config-line)# exit
router(config)#
```

What to do next

To remove a Raw Socket TCP server, use the **no raw-socket tcp server** command.

Configuring the Raw Socket TCP Client

Before you begin

Enable a serial port and Raw Socket TCP encapsulation for that port, as described in Enabling Raw Socket Transport on the Serial Interface, on page 270.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**line 0**/*slot* /*port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket tcp client** *dest_ip_address dest_port* [*local_ip_address*] [*local_port*]	Specifies settings for Raw Socket Transport TCP client sessions. *dest_ip_address* –Destination IP address of the remote Raw Socket server. *dest_port* –Destination port number to use for the TCP connection to the remote server. *local_ip_address* –(Optional) Local IP address that the client can also bind to. *local_port* –(Optional) Local port number that the client can also bind to.
Step 4	**raw-socket tcp idle-timeout** *session_timeout*	Sets the Raw Socket Transport TCP session timeout for the asynchronous line interface. If no data is transferred between the client and server over this interval, then the TCP session is closed. The client then automatically attempts to reestablish the TCP session with the server. This timeout setting applies to all Raw Socket Transport TCP sessions under this particular line. *session_timeout* –Currently configured session idle timeout in minutes. The default is 5 minutes.
Step 5	**raw-socket tcp keepalive** *interval*	Sets the Raw Socket Transport TCP session keepalive interval for the asynchronous line interface. The router sends keepalive messages based on the configured interval. You may need to configure this interval, for example, when

	Command or Action	Purpose
		sending raw TCP traffic over a cellular interface. *interval* –Currently configured keepalive interval in seconds. Range is 1-864000 seconds. The default is 1 second.

Example

This example shows how to configure a Raw Socket TCP client for an asynchronous serial line. The IR8340 (router), serving as a Raw Socket client, initiates TCP sessions with a Raw Socket server and forwards packetized serial data to it. The router collects streams of serial data in its buffer; when it accumulates 827 bytes in its buffer, the router packetizes the data and forwards it to the Raw Socket server. If the router and the Raw Socket server do not exchange any data for 10 minutes, then the TCP session with the Raw Socket server closes, and the router attempts to reestablish the session with the Raw Socket server.

```
router# configure terminal

router(config)# line 0/3/2
router(config-line)# raw-socket tcp client 10.0.0.1 4000
router(config-line)# raw-socket packet-length 827
router(config-line)# raw-socket tcp idle-timeout 10
router(config-line)# exit
router(config)#
```

What to do next

To remove a Raw Socket TCP client, use the **no raw-socket tcp client** command.

Configuring a Raw Socket UDP Peer-to-Peer Connection

After enabling Raw Socket UDP encapsulation and the common line options, you configure the Raw Socket UDP peer-to-peer connection. The local port on one end of the connection should be the destination port on the other end.

Before you begin

Enable a serial port and Raw Socket UDP encapsulation for that port, as described in Enabling Raw Socket Transport on the Serial Interface, on page 270.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**line 0**/*slot* /*port*	Enters line command mode for the serial slot/port.

	Command or Action	Purpose
Step 3	**raw-socket udp connection** *dest_ip_address dest_port local_port* [*local_ip_address*]	Specifies settings for Raw Socket Transport UDP connections. *dest_ip_address* –Destination IP address to use for the UDP connection. *dest_port* –Destination port number to use for the UDP connection. *local_port* –Local port number for the UDP connection. *local_ip_address* –(Optional) Local IP address for the UDP connection.

Example

This example shows how to configure a Raw Socket UDP connection between router A (local IP address 192.168.0.8) and router B (local IP address 192.168.0.2).

Router A

```
router# configure terminal
router(config)# line 0/2/0
router(config-line)# raw-socket udp connection 192.168.0.2 5000 7000
router(config-line)# exit
router(config)#
```

Router B

```
router# configure terminal
router(config)# line 0/2/0
router(config-line)# raw-socket udp connection 192.168.0.8 7000 5000
router(config-line)# exit
router(config)#
```

What to do next

To remove a Raw Socket UDP connection, use the **no raw-socket udp connection** command.

Verifying Configuration

Command	Purpose
show running-config	Shows the configuration of the IR8340, including those features that are active and their settings.
show raw-socket tcp detail	Displays information about Raw Socket Transport TCP activity.
show raw-socket tcp sessions	Displays information about Raw Socket Transport TCP sessions.

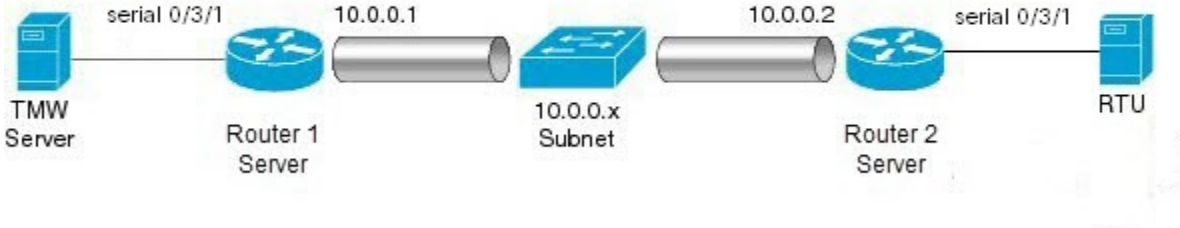
Command	Purpose
show raw-socket tcp statistics	Displays Raw Socket Transport TCP statistics for each asynchronous serial line.
show raw-socket udp detail	Displays information about Raw Socket Transport UDP activity.
show raw-socket udp sessions	Displays information about Raw Socket Transport UDP sessions.
show raw-socket udp statistics	Displays Raw Socket Transport UDP statistics for each asynchronous serial line.
clear raw-socket statistics	Clears Raw Socket Transport statistics for a specific TTY interface or for all asynchronous serial lines.

Configuration Example

The following sections include Raw Socket Transport configuration examples:

Raw Socket TCP

The following example shows a Raw Socket Transport configuration in which an IR8340 router (Router 1) acts as the server, and another IR8340 (Router 2) acts as the client.



The following table displays the configuration of the server and client IR8340s highlighted in the above figure:

IR8340 Server Configuration	IR8340 Client Configuration
... interface serial 0/3/1 physical-layer async no ip address **encapsulation raw-tcp** ! ... line 0/3/1 **raw-socket tcp server 5000 10.0.0.1** **raw-socket packet-timer 3** **raw-socket tcp idle-timeout 5** ...	... interface serial 0/3/1 physical-layer async no ip address **encapsulation raw-tcp** ! interface serial 0/3/2 physical-layer async no ip address **encapsulation raw-tcp** ! ... line 0/3/1 **raw-socket tcp client 10.0.0.1 5000 10.0.0.2 9000** **raw-socket packet-length 32** **raw-socket tcp idle-timeout 5** line 0/3/2 **raw-socket tcp client 10.0.0.1 5000 10.0.0.2 9001** **raw-socket packet-length 32** **raw-socket tcp idle-timeout 5**

Raw Socket UDP

This example shows the configuration for a Raw Socket UDP connection between two IR8340 routers:

From Router1

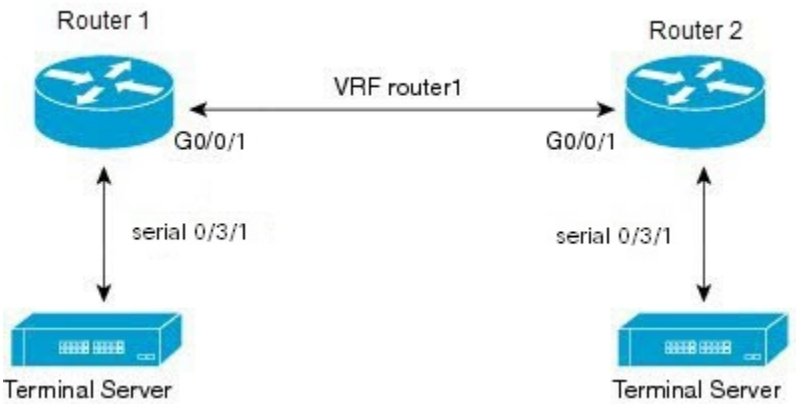
```
interface GigabitEthernet 0/1/1
ip address 192.168.0.8 255.255.255.0
duplex auto
speed auto
interface serial 0/3/2
physical-layer async
no ip address
encapsulation raw-udp
line 0/3/2
raw-socket udp connection 192.168.0.2 4000 4000
```

From Router2

```
interface GigabitEthernet0/1/2
ip address 192.168.0.2 255.255.255.0
load-interval 60
duplex auto
speed auto
no keepalive
interface serial 0/3/2
physical-layer async
no ip address
encapsulation raw-udp
line 0/3/2
raw-socket udp connection 192.168.0.8 4000 4000
```

Raw Socket VRF

The following example shows a Raw Socket VRF configuration in which two routers, configured for Raw Socket Transport, connect through a VRF. Router1 is an IR8340, serves as the Raw Socket TCP server, and Router2 is an IR8340 serves as the Raw Socket TCP client.



Following are the configurations of Router1 and Router2 as shown in the above figure:

Router1 Configuration

Defining VRF on the router:

```
vrf definition router1
 rd 100:1
 route-target export 100:3
 route-target import 100:3
 !
 address-family ipv4
 exit-address-family
```

Applying VRF configuration on the interface:

```
interface GigabitEthernet0/0/1
 vrf forwarding router1
 ip address 100.100.100.2 255.255.255.0
 duplex auto
 speed auto
```

Applying raw-tcp on the serial interface:

```
interface serial 0/3/1
 physical-layer async
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on the line:

```
line 0/3/1
 raw-socket tcp server 5000 4.4.4.4
```

Router2 Configuration

Defining VRF on the router:

```
vrf definition router1
 rd 100:1
 route-target export 100:3
 route-target import 100:3
 !
 address-family ipv4
 exit-address-family
```

Applying VRF configuration on the interface:

```
interface GigabitEthernet0/0/1
 vrf forwarding router1
 ip address 100.100.100.1 255.255.255.0
 duplex auto
 speed auto
```

Applying raw-tcp on the serial interface:

```
interface serial 0/3/1
 physical-layer async
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on line:

```
line 0/3/1
  raw-socket tcp client 4.4.4.4 5000
```

Show Line Details for Configuring Raw-TCP/UDP

The **show line** command shows all TTY line summary information. The output contains information about mapping between async interface and line number, the line speed, uses, noise, and so on. The line that begins with the asterisk "*" indicates that the line is in use.

By enabling the physical layer Async on the serial port, the corresponding line will be activated and it will be displayed in the **show line** output. In the following example, 0/3/0 and 0/3/4 are configured as async mode and displayed.

```
U1#show line
   Tty Line Typ     Tx/Rx    A Modem  Roty AccO AccI  Uses  Noise Overruns  Int
*     0    0 CTY              -    -      -    -    -     0      0    0/0      -
      1    1 AUX   9600/9600  -    -      -    -    -     0      0    0/0      -
  0/3/0   50 TTY   9600/9600  -    -      -    -    -     0     17    0/0    Se0/3/0
  0/3/4   54 TTY   9600/9600  -    -      -    -    -     0      0    0/0    Se0/3/4
    866  866 VTY              -    -      -    -    -     3      0    0/0      -
    867  867 VTY              -    -      -    -    -     0      0    0/0      -
    868  868 VTY              -    -      -    -    -     0      0    0/0      -
    869  869 VTY              -    -      -    -    -     0      0    0/0      -
    870  870 VTY              -    -      -    -    -     0      0    0/0      -
    871  871 VTY              -    -      -    -    -     0      0    0/0      -
    872  872 VTY              -    -      -    -    -     0      0    0/0      -
    873  873 VTY              -    -      -    -    -     0      0    0/0      -
    874  874 VTY              -    -      -    -    -     0      0    0/0      -
    875  875 VTY              -    -      -    -    -     0      0    0/0      -
    876  876 VTY              -    -      -    -    -     0      0    0/0      -
    877  877 VTY              -    -      -    -    -     0      0    0/0      -
    878  878 VTY              -    -      -    -    -     0      0    0/0      -
    879  879 VTY              -    -      -    -    -     0      0    0/0      -
    880  880 VTY              -    -      -    -    -     0      0    0/0      -
Line(s) not in async mode -or- with no hardware support:
```

```
2-49, 51-53, 55-865
U1#
```

Raw-Socket Show and Debug Commands

Use the following show commands for the verification:

- **show raw-socket {tcp|udp} session**
- **show raw-socket {tcp|udp} statistic**

Use the following commands to debug:

```
debug raw-socket ?
  driver      Driver level debug
  transport   Raw-socket over any transport

debug raw-socket driver ?
  event    Driver event trace
  packet   Driver packet trace

debug raw-socket driver event ?
  <cr>  <cr>

debug raw-socket driver packet ?
  line   Serial/Async line
  <cr>  <cr>

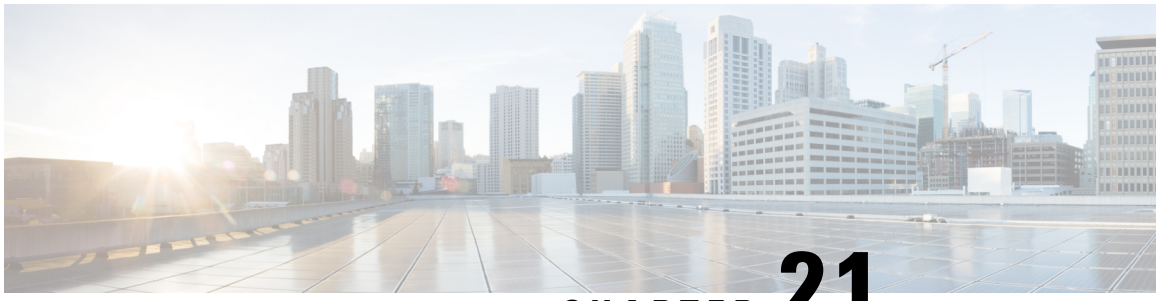
debug raw-socket driver packet line ?
  <0-726>  First Line range
  console  Primary terminal line
  tty      Terminal controller
  vty      Virtual terminal
  x/y/z    Slot/Subslot/Port for Modems

debug raw-socket transport ?
  event    TCP or UDP event trace
  packet   TCP or UDP packet trace

debug raw-socket transport event ?
  <cr>  <cr>

debug raw-socket transport packet ?
  line   Serial/Async line
  <cr>  <cr>

debug raw-socket transport packet line ?
  <0-726>  First Line range
  console  Primary terminal line
  tty      Terminal controller
  vty      Virtual terminal
  x/y/z    Slot/Subslot/Port for Modems
```



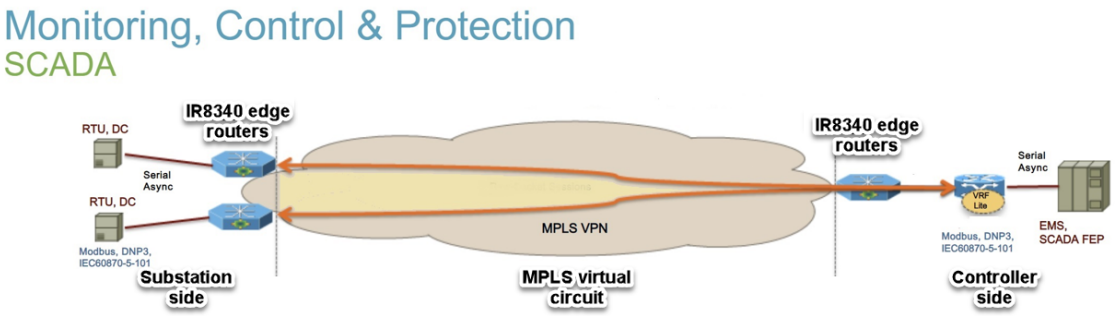
CHAPTER 21

Serial MPLS Pseudowire

- Serial MPLS pseudowire on IR8340, on page 283
- Data exchange between substation RTU and SCADA controller using MPLS pseudowire, on page 284
- Serial MPLS pseudowire deployment scenarios, on page 285
- Serial to serial MPLS pseudowire deployment configuration using CLI, on page 286
- Serial to raw socket (IP) MPLS pseudowire deployment configuration using CLI, on page 288
- Raw socket configuration using CLI, on page 289
- Verify serial to serial configurations using CLI, on page 290
- Verify Serial to raw socket (IP) configurations using CLI, on page 292
- Limitations and restrictions of current serial MPLS pseudowire release, on page 294

Serial MPLS pseudowire on IR8340

From IOS XE Release 17.17.1, you can transfer the serial async data using an MPLS label over a pseudowire between two IR8340 edge routers. In this scenario, one of the routers is located at the substation end and the other router is located at the controller end. Serial PW feature is configurable through CLI, Netconf/Yang and SDWAN/Vmanage interfaces. Oper Netconf/Yang model for this feature is not supported in 17.17.1 IOS-XE Release.



Following are the key networking elements in this scenario:

- Substation: This refers to a location within an IIoT utility setup where edge routers are present, typically involved in collecting data from Remote Terminal Units (RTUs).
- Controller: This is the location where the controller edge router is deployed, which receives the MPLS-encapsulated data from the substation.

- Substation RTUs: These are responsible for collecting and transmitting data from various sensors and devices located at the substation. Substation RTU interfaces with an IR8340 edge router to enable data transfer to the controller and these are located at the substation side. One IR8340 router is installed at the end of each substation RTU.
- SCADA controller: This is located at the controller side. One IR8340 router is installed at the end of the SCADA controller.
- MPLS virtual circuit (VC): It is created between the two edge routers using MPLS LDP signaling protocol.

For more information about MPLS forwarding and raw socket configuration, refer the configuration guidelines reference links MPLS Pseudowire Status Signaling and Raw Socket Transport.

Serial MPLS pseudowire for compliance and cost efficiency data transfer

Async serial data is exchanged between substation RTUs and a remote SCADA management station through a public IP network. However, due to compliance with regulatory requirements, you must perform a firewall inspection for TCP/IP packets at the substation RTUs before data enters the cloud. In this scenario, customers have large numbers of substation plants and few remote SCADA servers. Using existing IP-based data transfer, user must install firewalls at all substation plant locations, thereby increasing investment for data transfer.

In the above async serial data exchange scenario, you can control data transfer costs by skipping the TCP/IP headers and sending the data in the MPLS pseudowire format. This approach reduces the cost by eliminating the need for firewall inspections at the substation RTUs. After you decapsulate the data at the controller edge router, add the TCP/IP headers and the IP address of the remote SCADA server. Then, forward this data through the IP network and the firewall over an established raw socket session.

Data exchange between substation RTU and SCADA controller using MPLS pseudowire

Data is exchanged between the substation and the controller’s edge routers using a serial MPLS pseudowire.

Data transfer from substation RTU to SCADA controller

DNP3 or SCADA serial data received at the substation router is encapsulated with an MPLS label and transferred over a pseudowire towards the controller's edge router. An MPLS VC is created between the two edge routers with an RS232 async serial interface as the Attachment Circuit (AC) on the substation router and loopback as an AC on the controller router. The substation router labels each async payload with a configured VC identifier. This allows the controller-side router to map the incoming serial packet to the correct VC and forward it to the right SCADA controller IP address. After receiving the data from the substation edge router, the controller edge router performs these tasks:

- Decapsulates the MPLS-encapsulated async serial packets to retrieve the original async serial data.
- Adds TCP/IP headers for the remote SCADA server and send the data through an established raw socket session over the IP network and firewall.

Data transfer from SCADA controller to substation RTU

When data is received from a remote SCADA, the controller edge router removes TCP/IP headers and encapsulates the raw serial data into MPLS frames to send back to the substation router through an MPLS pseudowire.

Serial MPLS pseudowire deployment scenarios

Serial pseudowire feature supports following two deployment scenarios between the two IR8340 edge routers:

- Serial to serial MPLS pseudowire deployment
- Serial to raw socket (IP) MPLS pseudowire deployment

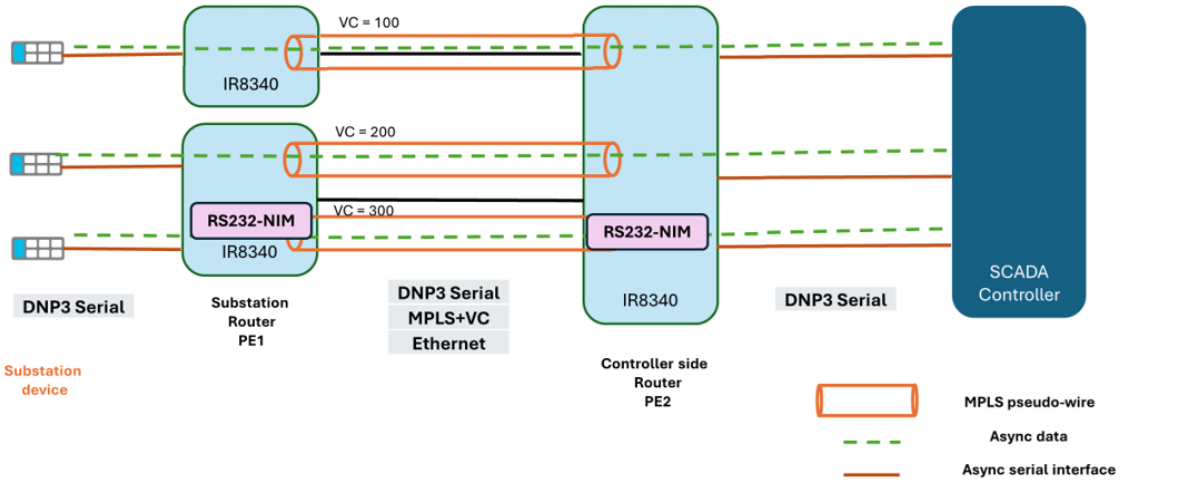
Serial to serial MPLS pseudowire deployment

A like-to-like MPLS virtual circuit is created between the two IR8340 edge routers with RS232 async serial interface as Attachment Circuit. In this scenario, both substation and controller-side edge routers have serial interface endpoints. In this deployment method, two IR8340 edge routers performs these tasks:

- Transfers data using serial interfaces between the two edge routers.
- Creates an MPLS VC between the two edge routers.
- Encapsulates the data packet from the substation RTU into an MPLS label stack and transfer it to the controller's edge router.
- Decapsulates the received data at the controller router and transfer the raw data towards the FEP controller.

Note

- The controller's edge router transfers data to the substation's edge router in the same manner.
- Both the encapsulation and decapsulation of data packet occurs at substation and controller edge routers.



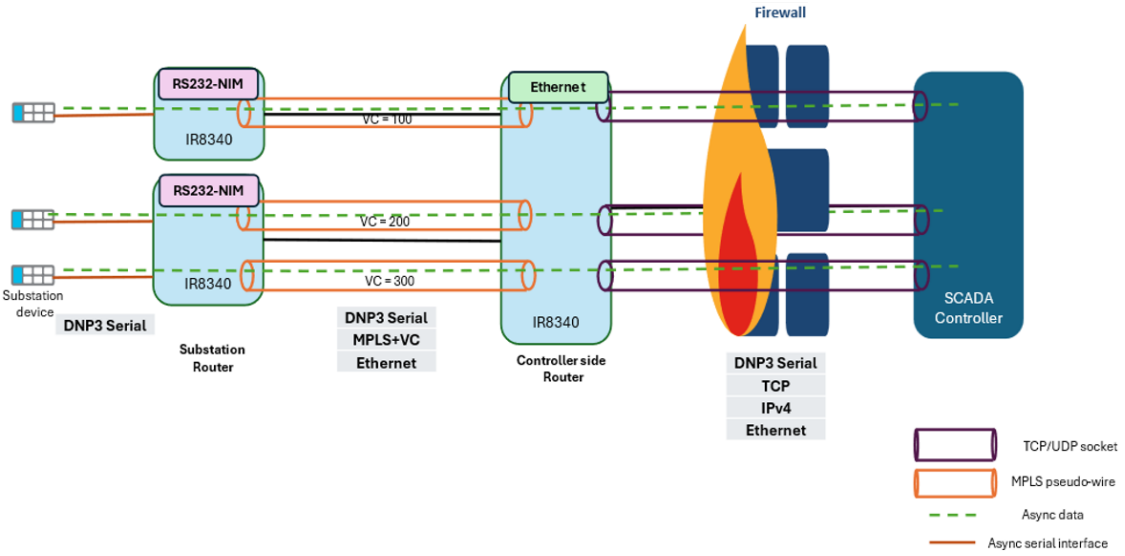
Serial to raw socket (IP) MPLS pseudowire deployment

In this deployment, to transfer the data, the customer has an RS232 serial endpoint at substation edge router and raw socket virtual interface as terminating endpoint on the controller side router for end-to-end SCADA service. This necessitates serial to raw socket interworking between the two edge routers to interconnect two ACs. The Loopback interface is configured as an AC interface on the controller router to set up the serial to raw socket virtual circuit. In this scenario, the following data exchange happens between the serial to raw socket of IR8340 edge routers:

- Transfers data using serial to raw socket between both edge routers.
- Creates an MPLS VC between the two edge routers.
- Encapsulates the data packet from the substation RTU into an MPLS label stack and transfer it to the controller's edge router.
- Decapsulates the received data at the controller router and transfer the raw data towards the FEP controller.

Note

- The controller's edge router transfers data to the substation's edge router in the same manner.
- Both the encapsulation and decapsulation of data packet occurs at substation and controller edge routers.



Serial to serial MPLS pseudowire deployment configuration using CLI

Perform these tasks for serial to serial configuration between two edge routers to transfer the data.

Procedure

Step 1 Use the **pseudowire-class** *PW-class-name* command to create a pseudowire class.

```
device#pseudowire-class SERIAL_PW
```

Step 2 Use the **encapsulation mpls** command to facilitate the encapsulation and transfer of data over an MPLS network.

```
device#encapsulation mpls
```

Step 3 Use the **status control-plane route-watch** command to monitor the status of the control plane and to track route changes on the device.

```
device#status control-plane route-watch
```

Step 4 Use the **switching tlv** command to configure the TLV settings on the device.

```
device#switching tlv
```

Step 5 Use the **interface Serial***0/2/4* command to configure the specific serial interface on the device.

```
device#interface Serial0/2/4
```

Note

- 0: refers to the chassis,
- 2: represents the slot number within the chassis, and
- 4: denotes the specific port or interface number on the module.

Step 6 Use the **physical-layer async** command to configure serial interface for async data transfer.

```
device#physical-layer async
```

Step 7 Use the **xconnect** *IP address VC ID value* **encapsulation mpls pw-class SERIAL_PW** command to configure the VC to a remote endpoint.

```
device(config-if)#xconnect 199.11.1.1 2000 e
device(config-if)#xconnect 199.11.1.1 2000 encapsulation m
device(config-if)#xconnect 199.11.1.1 2000 encapsulation mpls pw
device(config-if)#$000 encapsulation mpls pw-class SERIAL_TO_SERIAL
device(config-if-xconn)#
```

Note

- You can configure a similar serial xconnect on the serial interface of the controller-side router.
- Here xconnect IP address denotes the IP address of the remote endpoint. This command establishes a point-to-point connection over a pseudowire between the local and remote devices.

For more information about MPLS configuration guide, refer MPLS Pseudowire Status Signaling.

Serial to raw socket (IP) MPLS pseudowire deployment configuration using CLI

Perform these tasks for serial to raw socket (IP) configuration between two edge routers to transfer the data.

Before you begin

Note First you must configure the raw socket on the loop back interface, before MPLS pseudowire VC is established.

Procedure

Step 1 Use the **pseudowire-class** *PW-class-name* command to create a pseudowire class.

```
device#pseudowire-class SERIAL_PW
```

Step 2 Use the **encapsulation mpls** command to facilitate the encapsulation and transfer of data over an MPLS network.

```
device#encapsulation mpls
```

Step 3 Use the **interworking ethernet** command to enable Ethernet interworking on a network device.

```
device#interworking ethernet
```

Note
- Interworking Ethernet configuration is also required on the serial interface of the substation router.
- As we have different types of ACs at both ends, interworking ethernet is required to make the VC functional.

Step 4 Use the **status** command to verify the current state of the device.

```
device#status
```

Step 5 Use the **status control-plane route-watch** command to monitor the status of the control plane and to track route changes on the device.

```
device#status control-plane route-watch
```

Step 6 Use the **switching tlv** command to configure the TLV settings on the device.

```
device#switching tlv
```

Step 7 Use the **interface Loopback***1* command to configure the loopback interface on the device.

```
device(config)#interface Loopback1
```

Note 1: Identifier for the specific loopback interface. This numbering helps distinguish between different loopback interfaces on a network device.

Step 8 Use the **no ip address** command is used to ensure that no IP address is assigned to the interface. If you don't assign an IP address, the loopback interface is used solely as an AC for the VC in the MPLS network.

```
device#no ip address
```

Step 9 Use the **mtu** *value* command to configure the mtu value as 1514 bytes on the specified interface.

```
device (config)#
device (config)#int ser0/3/4
device (config-if)#mtu 1514
device (config-if)#
```

Note
- Configuring the MTU default value as 1514 bytes ensures that the data packets are of the correct size for your network configuration. This prevents them from breaking up and improves performance.
- MTU value should be same on both the serial interfaces of substation router and loopback interface on controller router.

Step 10 Use the **xconnect** *IP address VC ID value* **encapsulation mpls pw-class SERIAL_PW** command to configure the VC to a remote endpoint.

```
device(config-if)#xconnect 199.11.1.1 2000 e
device(config-if)#xconnect 199.11.1.1 2000 encapsulation m
device(config-if)#xconnect 199.11.1.1 2000 encapsulation mpls pw
device(config-if)#$000 encapsulation mpls pw-class SERIAL_TO_IP
device(config-if-xconn)#
```

Note Here xconnect IP address denotes the IP address of the remote endpoint. This command establishes a point-to-point connection over a pseudowire between the local and remote devices.

For more methods of serial to raw socket configurations, refer chapter Raw Socket Transport.

Raw socket configuration using CLI

Use the **Interface loopback** *value* raw-socket *tcp* client *server_ip server_port client_ip client_port* command to configure a loopback interface with a raw TCP socket setup, where the interface acts as a TCP client connecting to a specified server IP and port.

```
device(config)#
device(config)#inter
device(config)#interface Loo
device(config)#interface Loopback 10
device(config-if)#raw
```

```
device(config-if)#raw-socket tcp cl
device(config-if)#raw-socket tcp client 6.6.6.1 5000 6.6.6.2 4000
device(config-if)#
```

Verify serial to serial configurations using CLI

Use the **sh xconnect all** command to verify the serial to serial configuration on the device.

```
Device#sh xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                          S2
------+---------------------------------+--+---------------------------------+--
UP pri   ac Se0/2/4(HDLC)                UP mpls 192.168.1.102:1000            UP
UP pri   ac Se0/2/5(HDLC)                UP mpls 192.168.1.102:2000            UP
```

verify serial interface configuration using CLI

Use the **sh run int ser0/2/5** command to verify the serial interface configuration on the device.

```
Device#sh run int ser0/2/5
Building configuration...

Current configuration : 150 bytes
!
interface Serial0/2/5
 physical-layer async
 no ip address
 cdp enable
 xconnect 192.168.1.102 1000 encapsulation mpls pw-class SERIAL_PW_TEST
end
```

Verify serial to serial MPLS Layer2 transport using CLI

Use the **sh mpls l2transport vc** *id* **detail** command to verify layer2 vpn provisioned vc details on the device.

```
Device#sh mpls l2transport vc 1000 detail
Local interface: Se0/2/5 up, line protocol up, HDLC up
  Destination address: 192.168.1.102, VC ID: 1000, VC status: up
    Output interface: Gi0/0/1, imposed label stack {19 20}
    Preferred path: not configured
    Default path: active
    Next hop: 11.1.1.102
  Create time: 01:04:45, last status change time: 00:57:29
    Last label FSM state change time: 00:57:29
  Signaling protocol: LDP, peer 192.168.1.102:0 up
    Targeted Hello: 192.168.1.103(LDP Id) -> 192.168.1.102, LDP is UP
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
```

```
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 19, remote 20
    Group ID: local n/a, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 192.168.1.102/1000, local label: 19
  Dataplane:
    SSM segment/switch IDs: 8199/8195 (used), PWID: 1
  VC statistics:
    transit packet totals: receive 9, send 11
    transit byte totals:   receive 540, send 319
    transit packet drops:  receive 0, seq error 0, send 0
```

Verify the serial to serial data and operational statistics of the specified serial interface

Use the **hardware subslot 0/2 module interface ser0/2/5 statistics** command is to gather and display performance data and operational statistics for the specified serial interface on the device.

```
Device#$ hardware subslot 0/2 module interface ser0/2/5 statistics

SCC driver stats
----------------
 Rx intr: 0 Tx intr: 11
 Rx frames: 0 Tx frames: 11
 Rx bytes: 0 Tx bytes: 11
 Rx host-if down drops: 0
 Rx Resource errors: 0
 Rx Unrecoverable errors: 0
 Tx unrecoverable errors: 0
 Tx chain errors: 0
 Tx underruns: 0
 Rx ring head: 11
 HP1 Tx ring head: 20  tail: 20
 HP2 Tx ring head: 0  tail: 0
 LP Tx ring head:  0  tail: 0
 Tx ring size: 1024  Rx ring size: 128

Tx Xmit Checks
--------------
 Tx skb null: 0
 Tx num ports not initialized: 0
 Tx invalid iid: 0
 Tx port disabled: 0
 Tx line down: 0
 Tx invalid qos priority: 0
 Tx dring full: 0
 Tx buf exceeds mtu: 0


Supported HQF Queues: HP1 HP2 LP

HQF stats port 5                HP1                 HP2                  LP
-------------------------------------------------------------------------------
       Throttles                  0                   0                   0
         Enables                  0                   0                   0
Throttle refresh                  0                   0                   0
  Enable refresh                  0                   0                   0
       Throttled                  0                   0                   0
      Tx Packets                 11                   0                   0
        Tx bytes                 11                   0                   0
```

```
      Tx Drops                    0                    0                    0
  Overflow drop                   0                    0                    0
  Tx Queue size                1024                 1024                 1024
     Max Qdepth                   0                    0                    0
     Cur Qdepth                   0                    0                    0
```

Verify Serial to raw socket (IP) configurations using CLI

Use the **sh xconnect all** command to verify the serial to raw socket configuration.

```
Device#sh xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down             AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby      RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                          S2
------+-----------------------------------+--+-----------------------------------+--
UP pri   ac Lo8(ASYNC)                   UP mpls 192.168.1.103:1000            UP
```

Verify serial to raw socket MPLS Layer2 transport using CLI

Use the **sh mpls l2transport vc** *id* **detail** command to verify serial to raw socket MPLS Layer2 vpn provisioned vc details.

```
Device#sh mpls l2transport vc 1000 detail
Local interface: Lo8 up, line protocol up, ASYNC up
  Interworking type is Ethernet
  Destination address: 192.168.1.103, VC ID: 1000, VC status: up
    Output interface: Gi0/0/1, imposed label stack {16 16}
    Preferred path: not configured
    Default path: active
    Next hop: 13.1.1.102
  Create time: 00:04:07, last status change time: 00:04:07
    Last label FSM state change time: 00:04:07
  Signaling protocol: LDP, peer 192.168.1.103:0 up
    Targeted Hello: 192.168.1.102(LDP Id) -> 192.168.1.103, LDP is UP
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 21, remote 16
    Group ID: local n/a, remote 0
    MTU: local 1514, remote 1514
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 192.168.1.103/1000, local label: 21
  Dataplane:
    SSM segment/switch IDs: 4097/4096 (used), PWID: 1
  VC statistics:
```

```
    transit packet totals: receive 0, send 0
    transit byte totals:   receive 0, send 0
    transit packet drops:  receive 0, seq error 0, send 0
```

Verify serial to raw socket bindings between Layer 2 circuits and MPLS labels using AToM

Use the **sh l2vpn atom binding** command to verify bindings between Layer 2 circuits and MPLS labels within an L2VPN using AToM on the device.

```
Device#sh l2vpn atom binding
  Destination Address: 192.168.1.103,VC ID: 1000
    Local Label:  21
        Cbit: 1,    VC Type: Ethernet,    GroupID: n/a
        MTU: 1514,   Interface Desc: n/a
        VCCV: CC Type: RA [2], TTL [3]
              CV Type: LSPV [2]
    Remote Label: 16
        Cbit: 1,    VC Type: Ethernet,    GroupID: 0
        MTU: 1514,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2], TTL [3]
              CV Type: LSPV [2]
```

Verify raw-socket tcp sessions

Use the **sh raw-socket tcp sessions** command to verify established tcp session with loopback AC and remote raw socket server on the device.

```
Device#sh raw-socket tcp sessions
------------------------------------------------- TCP Sessions
-----------------------------------------------------
Interface   tty/(Idx)     vrf_name           socket   mode    local_ip_addr  local_port
 dest_ip_addr  dest_port    up_time     idle_time/timeout
  Lo8       ( 35)                                0    client   55.0.0.2         4000
    55.0.0.1       5000     00:00:29    00:00:29/300sec
```

Verify raw-socket tcp statistic

Use the **sh raw-socket tcp statistic** command to verify the raw socket transport tcp session packet statistics created between loopback based client and SCADA server over IP network on the device.

```
Device#sh raw-socket tcp statistic
------------------------------------------- TCP-Serial Statistics
-----------------------------------------
Interface  idx             vrf_name             sessions      tcp_in_bytes
tcp_out_bytes   tcp_to_tty_frames    tty_to_tcp_frames
Lo8         35                                      1             0                    0
                  0                    0
Se0/3/4     54                                      1             0                    0
                  0                    0
```

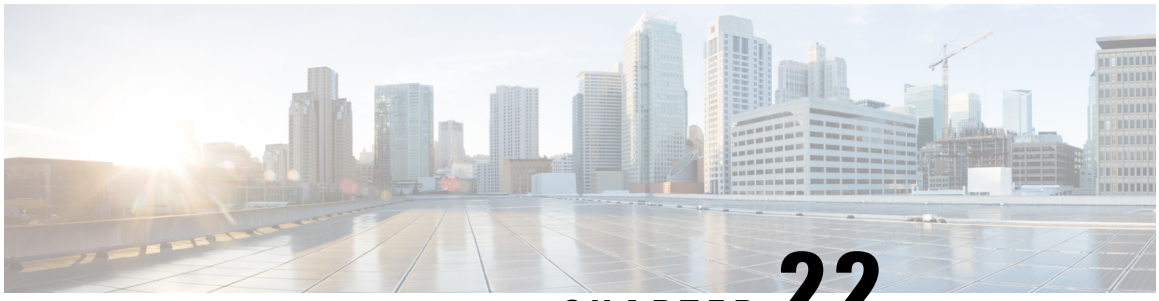
Verify raw socket configuration using CLI

Use the **sh raw-socket tcp sessions local | inc Lo10** command to verify the raw socket configuration on the device.

```
device#sh raw-socket tcp sessions local | inc Lo10
Lo10       48   6.6.6.1           5000     6.6.6.2           4000      DOWN
```

Limitations and restrictions of current serial MPLS pseudowire release

- Current IOS XE Release 17.17.1 supports only RS232-NIM based serial to serial and Serial to IP use case scenarios. It does not support T1E1-NIM based deployment cases in this release.
- This feature is supported only on the IR8340 and not on other IoT routing devices.
- Serial and loopback AC ports support interface-based xconnect configuration but do not support L2VPN xconnect configuration.
- For the serial to IP use case, only raw socket functionality is enabled for the loopback interface.
- Current IOS XE Release 17.17.1 does not support the raw socket server and other packetization criteria such as packet length, timer, and special character. Data packets are exchanged between substation and controller.
- The loopback does not support a raw socket using the UDP protocol.
- In a multi-controller environment, user can configure up to 1000 raw socket TCP clients on a loopback interface to support high availability.
- The serial PW feature on the controller router mainly supports SCADA aggregator functions for substation RTUs, using loopback as the PW endpoint and a raw socket client. Although having both serial and loopback-based raw sockets on the controller router is not typical for utility customers, this setup is supported for future needs. In this case, unique destination and source IP addresses and TCP port numbers are recommended to ensure proper functionality.
- In a serial-IP deployment, the VC statistics show Rx counters as zero because the packets are sent to the raw socket IOS application for further TCP/IP processing.
- You cannot configure the same raw socket TCP client on two loopbacks.
- Enable the raw socket before configuring xconnect on a loopback.
- You cannot configure both the VRF and xconnect on the same serial async interface.



CHAPTER 22

Configuring MODBUS TCP

This chapter provides the following sections:

- Understanding MODBUS TCP, on page 295
- Configuring the Router as the MODBUS TCP Server, on page 297
- MODBUS TCP Registers, on page 298
- NERC CIP compliance, on page 309

Understanding MODBUS TCP

Use Modicon Communication Bus (MODBUS) TCP over an Ethernet network when connecting the router to devices such as intelligent electronic devices (IEDs), distributed controllers, substation routers, Cisco IP Phones, Cisco Wireless Access Points, and other network devices such as redundant substation routeres.

MODBUS is a serial communications protocol for client-server communication between a router (server) and a device in the network running MODBUS client software (client). You can use MODBUS to connect a computer to a remote terminal unit (RTU) in supervisory control and data acquisition (SCADA) systems.

The client can be an IED or a human machine interface (HMI) application that remotely configure and manage devices running MODBUS TCP. The router functions as the server.

The router encapsulates a request or response message in a MODBUS TCP application data unit (ADU). A client sends a message to a TCP port on the router. The default port number is 502.

Note For information about the registers that a client can query on the router that functions as a MODBUS TCP server, see MODBUS TCP Registers, on page 298.

MODBUS and Security

If a firewall or other security services are enabled, the router TCP port might be blocked, and the router and the client cannot communicate.

If a firewall and other security services are disabled, a denial-of-service attack might occur on the router.

To configure quality of service (QoS) to set the rate-limit for MODBUS TCP traffic, create an access-list that only permits traffic sending to port number 502 that is reserved for MODBUS communication. Then attach

the access-list to the input class-map and attach it to the interface and set the rate limit to permit traffic via default port 502 and prioritize SCADA packets.

```
DUT-1:
!
class-map match-any Modbus-out-Traffic
 match qos-group 1
class-map match-any Modbus-In-Traffic
 match access-group 101
!
policy-map Modbus-In
 class Modbus-In-Traffic
  set qos-group 1
policy-map Modbus-Out
 class Modbus-out-Traffic
  police 10000000
  priority
!
!
interface GigabitEthernet0/1/1
 switchport mode access
 service-policy input Modbus-In
!
interface GigabitEthernet0/1/2
 switchport mode access
 service-policy output Modbus-Out
!
!
interface Vlan1
 no ip address
 ip access-group 101 in
 rate-limit input access-group 101 8000 8000 8000 conform-action transmit exceed-action
drop
!
!
!
access-list 101 permit tcp 10.10.10.0 0.0.0.255 any eq 502

DUT-2:

interface Vlan1
 ip address 192.168.1.2 255.255.255.0
```

This example shows that 133 SCADA packets were classified.

```
DUT-1#show policy-map interface GigabitEthernet0/1/2
 GigabitEthernet0/1/2
  Service-policy output: Modbus-Out
    Class-map: Modbus-out-Traffic (match-any)
      133 packets
      Match: qos-group 1
      police cir 10000000 bc 312500
         conform-action transmit
         exceed-action drop
      conform: 133 (packets) exceed: 0 (packets)
        Priority
      Output Queue:
        Max queue-limit default threshold: 272
        Tail Packets Drop: 0
```

Multiple Request Messages

The router can receive multiple request messages from clients and respond to them simultaneously.

You can set the number of client connections from 1 to 5. The default is 1.

Configuring the Router as the MODBUS TCP Server

Defaults

The router is not configured as a MODBUS TCP server.

The TCP port number is 502.

The number of simultaneous connection requests is 1.

Enabling MODBUS TCP on the Switch

Beginning in privileged EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 2	**scada modbus tcp server**	Enables MODBUS TCP on the router.
Step 3	**scada modbus tcp server** [**port** *tcp-port-number*]	(Optional) Sets the TCP port to which clients send messages. The range for *tcp-port-number* is 1 to 65535. The default is 502.
Step 4	**scada modbus tcp server** [**connection** *connection-requests*]	(Optional) Sets the number of simultaneous connection requests sent to the router. The range for *connection-requests* is 1 to 5. The default is 1.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

To disable MODBUS on the router and return to the default settings, enter the **no scada modbus tcp server** global configuration command.

To add security when using MODBUS TCP, configure an ACL to permit traffic from specific clients or configure QoS to rate-limit traffic.

MODBUS TCP Registers

This section lists the read-only MODBUS registers. MODBUS clients use them to communicate with a MODBUS server (i.e., the IR8340 router). There are no writable registers.

System Information Registers

Memory address spaces 0x0800 through 0x0FFF are system information registers. Clients use the 0x03 Read Multiple Registers MODBUS function code. The system-information register mapping is as follows:

Table 19: System Information Registers

Address	# of Registers	Description	R/W	Format
0x0800	64	Product ID	R	Text
0x0840	64	Software image name	R	Text
0x0880	64	Software image version	R	Text
0x08C0	64	Host name	R	Text
0x0900	1	Number of Gigabit Ethernet ports	R	Uint16
CPU Core Temperature Related Registers				
0x0901	1	CPU core-0 temperature (in Celsius)	R	Uint16
0x0902	1	CPU core-1 temperature (in Celsius)	R	Uint16
0x0903	1	CPU core-2 temperature (in Celsius)	R	Uint16
0x0904	1	CPU core-3 temperature (in Celsius)	R	Uint16
0x0905	1	CPU core-4 temperature (in Celsius)	R	Uint16
0x0906	1	CPU core-5 temperature (in Celsius)	R	Uint16

Address	# of Registers	Description	R/W	Format
0x0907	1	CPU core-6 temperature (in Celsius)	R	Uint16
0x0908	1	CPU core-7 temperature (in Celsius)	R	Uint16

Port Information Registers

Memory address spaces 0x1000 through 0x3FFF are read-only interface registers. Clients use the 0x03 Read Multiple Registers MODBUS function code to access the registers.

The following table shows the memory map for all interface registers, with 64-bit counters (address space 0x1000 – 0x2FFF, 8K registers):

Table 20: System Information Registers

Address	# of Registers	Description	R/W	Format
0x1000	64	WAN Port 1 name	R	Text
0x1040	64	WAN Port 2 name	R	Text
0x1080	64	LAN Port 1 name	R	Text
0x10C0	64	LAN Port 2 name	R	Text
0x1100	64	LAN Port 3 name	R	Text
0x1140	64	LAN Port 4 name	R	Text
0x1180	64	LAN Port 5 name	R	Text
0x11C0	64	LAN Port 6 name	R	Text
0x1200	64	LAN Port 7 name	R	Text
0x1240	64	LAN Port 8 name	R	Text
0x1280	64	LAN Port 9 name	R	Text
0x12C0	64	LAN Port 10 name	R	Text
0x1300	64	LAN Port 11 name	R	Text
0x1340	64	LAN Port 12 name	R	Text
0x1380	1	WAN Port 1 state	R	Uint16
0x1381	1	WAN Port 2 state	R	Uint16
0x1382	1	LAN Port 1 state	R	Uint16
0x1383	1	LAN Port 2 state	R	Uint16
0x1384	1	LAN Port 3 state	R	Uint16

Address	# of Registers	Description	R/W	Format
0x1385	1	LAN Port 4 state	R	Uint16
0x1386	1	LAN Port 5 state	R	Uint16
0x1387	1	LAN Port 6 state	R	Uint16
0x1388	1	LAN Port 7 state	R	Uint16
0x1389	1	LAN Port 8 state	R	Uint16
0x138A	1	LAN Port 9 state	R	Uint16
0x138B	1	LAN Port 10 state	R	Uint16
0x138C	1	LAN Port 11 state	R	Uint16
0x138D	1	LAN Port 12 state	R	Uint16
Values for 64-Bit Counters				
0x138E	4	WAN Port 1 Statistics – Number of packets received	R	Uint64
0x1392	4	WAN Port 2 Statistics – Number of packets received	R	Uint64
0x1396	4	LAN Port 1 Statistics – Number of packets received	R	Uint64
0x139A	4	LAN Port 2 Statistics – Number of packets received	R	Uint64
0x139E	4	LAN Port 3 Statistics – Number of packets received	R	Uint64
0x13A2	4	LAN Port 4 Statistics – Number of packets received	R	Uint64
0x13A6	4	LAN Port 5 Statistics – Number of packets received	R	Uint64
0x13AA	4	LAN Port 6 Statistics – Number of packets received	R	Uint64
0x13AE	4	LAN Port 7 Statistics – Number of packets received	R	Uint64

Address	# of Registers	Description	R/W	Format
0x13B2	4	LAN Port 8 Statistics – Number of packets received	R	Uint64
0x13B6	4	LAN Port 9 Statistics – Number of packets received	R	Uint64
0x13BA	4	LAN Port 10 Statistics – Number of packets received	R	Uint64
0x13BE	4	LAN Port 11 Statistics – Number of packets received	R	Uint64
0x13C2	4	LAN Port 12 Statistics – Number of packets received	R	Uint64
0x13C6	4	WAN Port 1 Statistics – Number of packets sent	R	Uint64
0x13CA	4	WAN Port 2 Statistics – Number of packets sent	R	Uint64
0x13CE	4	LAN Port 1 Statistics – Number of packets sent	R	Uint64
0x13D2	4	LAN Port 2 Statistics – Number of packets sent	R	Uint64
0x13D6	4	LAN Port 3 Statistics – Number of packets sent	R	Uint64
0x13DA	4	LAN Port 4 Statistics – Number of packets sent	R	Uint64
0x13DE	4	LAN Port 5 Statistics – Number of packets sent	R	Uint64
0x13E2	4	LAN Port 6 Statistics – Number of packets sent	R	Uint64
0x13E6	4	LAN Port 7 Statistics – Number of packets sent	R	Uint64

Address	# of Registers	Description	R/W	Format
0x13EA	4	LAN Port 8 Statistics – Number of packets sent	R	Uint64
0x13EE	4	LAN Port 9 Statistics – Number of packets sent	R	Uint64
0x13F2	4	LAN Port 10 Statistics – Number of packets sent	R	Uint64
0x13F6	4	LAN Port 11 Statistics – Number of packets sent	R	Uint64
0x13FA	4	LAN Port 12 Statistics – Number of packets sent	R	Uint64
0x13FE	4	WAN Port 1 Statistics – Number of bytes received	R	Uint64
0x1402	4	WAN Port 2 Statistics – Number of bytes received	R	Uint64
0x1406	4	LAN Port 1 Statistics – Number of bytes received	R	Uint64
0x140A	4	LAN Port 2 Statistics – Number of bytes received	R	Uint64
0x140E	4	LAN Port 3 Statistics – Number of bytes received	R	Uint64
0x1412	4	LAN Port 4 Statistics – Number of bytes received	R	Uint64
0x1416	4	LAN Port 5 Statistics – Number of bytes received	R	Uint64
0x141A	4	LAN Port 6 Statistics – Number of bytes received	R	Uint64
0x141E	4	LAN Port 7 Statistics – Number of bytes received	R	Uint64

Address	# of Registers	Description	R/W	Format
0x1422	4	LAN Port 8 Statistics – Number of bytes received	R	Uint64
0x1426	4	LAN Port 9 Statistics – Number of bytes received	R	Uint64
0x142A	4	LAN Port 10 Statistics – Number of bytes received	R	Uint64
0x142E	4	LAN Port 11 Statistics – Number of bytes received	R	Uint64
0x1432	4	LAN Port 12 Statistics – Number of bytes received	R	Uint64
0x1436	4	WAN Port 1 Statistics – Number of bytes sent	R	Uint64
0x143A	4	WAN Port 2 Statistics – Number of bytes sent	R	Uint64
0x143E	4	LAN Port 1 Statistics – Number of bytes sent	R	Uint64
0x1442	4	LAN Port 2 Statistics – Number of bytes sent	R	Uint64
0x1446	4	LAN Port 3 Statistics – Number of bytes sent	R	Uint64
0x144A	4	LAN Port 4 Statistics – Number of bytes sent	R	Uint64
0x144E	4	LAN Port 5 Statistics – Number of bytes sent	R	Uint64
0x1452	4	LAN Port 6 Statistics – Number of bytes sent	R	Uint64
0x1456	4	LAN Port 7 Statistics – Number of bytes sent	R	Uint64

Address	# of Registers	Description	R/W	Format
0x145A	4	LAN Port 8 Statistics – Number of bytes sent	R	Uint64
0x145E	4	LAN Port 9 Statistics – Number of bytes sent	R	Uint64
0x1462	4	LAN Port 10 Statistics – Number of bytes sent	R	Uint64
0x1466	4	LAN Port 11 Statistics – Number of bytes sent	R	Uint64
0x146A	4	LAN Port 12 Statistics – Number of bytes sent	R	Uint64
Values for 32-Bit Counters				
0x146E	2	WAN Port 1 Statistics – Number of packets received	R	Uint32
0x1470	2	WAN Port 1 Statistics – Number of packets received	R	Uint32
0x1472	2	LAN Port 1 Statistics – Number of packets received	R	Uint32
0x1474	2	LAN Port 2 Statistics – Number of packets received	R	Uint32
0x1476	2	LAN Port 3 Statistics – Number of packets received	R	Uint32
0x1478	2	LAN Port 4 Statistics – Number of packets received	R	Uint32
0x147A	2	LAN Port 5 Statistics – Number of packets received	R	Uint32
0x147C	2	LAN Port 6 Statistics – Number of packets received	R	Uint32

Address	# of Registers	Description	R/W	Format
0x147E	2	LAN Port 7 Statistics – Number of packets received	R	Uint32
0x1480	2	LAN Port 8 Statistics – Number of packets received	R	Uint32
0x1482	2	LAN Port 9 Statistics – Number of packets received	R	Uint32
0x1484	2	LAN Port 10 Statistics – Number of packets received	R	Uint32
0x1486	2	LAN Port 11 Statistics – Number of packets received	R	Uint32
0x1488	2	LAN Port 12 Statistics – Number of packets received	R	Uint32
0x148A	2	WAN Port 1 Statistics – Number of packets sent	R	Uint32
0x148C	2	WAN Port 2 Statistics – Number of packets sent	R	Uint32
0x148E	2	LAN Port 1 Statistics – Number of packets sent	R	Uint32
0x1490	2	LAN Port 2 Statistics – Number of packets sent	R	Uint32
0x1492	2	LAN Port 3 Statistics – Number of packets sent	R	Uint32
0x1494	2	LAN Port 4 Statistics – Number of packets sent	R	Uint32
0x1496	2	LAN Port 5 Statistics – Number of packets sent	R	Uint32
0x1498	2	LAN Port 6 Statistics – Number of packets sent	R	Uint32

Address	# of Registers	Description	R/W	Format
0x149A	2	LAN Port 7 Statistics – Number of packets sent	R	Uint32
0x149C	2	LAN Port 8 Statistics – Number of packets sent	R	Uint32
0x149E	2	LAN Port 9 Statistics – Number of packets sent	R	Uint32
0x14A0	2	LAN Port 10 Statistics – Number of packets sent	R	Uint32
0x14A2	2	LAN Port 11 Statistics – Number of packets sent	R	Uint32
0x14A4	2	LAN Port 12 Statistics – Number of packets sent	R	Uint32
0x14A6	2	WAN Port 1 Statistics – Number of bytes received	R	Uint32
0x14A8	2	WAN Port 2 Statistics – Number of bytes received	R	Uint32
0x14AA	2	LAN Port 1 Statistics – Number of bytes received	R	Uint32
0x14AC	2	LAN Port 2 Statistics – Number of bytes received	R	Uint32
0x14AE	2	LAN Port 3 Statistics – Number of bytes received	R	Uint32
0x14B0	2	LAN Port 4 Statistics – Number of bytes received	R	Uint32
0x14B2	2	LAN Port 5 Statistics – Number of bytes received	R	Uint32
0x14B4	2	LAN Port 6 Statistics – Number of bytes received	R	Uint32

Address	# of Registers	Description	R/W	Format
0x14B6	2	LAN Port 7 Statistics – Number of bytes received	R	Uint32
0x14B8	2	LAN Port 8 Statistics – Number of bytes received	R	Uint32
0x14BA	2	LAN Port 9 Statistics – Number of bytes received	R	Uint32
0x14BC	2	LAN Port 10 Statistics – Number of bytes received	R	Uint32
0x14BE	2	LAN Port 11 Statistics – Number of bytes received	R	Uint32
0x14C0	2	LAN Port 12 Statistics – Number of bytes received	R	Uint32
0x14C2	2	WAN Port 1 Statistics – Number of bytes sent	R	Uint32
0x14C4	2	WAN Port 2 Statistics – Number of bytes sent	R	Uint32
0x14C6	2	LAN Port 1 Statistics – Number of bytes sent	R	Uint32
0x14C8	2	LAN Port 2 Statistics – Number of bytes sent	R	Uint32
0x14CA	2	LAN Port 3 Statistics – Number of bytes sent	R	Uint32
0x14CC	2	LAN Port 4 Statistics – Number of bytes sent	R	Uint32
0x14CE	2	LAN Port 5 Statistics – Number of bytes sent	R	Uint32
0x14D0	2	LAN Port 6 Statistics – Number of bytes sent	R	Uint32

Address	# of Registers	Description	R/W	Format
0x14D2	2	LAN Port 7 Statistics – Number of bytes sent	R	Uint32
0x14D4	2	LAN Port 8 Statistics – Number of bytes sent	R	Uint32
0x14D6	2	LAN Port 9 Statistics – Number of bytes sent	R	Uint32
0x14D8	2	LAN Port 10 Statistics – Number of bytes sent	R	Uint32
0x14DA	2	LAN Port 11 Statistics – Number of bytes sent	R	Uint32
0x14DC	2	LAN Port 12 Statistics – Number of bytes sent	R	Uint32

Interpreting the Port State

Table 21: Interpreting the Port State

Address	Description	Value
0x1380 to 0x138D	Port state information	The upper byte represents the interface state: • 0x0: Interface is down • 0x1: Interface is going down • 0x2: Interface is in the initializing state • 0x3: Interface is coming up • 0x4: Interface is up and running • 0x5: Interface is reset by the user • 0x6: Interface is shut down by the user • 0x7: Interface is being deleted The lower byte represents the line protocol state: • 0x0: Line protocol state is down • 0x1: Line protocol state is up

NERC CIP compliance

This topic lists the common configurable listening ports available on the IR8340 router. A port appears in the runtime state only when the associated service is enabled and its configuration is complete.

The Critical Infrastructure Protection (CIP) standards developed by the North American Electric Reliability Corporation (NERC) include requirements for managing logical network-accessible ports on applicable systems. Where technically feasible, the Responsible Entity must enable only the ports, port ranges, and services that it has determined are needed. This information supports an organization's compliance process but does not establish compliance.

Listening ports configuration reference

The tables list the TCP and UDP ports associated with customer-configurable services on the IR8340 router.

Note Verify the runtime state because a port listens only when the associated service is enabled and its configuration is complete.

Table 22: TCP Port Services

Port	Service	Configuration command	Description
22	SSH and SCP	`ip ssh version 2` and `transport input ssh`; for SCP, `ip scp server enable`	Provides encrypted remote CLI access and, when configured, secure file transfer by SCP.
23	Telnet	`transport input telnet` under the VTY lines	Provides unencrypted remote CLI access.
49	TACACS+ client (outbound)	`tacacs server` *`server-name`* with a configured server address	Remote destination port used by the router to reach a TACACS+ server. It is not normally a local listening port.
53	DNS server	`ip dns server`	Enables the router to answer DNS queries from clients.
80	HTTP server	`ip http server`	Provides unencrypted HTTP management access.
179	BGP	`router bgp` *`autonomous-system-number`* with an active neighbor configuration	Supports BGP peering sessions.
443	HTTPS server	`ip http secure-server`	Provides encrypted HTTPS access used by configured web-based services.
502	MODBUS TCP server	`scada modbus tcp server`	Enables the Modbus TCP server. The default port is 502; a different port in the range 1 through 65535 can be configured.
646	MPLS Label Distribution Protocol	MPLS LDP configuration, including `mpls ip` on an interface	Establishes MPLS LDP sessions with peer routers. LDP also uses UDP port 646 for discovery.
830	NETCONF over SSH	`netconf-yang`	Enables NETCONF management over SSH.

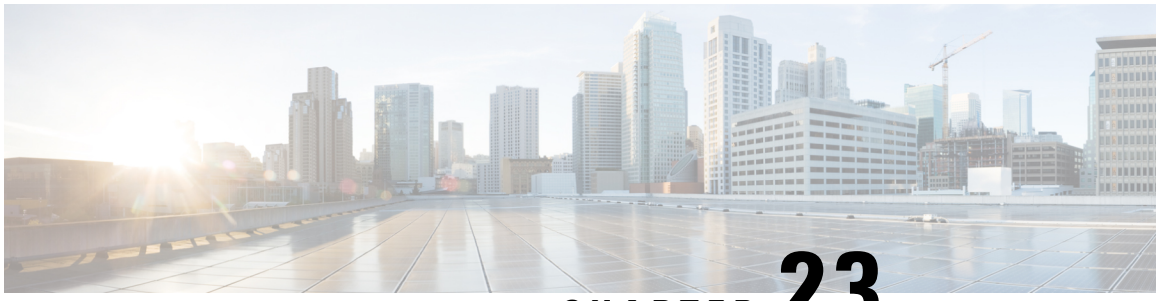
Port	Service	Configuration command	Description
2049	Unverified observed TCP endpoint	No customer configuration identified	IANA registers TCP port 2049 for NFS, but no customer-configurable IR8340 NFS listener was validated. Confirm process ownership on the target image before publication.
2404	IEC 60870-5-104 gateway	Configure an IEC 60870-5-104 channel with `local-port default`, and then enter `scada-gw enable`.	Enables the IEC 60870-5-104 SCADA gateway service. A port in the range 2000 through 65535 can be configured.
20000	DNP3-IP gateway	Configure a DNP3-IP channel with `local-port default`, and then enter `scada-gw enable`.	Enables the DNP3-IP SCADA gateway service. A port in the range 2000 through 65535 can be configured.
6666	Unverified observed TCP endpoint	No customer configuration identified	No IR8340 service association was validated. Confirm process ownership on the target image before publication; do not infer the service from the generic IANA assignment.
Configurable port	Raw Socket TCP server	`raw-socket tcp server` *`local-port`* [*`local-ip-address`*]	Enables the Raw Socket TCP server on the configured local port.

A UDP endpoint appears when the associated service is enabled. Verify the configured and operational state on the router.

Table 23: UDP Port Services

Port	Service	Configuration command	Description
53	DNS server	`ip dns server`	Enables the router to answer DNS queries from clients.
67	DHCP server	`ip dhcp pool` *`pool-name`*	Enables DHCP server operation for configured address pools.

Port	Service	Configuration command	Description
69	TFTP server	`tftp-server` *`file-system`*`:`*`file-name`*	Makes the configured file available through the TFTP server.
123	NTP	NTP server-mode configuration, such as `ntp master`	Provides network time synchronization service when the router operates as an NTP server.
161	SNMP agent	SNMP agent configuration, such as `snmp-server community` *`community-string`*	Accepts SNMP polling and management requests.
985	Unverified observed UDP endpoint	No customer configuration identified	UDP port 985 is unassigned by IANA, and no IR8340 service association was validated.
1985	HSRP	HSRP configuration, such as `standby` *`group-number`* `ip` *`virtual-ip-address`* on an interface	Exchanges Hot Standby Router Protocol hello messages with peer routers.
3503	MPLS LSP ping	MPLS forwarding and LSP ping support	Receives MPLS Label Switched Path echo requests used by MPLS ping and traceroute operations.
4500	IPsec NAT Traversal	IPsec and Internet Key Exchange configuration when NAT Traversal is used	Carries IKE and IPsec traffic through Network Address Translation when NAT Traversal is active.
49152–65535	Dynamic or private UDP ports	Allocated automatically by the operating system and enabled features	Provides temporary source or internal endpoints. Observed SNMP-related ports can include 50754, 54499, 59090, and 60221, but the selected values can change after a reload or an SNMP process restart.



CHAPTER 23

vCPU and RAM Distribution

- Introduction, on page 313
- Distribution of vCPU and RAM Resources for Cisco IOx Applications, on page 313
- Higher CPU and RAM Allocation for IOx Applications , on page 314
- Configure Data Plane Heavy Template, on page 314
- Verify the Active vCPU and RAM Distribution, on page 314
- Configure Service Plane Heavy Template, on page 315
- Verify Service Plane Heavy, on page 316

Introduction

This chapter provides information on how to distribute Virtual Central Processing Unit (vCPU) cores and RAM resources for Cisco IOx applications on Cisco Catalyst IR8340 router.

Note vCPU is also known as physical processor.

Distribution of vCPU and RAM Resources for Cisco IOx Applications

Distributing the available resources efficiently allows you to run multiple IOx applications simultaneously.

Use these templates to distribute the vCPU and RAM resources:

- **Data Plane Heavy—**Refers to a router configuration where majority of system resources are dedicated to the data plane, which is responsible for processing and forwarding network packets.

  Data Plane Heavy template maximizes throughput and ensures high-speed packet transfer, which is essential for network traffic demands. This ensures more processing power and memory to handle the increased load on the data plane, enhancing router's ability to move large volumes of data efficiently.

- **Service Plane Heavy—**Refers to a router configuration where majority of system resources are allocated to the service plane, which is responsible for providing network services such as Quality of Service (QoS), security functions, and load balancing.

Service Plane Heavy template allocates additional vCPU and RAM to IOx applications. However, it reduces data throughput (bandwidth).

Note The Service Plane Heavy is the default template for vCPU and RAM distribution in the IR8340 routers.

Routers with 2 GB RAM and a single core vCPU (IOx resources) cannot run multiple IOx applications such as Unified Threat Defense and Cisco Cyber Vision.

Higher CPU and RAM Allocation for IOx Applications

From Cisco IOS XE Release 17.15.1, the IR8340 router with 8 GB RAM supports Data Plane Heavy and Service Plane Heavy distribution templates. You can allocate 3 GB RAM and three vCPU cores. We recommend the Service Plane Heavy template to allocate resources for hosting IOx applications.

Note IR8340 also supports Control Plane Heavy distribution template.

Configure Data Plane Heavy Template

Procedure

Step 1 Enter the configuration command to enable the data plane heavy template:

```
Router(config)#platform resource data-plane-heavy
```

Step 2 Enter the reload command to reboot the router and activate the data plane heavy template:

```
Router#reload
```

What to do next

Verify the active vCPU and RAM distribution.

Verify the Active vCPU and RAM Distribution

Use the **show** command to verify the vCPU cores allocation for IOx applications.

```
Router#show platform software cpu allocation
CPU alloc information:
Control plane cpu alloc: 0-1
```

```
Data plane cpu alloc: 2-7
Service plane cpu alloc: 0-1
Slow control plane cpu alloc: 0-1
Template used: CLI-data_plane_heavy
```

Use the **show** command to verify the RAM allocation for IOx applications.

```
Router#show app-host resource
Resource Allocation:
  CPU Quota: 22%
  Available: 22
  VCPU:
  Count: 4
  Memory:
  Quota: 2048MB
  Available: 2048(MB)
Storage device: bootflash
  Quota: 1500(MB)
  Available: 1500(MB)
Storage device: IOx persist-disk
  Quota: 7281(MB)
  Available: 3468(MB)
```

Use the **show** command to verify the CPU units resource allocation for IOx applications.

```
Router#show app-host infra
IOX version: 2.11.0.3
App signature verification: disabled
CAF Health: Stable
Internal working directory: /bootflash/iox
Application Interface Mapping
AppGigabitEthernet Port #  Interface Name                Port Type            Bandwidth
          1                AppGigabitEthernet0/1/1       KR Port - Internal   10G
CPU:
  Quota: 22%
  Available: 22%
  Quota: 770(Units)
  Available: 770(Units)
```

Configure Service Plane Heavy Template

Procedure

Step 1 Enter the configuration command to enable the service plane heavy template:

```
Router(config)#platform resource service-plane-heavy
```

Step 2 Enter the reload command to reboot the router and activate the service plane heavy template:

```
Router#reload
```

What to do next

Verify the active vCPU and RAM distribution.

Verify Service Plane Heavy

Use the **show** command to verify the vCPU cores allocation.

```
Router#show platform software cpu allocation
CPU Allocation Information:
  Control plane cpu alloc: 0
  Data plane cpu alloc: 4-7
  Service plane cpu alloc: 1-3
  Template used: default-service_plane_heavy
```

Use the **show** command to verify the RAM allocation.

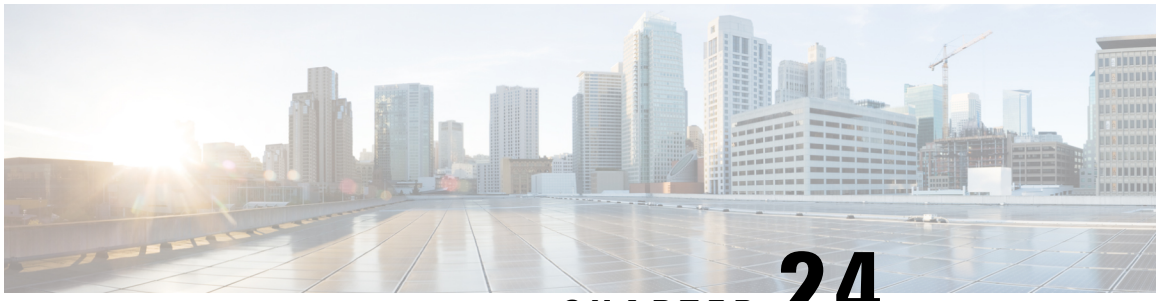
```
Router#show app-host resource
Resource Allocation:
  CPU Quota: 60%
  Available: 60%
  VCPU:
  Count: 4
Memory:
  Quota: 3072(MB)
  Available: 3072(MB)
Storage device: bootflash
  Quota: 1500(MB)
  Available: 1500(MB)
Storage device: IOx persist-disk
  Quota: 61460(MB)
  Available: 55358(MB)
```

Use the **show** command to verify the CPU units resource allocation for IOx applications.

```
Router#show app-host infra
IOX version: 2.11.0.1
App signature verification: disabled
CAF Health: Stable
Internal working directory: /vol/harddisk/iox

Application Interface Mapping
AppGigabitEthernet Port #  Interface Name    Port Type   Bandwidth
AppGigabitEthernet0/1/1    KR Port - Internal   10G

CPU:
  Quota: 60%
  Available: 60%
  Quota: 2100(Units)
  Available: 2100(Units)
```



CHAPTER 24

VLAN Access Control Lists

This chapter provides information about VLAN Access Control Lists (ACLs) and how to configure them.

- Information About VLAN Access Control Lists, on page 317
- Configuring VACLs, on page 317

Information About VLAN Access Control Lists

VLAN access control lists (VACLs) or VLAN maps are used to control network traffic within a VLAN. VACLs are configured globally, and the rules are applied on VLANs. VACLs are supported in both ingress and egress directions. In ingress direction VACLs are applied after Port ACL and before Routed ACL. In egress direction VACLs are applied after Routed ACL and before Port ACL. VLAN map is applied to both routed and switched traffic. VLAN map can contain both IP and MAC ACLs to be applied to IP and non-IP traffic respectively.

VLAN Maps

VLAN ACLs or VLAN maps are used to control network traffic within a VLAN. You can apply VLAN maps to all packets that are bridged within a VLAN in the switch or switch stack. VACLs are strictly for security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

All non-IP protocols are access-controlled through MAC addresses and Ethertype using MAC VLAN maps. (IP traffic is not access controlled by MAC VLAN maps.) You can enforce VLAN maps only on packets going through the switch; you cannot enforce VLAN maps on traffic between hosts on a hub or on another switch connected to this switch.

With VLAN maps, forwarding of packets is permitted or denied, based on the action specified in the map.

Configuring VACLs

VACL allows you to define VLAN maps and attach them to VLANs. This allows the ability to apply a single access policy across the VLAN and have different policies across different VLANs.

- VLAN map is attached using **vlan filter** <*word*> **vlan-list** <*vlanid*> command
- VLAN map is defined using **vlan access-map** <*word*> command

- In this mode, a match for IP Access-lists is specified using **match ip address** command. This will filter the traffic based on L3/L4 fields specified in the IP ACL. Match for IPv6 access lists can be configured using **match ipv6 address** command. This will filter traffic based on L3/L4 fields specified in IPv6 ACL
- A match for MAC Access-lists is specified using **match mac address** command. This will filter the traffic based on L2 fields specified in the MAC ACL.
- Each VLAN map sequence has an action forward or drop specified which specifies what should happen when the traffic matches a specified match criteria in the VLAN map.

Defining a VLAN Access Map

To define a VLAN access map, perform this task:

Command	Purpose
Router(config)# **vlan access-map** *map_name* [**0-65535**]	Defines the VLAN access map. Optionally, you can specify the VLAN access map sequence number.
Router(config)# **no vlan access-map** *map_name* **0-65535**	Deletes a map sequence from the VLAN access map.
Router(config)# **no vlan access-map** *map_name*	Deletes the VLAN access map.

When defining a VLAN access map, note the following information:

- To insert or modify an entry, specify the map sequence number.
- If you do not specify the map sequence number, a number is automatically assigned.
- You can specify only one match clause and one action clause per map sequence.
- Use the **no** keyword with a sequence number to remove a map sequence.
- Use the **no** keyword without a sequence number to remove the map.

Configuring a Match Clause in a VLAN Access Map Sequence

To configure a match clause in a VLAN access map sequence, perform this task:

Command	Purpose
Router (config-access-map)# **match** {**ip address** {**1-199** | **1300-2699** | *acl_name*}| {**mac address** *acl_name*}}	Configures the match clause in a VLAN access map sequence.
Router (config-access-map)# **no match** {**ip address** {**1-199** | **1300-2699** | *acl_name*}| {**mac address** *acl_name*}}	Deletes the match clause in a VLAN access map sequence.

When configuring a match clause in a VLAN access map sequence, note the following information:

- You can select one or more ACLs.

- Use the **no** keyword to remove a match clause or specified ACLs in the clause.

Configuring an Action Clause in a VLAN Access Map Sequence

To configure an action clause in a VLAN access map sequence, perform this task:

Command	Purpose
Router (config-access-map)# **action** {**drop** [**log**]} | {**forward** [**capture** | **vlan** *vlan_ID*]} | {**redirect** {{**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*} | {*port-channel channel_id*}}	Configures the action clause in a VLAN access map sequence.
Router (config-access-map)# **no action** {**drop** [**log**]} | {**forward** [**capture** | **vlan** *vlan_ID*]} | {**redirect** {{**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*} | {*port-channel channel_id*}}	Deletes the action clause in from the VLAN access map sequence.

Applying a VLAN Access Map

To apply a VLAN access map, perform this task:

Command	Purpose
Router(config)# **vlan filter** *map_name* {**vlan-list** *vlan_list* | interface *type number*}	Applies the VLAN access map to the specified VLANs or WAN interfaces.

Verifying VLAN Access Map Configuration

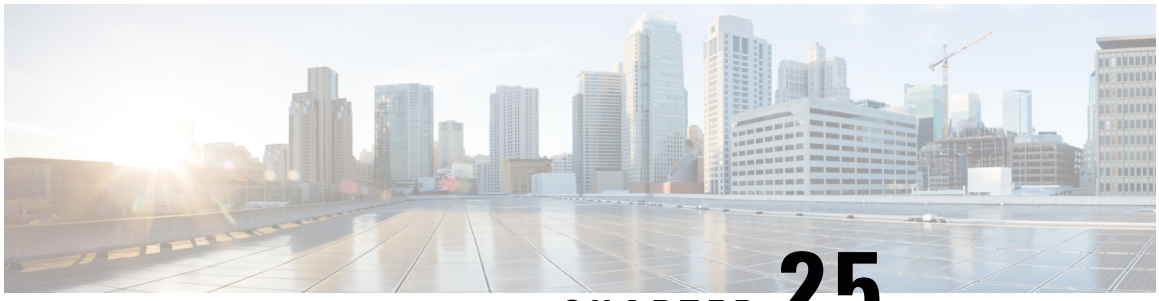
To verify VLAN access map configuration, perform this task:

Command	Purpose
Router# **show vlan access-map** [*map_name*]	Verifies VLAN access map configuration by displaying the content of a VLAN access map.
Router# **show vlan filter** [**access-map** *map_name* | **vlan** *vlan_id* | **interface** *type number*]	Verifies VLAN access map configuration by displaying the mappings between VACLs and VLANs.
Router# **show platform software fed 0/1 acl info**	Verifies that VACL is installed correctly.
Router# **show platform software fed 0/1 acl cam**	Verifies that VACL is installed correctly on TCAM.

Debugging VACLs

To enable debugs for ACl/VACL, perform this task:

Command	Purpose
Router# **set platform software trace fed 0/1 acl** <*level*>	Enable debug for FED ACL.
Router# **set platform software trace forwarding-manager RP/FP active vlan-acl** <*level*>	Enable debug for Forward-mgr VACL.



CHAPTER 25

Configuring MACsec

This section describes how to configure MACsec on Cisco IR8340 Routers.

- MACsec Encryption Overview, on page 321
- Limitations and Restrictions, on page 321
- Media Access Control Security and MACsec Key Agreement, on page 322
- Configuring MACsec Encryption, on page 323

MACsec Encryption Overview

MACsec is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. Cisco IR8340 Router supports 802.1AE encryption with MACsec Key Agreement (MKA) on switch-to-host links for encryption between the switch and host device. IR8340 also supports MACsec encryption for switch-to-switch (inter-network device) security using MKA-based key exchange protocol.

Note HSEC license is required to configure MACsec encryption.

Table 24: MACsec Support on Switch Ports

Connections	MACsec support
Switch-to-host	MACsec MKA encryption
Switch-to-switch	MACsec MKA encryption

MKA is supported on switch-to-host facing links. Host-facing links typically use flexible authentication ordering for handling heterogeneous devices with or without IEEE 802.1x, and can optionally use MKA-based MACsec encryption.

Limitations and Restrictions

- MACsec configuration is not supported on EtherChannel ports. Instead, MACsec configuration can be applied/removed on the individual member ports of an EtherChannel. To add/remove MACsec configuration, you must first unbundle the member ports from the EtherChannel.

- Configure either MACsec or PRP/HSR on the ports.
- Until PCH is available, MACsec and PTP are mutually exclusive.
- Packet number based rekey is not supported.
- Certificate based MKA (switch to switch) is not supported.
- VLAN Tag-in-clear is not supported.
- Legacy Cisco SAP (switch to switch) is not supported.
- Extended Packet Numbering (XPN) is not supported.
- Should-secure is not supported on LAN MACsec for this router.

Media Access Control Security and MACsec Key Agreement

MACsec, defined in 802.1AE, provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys. MKA and MACsec are implemented after successful authentication using certificate-based MACsec or Pre Shared Key (PSK) framework.

A device using MACsec accepts either MACsec or non-MACsec frames, depending on the policy associated with the MKA peer. MACsec frames are encrypted and protected with an integrity check value (ICV). When the device receives frames from the MKA peer, it decrypts them and calculates the correct ICV by using session keys provided by MKA. The device compares that ICV to the ICV within the frame. If they are not identical, the frame is dropped. The device also encrypts and adds an ICV to any frames sent over the secured port (the access point used to provide the secure MAC service to a MKA peer) using the current session key.

The MKA Protocol manages the encryption keys used by the underlying MACsec protocol. The basic requirements of MKA are defined in 802.1x-REV. The MKA Protocol extends 802.1x to allow peer discovery with confirmation of mutual authentication and sharing of MACsec secret keys to protect data exchanged by the peers.

The EAP framework implements MKA as a newly defined EAP-over-LAN (EAPOL) packet. EAP authentication produces a master session key (MSK) shared by both partners in the data exchange. Entering the EAP session ID generates a secure connectivity association key name (CKN). The device acts as the key server for both uplink and downlink; and acts as the authenticator for downlink. It generates a random secure association key (SAK), which is sent to the client partner. The client is never a key server and can only interact with a single MKA entity, the key server. After key derivation and generation, the device sends periodic transports to the partner at a default interval of 2 seconds.

The packet body in an EAPOL Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). MKA sessions and participants are deleted when the MKA lifetime (6 seconds) passes with no MKPDU received from a participant. For example, if a MKA peer disconnects, the participant on the device continues to operate MKA until 6 seconds have elapsed after the last MKPDU is received from the MKA peer.



Note Integrity check value (ICV) indicator in MKPDU is optional. ICV is not optional when the traffic is encrypted.

EAPoL Announcements indicate the use of the type of keying material. The announcements can be used to announce the capability of the supplicant as well as the authenticator. Based on the capability of each side, the largest common denominator of the keying material could be used.

Configuring MACsec Encryption

Configuring MKA and MACsec

MACsec is disabled by default. No MKA policies are configured.

Configuring an MKA Policy

Follow these steps to configure an MKA policy.

Note After changing any MKA policy or MACsec configuration for active sessions, execute the **shutdown** command, and then the **no shutdown** command on a port, so that the changes are applied to active sessions.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**mka policy** *policy name* **Example:** `Router(config)# mka policy mka_policy`	Identifies an MKA policy, and enters MKA policy configuration mode. The maximum policy name length is 16 characters. **Note** The default MACsec cipher
Step 4	**macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256**} **Example:** `Router(config-mka-policy)# macsec-cipher-suite gcm-aes-128`	Configures a cipher suite for deriving SAK with 128-bit or 256-bit encryption.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-mka-policy)# end`	Exit enters MKA policy configuration mode and returns to privileged EXEC mode.
Step 6	**show mka policy** **Example:** `Router# show mka policy`	Displays MKA policy configuration information.

Configuring MACsec MKA using PSK

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**key chain** *key-chain-name* **macsec** **Example:** `Router(config)# key chain keychain1 macsec`	Configures a key chain and enters the key chain configuration mode.
Step 4	**key** *hex string* **Example:** `Router(config-key-chain)# key 1000`	Configures a unique identifier for each key in the keychain and enters the keychain's key configuration mode. **Note** For 128-bit encryption, use any hex digit key-string.
Step 5	**cryptographic-algorithm** {**aes-128-cmac** | **aes-256-cmac**} **Example:** `Device(config-key-chain)# cryptographic-algorithm aes-128-cmac`	Set cryptographic authentication algorithm with 128-bit or 256-bit encryption.
Step 6	**key-string** { [**0**|**6**|**7**] *pwd-string* | *pwd-string*} **Example:**	Sets the password for a key string. Only hex characters must be entered.

	Command or Action	Purpose
	`Device(config-key-chain)# key-string 12345678901234567890123456789012`	
Step 7	**end** **Example:** `Router(config-key-chain)# end`	Exits key chain configuration mode and returns to privileged EXEC mode.

Configuring MACsec MKA on an Interface using PSK

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface GigabitEthernet 0/1/4`	Enters interface configuration mode.
Step 4	**switchport mode access** **Example:** `Router(config-if)# switchport mode access`	Sets the switchport mode to access.
Step 5	**switchport access vlan** *vlan-id* **Example:** `Router(config-if)# switchport access vlan 203`	Specifies the VLAN for which this access port will carry traffic.
Step 6	**macsec network-link** **Example:** `Router(config-if)# macsec network-link`	Enables MACsec on the interface.
Step 7	**mka policy** *policy name* **Example:** `Router(config-if)# mka policy MKA_128`	Configures an MKA policy.
Step 8	**mka pre-shared-key key-chain** *key-chain-name*	Configures an MKA pre-shared-key key-chain name.

	Command or Action	Purpose
	Example: `Router(config-if)# mka pre-shared-key key-chain KEY128`	**Note** The MKA pre-shared key can be
Step 9	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Example: Sample Configuration of Switch-to-Switch MACsec

This section shows sample configuration of switch-to-switch MACsec.

Device 1 Configuration

```
configure terminal
mka policy MKA_128
macsec-cipher-suite gcm-aes-128
key chain KEY128 macsec
key 1111
cryptographic-algorithm aes-128-cmac
key-string 11111111111111111111111111111111
end

configure terminal
interface Vlan203
ip address 22.1.1.1 255.255.255.0
end

configure terminal
interface GigabitEthernet0/1/4
switchport mode access
switchport access vlan 203
mka policy MKA_128
mka pre-shared-key key-chain KEY128
macsec network-link
no shutdown
end
```

Device 2 Configuration

```
configure terminal
mka policy MKA_128
macsec-cipher-suite gcm-aes-128
key chain KEY128 macsec
key 1111
cryptographic-algorithm aes-128-cmac
key-string 11111111111111111111111111111111
end

configure terminal
interface Vlan203
ip address 22.1.1.2 255.255.255.0
end

configure terminal
interface GigabitEthernet0/1/4
switchport mode access
```

```
switchport access vlan 203
mka policy MKA_128
mka pre-shared-key key-chain KEY128
macsec network-link
no shutdown
end
```

Example: Sample Configuration of Switch-to-Host MACsec

This section shows sample configuration of switch-to-host MACsec.

```
aaa new-model
!
!
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa authorization auth-proxy default group radius
aaa accounting network default start-stop group radius
aaa accounting system default start-stop group radius
!
!
aaa server radius dynamic-author
client <radius-server ip> server-key <key>
port 3799
!
aaa session-id common
!
!
ip dhcp-server 17.0.0.1
!
ip dhcp pool VLAN17
network 17.0.0.0 255.255.255.0
default-router 17.0.0.1
!
!
device-tracking policy DEVICE_MACSEC
no protocol udp
tracking enable
!
authentication critical recovery delay 300
!
service-template DEFAULT_LINKSEC_POLICY_MUST_SECURE
linksec policy must-secure
!
dot1x system-auth-control
!
class-map type control subscriber match-all DOT1X
match method dot1x
!
class-map type control subscriber match-all DOT1X_FAILED
match method dot1x
match result-type method dot1x authoritative
!
class-map type control subscriber match-all DOT1X_MEDIUM_PRIO
match authorizing-method-priority gt 20
!
class-map type control subscriber match-all DOT1X_NO_RESP
match method dot1x
match result-type method dot1x agent-not-found
!
class-map type control subscriber match-all DOT1X_TIMEOUT
match method dot1x
match result-type method dot1x method-timeout
```

```
!
class-map type control subscriber match-all LINKSEC_FAIL_DOT1X
match authorization-fail linksec-failed
match method dot1x
!
class-map type control subscriber match-all LINKSEC_FAIL_MAB
match authorization-fail linksec-failed
match method mab
!
class-map type control subscriber match-all MAB_FAILED
match method mab
match result-type method mab authoritative
!

policy-map type control subscriber POLICY_MUSTSECURE
event session-started match-all
10 class always do-until-failure
10 authenticate using dot1x priority 10
event authentication-failure match-first
5 class DOT1X_FAILED do-until-failure
10 terminate dot1x
20 authenticate using mab priority 20
10 class DOT1X_NO_RESP do-until-failure
10 terminate dot1x
20 authenticate using mab priority 20
20 class MAB_FAILED do-until-failure
10 terminate mab
20 authentication-restart 60
40 class always do-until-failure
10 terminate dot1x
20 terminate mab
30 authentication-restart 60
event agent-found match-all
10 class always do-until-failure
10 terminate mab
20 authenticate using dot1x priority 10
event authentication-success match-all
10 class always do-until-failure
10 activate service-template DEFAULT_LINKSEC_POLICY_MUST_SECURE
!
interface GigabitEthernet0/1/1
switchport access vlan 17
switchport mode access
device-tracking attach-policy DEVICE_MACSEC
macsec
access-session host-mode multi-host
access-session closed
access-session port-control auto
dot1x pae authenticator
service-policy type control subscriber POLICY_SHOULDSECURE
!
!
radius-server attribute 8 include-in-access-req
radius-server dead-criteria time 20 tries 2
radius-server timeout 20
radius-server deadtime 5
radius-server key <radius key>

radius server ACS
address ipv4 <radius-server ip> auth-port 1812 acct-port 1813
!
```

Verifying the Configuration

Use the following command to display MKA sessions on the interface:

```
# show mka sessions interface GigabitEthernet0/1/4

Summary of All Currently Active MKA Sessions on Interface GigabitEthernet0/1/4...

===============================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN
===============================================================================================
Gi0/1/4        b08b.d071.86ac/000e POLICY           NO                YES
14             b08b.d071.86ac/0000 0                Init              1000
```

Use the following command to display MACsec status on the interface:

```
# show macsec status interface GigabitEthernet0/1/5
 MACsec is enabled
  Replay protect : enabled
  Replay window : 0
  Include SCI : yes
  Use ES Enable : no
  Use SCB Enable : no
  Admin Pt2Pt MAC : forceTrue(1)
  Pt2Pt MAC Operational : no
  Cipher : GCM-AES-128
  Confidentiality Offset : 0

 Capabilities
  ICV length : 16
  Data length change supported: yes
  Max. Rx SA : 16
  Max. Tx SA : 16
  Max. Rx SC : 8
  Max. Tx SC : 8
  Validate Frames : strict
  PN threshold notification support : Yes
  Ciphers supported : GCM-AES-128
                      GCM-AES-256
```

Use the following command to display MACsec statistics on the interface:

```
# show macsec statistics interface GigabitEthernet0/1/5
 Transmit Secure Channels
  SCI : B08BD07186AD000F
  SC state : inUse(1)
   Elapsed time : 00:00:18
   Start time : 7w0d
   Current AN: 0
   Previous AN: -
   Next PN: 14
   SA State: inUse(1)
   Confidentiality : yes
   SAK Unchanged : yes
   SA Create time : 00:11:21
   SA Start time : 7w0d
   SC Statistics
    Auth-only Pkts : 0
    Auth-only Bytes : 0
    Encrypt Pkts : 0
    Encrypt Bytes : 0
   SA Statistics
    Auth-only Pkts : 0
```

```
    Encrypt Pkts : 13

  Port Statistics
   Egress untag pkts  0
   Egress long pkts  0

 Receive Secure Channels
  SCI : 0C75BDCC84A40007
  SC state : inUse(1)
   Elapsed time : 00:00:18
   Start time : 7w0d
   Current AN: 0
   Previous AN: -
   Next PN: 23
   RX SA Count: 0
   SA State: inUse(1)
   SAK Unchanged : yes
   SA Create time : 00:11:19
   SA Start time : 7w0d
   SC Statistics
    Notvalid pkts 0
    Invalid pkts 0
    Valid pkts 0
    Valid bytes 0
    Late pkts 0
    Uncheck pkts 0
    Delay pkts 0
    UnusedSA pkts 0
    NousingSA pkts 0
    Decrypt bytes 0
   SA Statistics
    Notvalid pkts 0
    Invalid pkts 0
    Valid pkts 21
    UnusedSA pkts 0
    NousingSA pkts 0

  Port Statistics
   Ingress untag pkts  0
   Ingress notag pkts  9
   Ingress badtag pkts  0
   Ingress unknownSCI pkts  0
   Ingress noSCI pkts  0
   Ingress overrun pkts  0
```

Use the following command to display detailed status for MKA session:

```
# show mka session interface GigabitEthernet0/1/5 detail

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI..............  b08b.d071.86ad/000f
Interface MAC Address....  b08b.d071.86ad
MKA Port Identifier.......  15
Interface Name............  GigabitEthernet0/1/5
Audit Session ID.........
CAK Name (CKN)............  1111
Member Identifier (MI)...  CBA23DA1D3D77DF725CD43BB
Message Number (MN).......  10
EAP Role..................  NA
Key Server................  NO
MKA Cipher Suite.........  AES-128-CMAC
```

```
Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... AD657F4EB0D237F5AFF9186F00000001 (1)
Old SAK Status........... FIRST-SAK
Old SAK AN............... 0
Old SAK KI (KN).......... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time.......... 0s (No Old SAK to retire)
SAK Rekey Time........... 0s (SAK Rekey interval not applicable)

MKA Policy Name.......... Policy_128
Key Server Priority...... 0
Delay Protection......... NO
Delay Protection Timer.......... 0s (Not enabled)

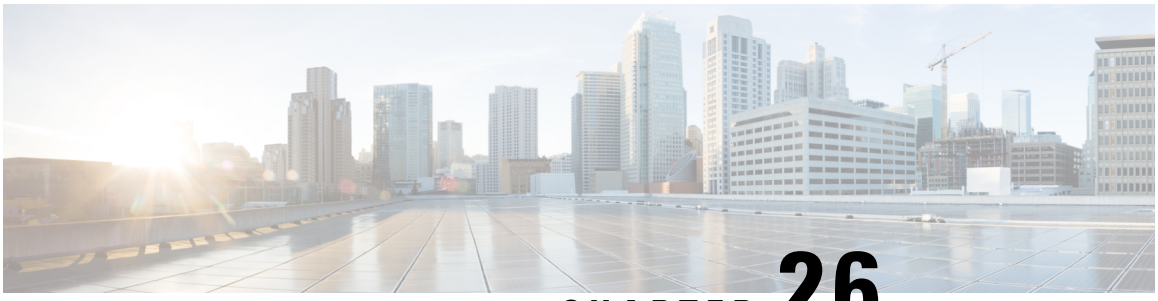
Confidentiality Offset... 0
Algorithm Agility........ 80C201
SAK Rekey On Live Peer Loss........ NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation.... NO
SAK Cipher Suite......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES

# of MACsec Capable Live Peers............. 1
# of MACsec Capable Live Peers Responded.. 0

Live Peers List:
  MI                       MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                            Priority  Installed
  ----------------------------------------------------------------------------------------
  AD657F4EB0D237F5AFF9186F  8          0c75.bdcc.84a4/0007  0         YES            0

Potential Peers List:
  MI                       MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                            Priority  Installed
  ----------------------------------------------------------------------------------------

Dormant Peers List:
  MI                       MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                            Priority  Installed
  ----------------------------------------------------------------------------------------
```



CHAPTER 26

WAN MACSEC and MKA Support Enhancements

The WAN MACsec and MKA feature introduces MACsec support on WAN, and uplink support and Pre-shared key support for the Macsec Key Agreement protocol (MKA).

- MACsec and MKA Overview, on page 333
- Benefits of WAN MACsec and MKA Support Enhancements, on page 334
- Best Practices for Implementing WAN MACsec and MKA Support Enhancements, on page 334
- Limitations in supported configurations, on page 335
- MKA Policy Inheritance, on page 335
- Key Lifetime and Hitless Key Rollover, on page 335
- Encryption Algorithms for Protocol Packets, on page 335
- Access Control Option for Smoother Migration, on page 336
- Extensible Authentication Protocol over LAN Destination Address, on page 337
- How to Configure WAN MACsec and MKA Support Enhancements, on page 338

MACsec and MKA Overview

MACsec is an IEEE 802.1AE standards based Layer 2 hop-by-hop encryption that provides data confidentiality and integrity for media access independent protocols.

MACsec, provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys. Only host facing links (links between network access devices and endpoint devices such as a PC or IP phone) can be secured using MACsec.

The 802.1AE encryption with MACsec Key Agreement (MKA) is supported on downlink ports for encryption between the routers or switches and host devices.

MACsec encrypts the entire data except for the Source and Destination MAC addresses of an Ethernet packet.

To provide MACsec services over the WAN, service providers offer Layer 2 transparent services such as E-Line or E-LAN using various transport layer protocols such as Ethernet over Multiprotocol Label Switching (EoMPLS) and L2TPv3.

The packet body in an EAP-over-LAN (EAPOL) Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). When no MKPDU is received from a participants after 3 hearbeats (each hearbeat is of 2 seconds), peers are deleted from the live peer list For example, if a client disconnects, the participant on the switch continues to operate MKA until 3 heartbeats have elapsed after the last MKPDU is received from the client.

The MKA feature support provides tunneling information such as VLAN tag (802.1Q tag) in the clear so that the service provider can provide service multiplexing such that multiple point to point or multipoint services can co-exist on a single physical interface and differentiated based on the now visible VLAN ID.

In addition to service multiplexing, VLAN tag in the clear also enables service providers to provide quality of service (QoS) to the encrypted Ethernet packet across the SP network based on the 802.1P (CoS) field that is now visible as part of the 802.1Q tag.

Benefits of WAN MACsec and MKA Support Enhancements

- Support for Point-to-point (P2P) deployment models.
- Support for Point-to-Multipoint (P2MP) deployment models.
- Support for multiple P2P and P2MP deployments on the same physical interface.
- Support for 128- and 256-bit Advanced Encryption Standard–Galois Counter Mode (AES-GCM) encryption for data packets.
- Support for 128- and 256-bit Advanced Encryption Standard-Cipher-based Message Authentication Code (AEC-CMAC) encryption for control packets.
- Support for VLAN tag in the clear option to enable Carrier Ethernet Service Multiplexing.
- Support for coexisting of MACsec and Non-MACsec subinterfaces.
- Support for configurable Extensible Authentication Protocol over LAN (EAPoL) destination address.
- Support for configurable option to change the EAPoL Ethernet type.
- Support for configurable replay protection window size to accommodate packet reordering in the service provider network.

Best Practices for Implementing WAN MACsec and MKA Support Enhancements

- Ensure basic Layer 2 Ethernet connectivity is established and verified before attempting to enable MACsec. Basic ping between the customer edge devices must work.
- When you are configuring WAN MACsec for the first time, ensure that you have out of band connectivity to the remote site to avoid locking yourself out after enabling MACsec, if the session fails to establish.
- We recommend that you configure the **access-control should-secure** command while enabling MACsec for the first time and subsequently remove the command to change to default **access-control must-secure** , once the session establishment is successful, unless it is needed for migration.
- We recommend that you configure an interface MTU, adjusting it for MACsec overhead, for example, 32 bytes. Although MACsec encryption and decryption occurs at the physical level and MTU is size does not effect the source or destination router, it may effect the intermediate service provider router. Configuring an MTU value at the interface allows for MTU negotiation that includes MACsec overhead.

Limitations in supported configurations

When configuring your network, observe these constraints:

- **Dot1q-in-clear 2**: This configuration is not supported.
- **MACsec on QinQ**: The system does not support MACsec encryption for double-tagged (QinQ) frames.

MKA Policy Inheritance

On WAN routers, MKA policy is inherited and also it has a default value. When a new session is started, the following rules apply:

- If an MKA policy is configured on a subinterface, it will be applied when an MKA session is started.
- If an MKA policy is not configured on a subinterface, a policy that is configured on the physical interface is be applied at session start.
- If a MKA policy is not configured on a subinterface or physical interface, default policy is applied at session start.

Key Lifetime and Hitless Key Rollover

A MACsec key chain can have multiple pre-shared keys (PSK) each configured with a key id and an optional lifetime. A key lifetime specifies at which time the key 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 key chain after the lifetime is expired. Time zone of the key can be local or UTC. Default time zone is UTC.

Use the **key chain** *name* **macsec** to configure the MACsec key chain.

You can Key rolls over to the next key within the same key chain by configuring a second key in the key chain 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, key rolls over without traffic interruption.



Note The lifetime of the keys need to be overlapped in order to achieve hitless key rollover.

Encryption Algorithms for Protocol Packets

Cryptographic Algorithm selection for MKA control protocol packets encryption is as follows:

- Cryptographic Algorithm to encrypt MKA control protocol packets is configured as part of the key chain. There can be only one cryptographic algorithm configured per key chain.
- A key server uses the configured MKA cryptographic algorithm from the key chain that is used.

- All nonkey servers must use the same cryptographic algorithm as the key server.

If an MKA cryptographic algorithm is not configured, a default cryptographic algorithm of AES-CMAC-128 (Cipher-based Message Authentication Code with 128-bit Advanced Encryption Standard) is used.

Encryption algorithm for Data packets:

```
mka policy p1
macsec-cipher-suite [gcm-aes-128 | gcm-aes-256
```

Encryption algorithm for MKA Control packets

```
key chain <name> macsec
key 01
key-string <Hex string>
cryptographic-algorithm [aes-256-cmac  | aes-128-cmac]
```

It is recommended to change data packets cipher suite in the key server for the cipher suite rollover to be seamless, if the nonkey servers have the same cipher-suite configured in the list or is with default configuration.

Access Control Option for Smoother Migration

When MACsec is enabled on an interface, the entire interface traffic is secured, by default. MACsec does not allow any unencrypted packets to be transmitted or received from the same physical interface. However, to enable MACsec on selected subinterfaces, an additional Cisco proprietary extension has been implemented to allow unencrypted packets to be transmitted or received from the same physical interface.

Use the **macsec access-control** {**must-secure** | **should-secure**} command to control the behavior of unencrypted packets.

- The **should-secure** keyword allows unencrypted packets from the physical interface or subinterfaces to be transmitted or received.
- The **must-secure** keyword does not allow unencrypted packets from physical interface or subinterfaces to be transmitted or received. All such packets are dropped except for MKA control protocol packets
- If MACsec is enabled only on selected subinterfaces, configure the **should-secure** keyword option on the corresponding interface.

The default configuration for MACsec on subinterfaces is **macsec access-control must-secure** . This option is enabled by default when the **macsec** command is configured on an interface.

Note The **macsec access-control should-secure** command can be configured only at the interface level and not the subinterface. Configuring this command allows unencrypted traffic on a secured MACsec session.

Note For non-MACsec subinterface, you must configure the **should-secure** option for traffic to pass.

Extensible Authentication Protocol over LAN Destination Address

Before establishing a MACsec secure session, MKA (MACsec Key Agreement) is used as the control protocol. MKA selects the cipher suite to be used for encryption and to exchange the required keys and parameters between peers.

MKA uses Extensible Authentication Protocol over LAN (EAPoL) as the transport protocol to transmit MKA messages. By default, EAPoL uses a destination multicast MAC address of 01:80:c2:00:00:03 to multicast packets to multiple destinations. EAPoL is a standards-based protocol and other authentication mechanisms such as IEEE 802.1X also use the same protocol. Devices in the service provider cloud might consume this packet (based on the destination multicast MAC address), and try to process the EAPoL packet and eventually drop the packet. This causes MKA session to fail.

Use the **eapol destination-address** command to change the destination MAC address of an EAPoL packet that is transmitted on an interface towards the service provider. This ensures that the service provider tunnels the packet like any other data packet instead of consuming them.

Note The EAPoL destination address can be configured independently on either physical or subinterface level. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on the subinterface overrides the inherited value or policy for that subinterface.

Replay Protection Window Size

Replay protection is a feature provided by MACsec to counter replay attacks. Each encrypted packet is assigned a unique sequence number and the sequence is verified at the remote end. Frames transmitted through a Metro Ethernet service provider network are highly susceptible to reordering due to prioritization and load balancing mechanisms used within the network.

A replay window is necessary to support use of MACsec over provider networks that reorder frames. Frames within the window can be received out of order, but are not replay protected. The default window size is set to 64. Use the **macsec replay-protection window-size** command to change the replay window size. The range for window size is 0 to 4294967295.

The replay protection window may be set to zero to enforce strict reception ordering and replay protection.

Note A replay protection window can be configured independently on either physical interface or subinterface. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on subinterface overrides the inherited value or policy for that sub-interface.

How to Configure WAN MACsec and MKA Support Enhancements

Configuring MKA

The MACsec Key Agreement (MKA) enables configuration and control of keying parameters. Perform the following task to configure MKA.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**mka policy** *policy-name* **Example:** `Router(config)# mka policy MKAPolicy`	Configures an MKA policy.
Step 4	**include-icv-indicator** **Example:** `Device(config-mka-policy)# include-icv-indicator`	(Optional) Includes ICV indicator in MKPDU.
Step 5	**key-server priority** *key-server-priority* **Example:** `Router(config-mka-policy)# key-server priority 200`	(Optional) Configures MKA key server priority.
Step 6	**macsec-cipher-suite {gcm-aes-128 | gcm-aes-256}** **Example:** `Router(config-mka-policy)# macsec-cipher-suite gcm-aes-128 gcm-aes-256`	(Optional) Configures cipher suite(s) for secure association key (SAK) derivation. Each of the cipher suite options can be repeated only once, but they can be used in any order.
Step 7	**sak-rekey interval** *interval* **Example:** `Device(config-mka-policy)# sak-rekey interval 30`	(Optional) Sets the SAK rekey interval (in seconds). The range is from 30 to 65535, and the default value is 0. The SAK rekey timer does not start by default until it is configured.

	Command or Action	Purpose
		• To stop the SAK rekey timer, use the **no sak-rekey interval** command under the defined MKA policy.
Step 8	**confidentiality-offset** *value* **Example:** `Router(config-mka-policy)# confidentiality-offset 30`	(Optional) Configures confidentiality offset for MACsec operation.
Step 9	**end** **Example:** `Router(config-mka-policy)# end`	Returns to privileged EXEC mode.

Configuring MKA Pre-Shared Key

Perform the following task to configure MACsec Key Agreement (MKA) pre-shared key.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**Key chain** *key-chain-name* [**macsec**] **Example:** `Router(config)# Key chain keychain1 macsec`	Configures a key chain and enters keychain configuration mode.
Step 4	**key** *hex string* **Example:** `Router(config-keychain)# key 9ABCD`	Configures a key and enters keychain key configuration mode.
Step 5	**cryptographic-algorithm** {**gcm-aes-128** | **gcm-aes-256**} **Example:** `Router(config-keychain-key)# cryptographic-algorithm gcm-aes-128`	Set cryptographic authentication algorithm.

	Command or Action	Purpose
Step 6	**key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*} **Example:** `Router(config-keychain-key)# key-string 0 pwd`	Sets the password for a key string.
Step 7	**lifetime local** {{*day month year* **duration** *seconds*} **Example:** `Device(config-keychain-key)# lifetime local 16:00:00 Nov 9 2014 duration 6000`	Sets the lifetime for a key string. The range you can specify for the duration is between 1 and 864000 seconds.
Step 8	**end** **Example:** `Router(config-keychain-key)# end`	Returns to privileged EXEC mode.

Configuring an Option to Change the EAPoL Ethernet Type

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Enters interface configuration mode.
Step 4	**eapol** *eth-type* **Example:** `Router(config-if)# eapol eth-type 876F`	Configures an ethernet type (Hexadecimal) for the EAPoL Frame on the interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Sample Configuration for Point-to-Point WAN MACsec

Example: Port Based WAN MACsec

Device1:

```
configure terminal
    key chain test128 macsec
    key 1111
   cryptographic-algorithm aes-128-cmac
   key-string 11111111111111111111111111111111
   exit

   mka policy test
   macsec-cipher-suite gcm-aes-128 gcm-aes-256
   exit

   interface GigabitEthernet0/0/0
   ip address 1.1.1.1 255.255.255.0
   mka policy test
   mka pre-shared-key key-chain test128
   macsec
   exit
```

Device2:

```
configure terminal
    key chain test128 macsec
    key 1111
   cryptographic-algorithm aes-128-cmac
   key-string 11111111111111111111111111111111
   exit

   mka policy test
   macsec-cipher-suite gcm-aes-128 gcm-aes-256
   exit

   interface GigabitEthernet0/0/0
   ip address 1.1.1.2 255.255.255.0
   mka policy test
   mka pre-shared-key key-chain test128
   macsec
   exit
```

Example: VLAN Based WAN MACsec

Device1:

```
configure terminal
    key chain test128 macsec
    key 1111
   cryptographic-algorithm aes-128-cmac
   key-string 11111111111111111111111111111111
   exit

   mka policy test
   macsec-cipher-suite gcm-aes-128 gcm-aes-256
   exit

   interface GigabitEthernet0/0/0
   eapol destination-address broadcast-address
   mka policy test
```

```
   macsec dot1q-in-clear 1

   interface GigabitEthernet0/0/0.1
   encapsulation dot1q 10
   ip address 1.1.1.1 255.255.255.0
   mka pre-shared-key key-chain test128
    macsec
    exit
```

Device2:

```
configure terminal
    key chain test128 macsec
    key 1111
   cryptographic-algorithm aes-128-cmac
   key-string 11111111111111111111111111111111
   exit

   mka policy test
   macsec-cipher-suite gcm-aes-128 gcm-aes-256
   exit

   interface GigabitEthernet0/0/0
   eapol destination-address broadcast-address
   mka policy test
   macsec dot1q-in-clear 1

   interface GigabitEthernet0/0/0.1
   encapsulation dot1q 10
   ip address 1.1.1.2 255.255.255.0
   mka pre-shared-key key-chain test128
   macsec
   exit
```

Sample Show Command Output for Port Based WAN MACsec

```
#show mka sessions

Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

=============================================================================================
Interface       Local-TxSCI          Policy-Name       Inherited          Key-Server

Port-ID         Peer-RxSCI           MACsec-Peers      Status             CKN

=============================================================================================
Gi0/0/0         b08b.d071.86a0/0007  test              NO                 YES

7               b08b.d079.8a10/0007  1                 Secured            1111



#show mka sessions detail

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI.............. b08b.d071.86a0/0007
Interface MAC Address.... b08b.d071.86a0
```

```
MKA Port Identifier...... 7
Interface Name........... GigabitEthernet0/0/0
Audit Session ID.........
CAK Name (CKN)........... 1111
Member Identifier (MI)... 58D1ED5150ED8811970C4DB9
Message Number (MN)...... 18
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... 58D1ED5150ED8811970C4DB900000001 (1)
Old SAK Status........... FIRST-SAK
Old SAK AN............... 0
Old SAK KI (KN).......... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time.......... 0s (No Old SAK to retire)
SAK Rekey Time........... 0s (SAK Rekey interval not applicable)

MKA Policy Name.......... test
Key Server Priority...... 0
Delay Protection......... NO
Delay Protection Timer.......... 0s (Not enabled)

Confidentiality Offset... 0
Algorithm Agility........ 80C201
SAK Rekey On Live Peer Loss........ NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation.... NO
SAK Cipher Suite......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES

# of MACsec Capable Live Peers............ 1
# of MACsec Capable Live Peers Responded.. 1

Live Peers List:
  MI                        MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                             Priority  Installed
  ----------------------------------------------------------------------------------------
  5225C3DE8DD5F35A3B55830C  53          b08b.d079.8a10/0007  0         YES            0

Potential Peers List:
  MI                        MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                             Priority  Installed
  ----------------------------------------------------------------------------------------

Dormant Peers List:
  MI                        MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                             Priority  Installed
  ----------------------------------------------------------------------------------------
```

```
#show macsec statistics interface gigabitEthernet 0/0/0
MACsec Statistics for GigabitEthernet0/0/0
 SecY Counters
  Ingress Untag Pkts:       0
  Ingress No Tag Pkts:      0
  Ingress Bad Tag Pkts:     0
```

```
 Ingress Unknown SCI Pkts: 0
 Ingress No SCI Pkts:      0
 Ingress Overrun Pkts:     0
 Ingress Validated Octets: 0
 Ingress Decrypted Octets: 966
 Egress Untag Pkts:        1
 Egress Too Long Pkts:     0
 Egress Protected Octets:  0
 Egress Encrypted Octets:  1387

Controlled Port Counters
 IF In Octets:             1086
 IF In Packets:            10
 IF In Discard:            0
 IF In Errors:             0
 IF Out Octets:            1519
 IF Out Packets:           11
 IF Out Errors:            0

Transmit SC Counters (SCI: B08BD07186A00007)
 Out Pkts Protected:       0
 Out Pkts Encrypted:       11
Transmit SA Counters (AN 0)
 Out Pkts Protected:       0
 Out Pkts Encrypted:       11

Receive SA Counters (SCI: B08BD0798A100007  AN 0)
 In Pkts Unchecked:        0
 In Pkts Delayed:          0
 In Pkts OK:               10
 In Pkts Invalid:          0
 In Pkts Not Valid:        0
 In Pkts Not using SA:     0
 In Pkts Unused SA:        0
 In Pkts Late:             0
```

Sample Show Command Output for VLAN Based WAN MACsec

```
#show mka sessions

Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

=============================================================================================================
Interface       Local-TxSCI          Policy-Name       Inherited          Key-Server

Port-ID         Peer-RxSCI           MACsec-Peers      Status             CKN

=============================================================================================================
Gi0/0/0.1       b08b.d071.86a0/0024  test              YES                YES

36              b08b.d079.8a10/001a  1                 Secured            1111



#show mka sessions detail

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI.............. b08b.d071.86a0/0024
```

```
Interface MAC Address.... b08b.d071.86a0
MKA Port Identifier...... 36
Interface Name........... GigabitEthernet0/0/0.1
Audit Session ID.........
CAK Name (CKN)........... 1111
Member Identifier (MI)... 920AF42A7C0F5D2BDC0CBB99
Message Number (MN)...... 35
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 1
Latest SAK KI (KN)....... 920AF42A7C0F5D2BDC0CBB9900000002 (2)
Old SAK Status........... No Rx, No Tx
Old SAK AN............... 0
Old SAK KI (KN).......... RETIRED (1)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time.......... 0s (No Old SAK to retire)
SAK Rekey Time........... 0s (SAK Rekey interval not applicable)

MKA Policy Name.......... test (inherited)
Key Server Priority...... 0
Delay Protection......... NO
Delay Protection Timer.......... 0s (Not enabled)

Confidentiality Offset... 0
Algorithm Agility........ 80C201
SAK Rekey On Live Peer Loss........ NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation.... NO
SAK Cipher Suite......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES

# of MACsec Capable Live Peers............ 1
# of MACsec Capable Live Peers Responded.. 1

Live Peers List:
  MI                        MN          Rx-SCI (Peer)         KS        RxSA          SSCI
                                                              Priority  Installed
  ------------------------------------------------------------------------------------------
  E500D4C281BE01C8777901D4  18          b08b.d079.8a10/001a  0         YES            0

Potential Peers List:
  MI                        MN          Rx-SCI (Peer)         KS        RxSA          SSCI
                                                              Priority  Installed
  ------------------------------------------------------------------------------------------

Dormant Peers List:
  MI                        MN          Rx-SCI (Peer)         KS        RxSA          SSCI
                                                              Priority  Installed
  ------------------------------------------------------------------------------------------
```

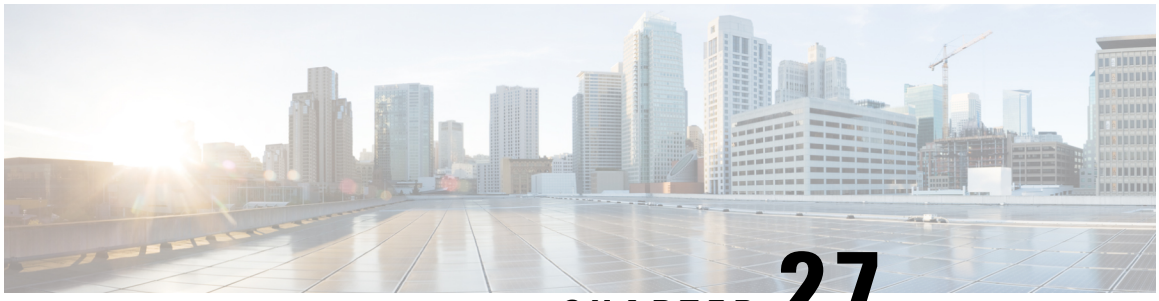
```
#show macsec statistics interface gigabitEthernet 0/0/0.1
MACsec Statistics for GigabitEthernet0/0/0.1
 SecY Counters
  Ingress Untag Pkts:       0
  Ingress No Tag Pkts:      0
  Ingress Bad Tag Pkts:     0
  Ingress Unknown SCI Pkts: 0
```

```
 Ingress No SCI Pkts:      0
 Ingress Overrun Pkts:     0
 Ingress Validated Octets: 0
 Ingress Decrypted Octets: 4038
 Egress Untag Pkts:        1
 Egress Too Long Pkts:     0
 Egress Protected Octets:  0
 Egress Encrypted Octets:  4476

Controlled Port Counters
 IF In Octets:             4926
 IF In Packets:            74
 IF In Discard:            0
 IF In Errors:             0
 IF Out Octets:            5460
 IF Out Packets:           82
 IF Out Errors:            0

Transmit SC Counters (SCI: B08BD07186A00024)
 Out Pkts Protected:       0
 Out Pkts Encrypted:       82
Transmit SA Counters (AN 1)
 Out Pkts Protected:       0
 Out Pkts Encrypted:       82

Receive SA Counters (SCI: B08BD0798A10001A  AN 1)
 In Pkts Unchecked:        0
 In Pkts Delayed:          0
 In Pkts OK:               74
 In Pkts Invalid:          0
 In Pkts Not Valid:        0
 In Pkts Not using SA:     0
 In Pkts Unused SA:        0
 In Pkts Late:             0
```



CHAPTER 27

Configuring IPv6 First Hop Security

- IPv6 First Hop Security Overview, on page 347
- Configuring an IPv6 DHCP Guard Policy, on page 348
- Configuring an IPv6 Router Advertisement Guard Policy, on page 350

IPv6 First Hop Security Overview

First Hop Security in IPv6 (FHS IPv6) is a set of IPv6 security features, whose policies can be attached to an interface or a VLAN. The following IPv6 policies are supported:

- DHCPv6 Guard
- IPv6 Router Advertisement (RA) Guard

Overview of DHCPv6 Guard

The DHCPv6 Guard feature blocks reply and advertisement messages that come from unauthorized DHCPv6 servers and relay agents. IPv6 DHCP guard can prevent forged messages from being entered in the binding table and block DHCPv6 server messages when they are received on ports that are not explicitly configured as facing a DHCPv6 server or DHCP relay. To use this feature, configure a policy and attach it to an interface or a VLAN.

To debug DHCP guard packets, use the following privileged EXEC command.

debug ipv6 snooping dhcp-guard [**filter** <*name*>] [**interface** <*interface-id*>] [**vlan** <*vlanid*>]

Restrictions of DHCPv6 Guard

The DHCPv6 guard feature is not supported on EtherChannel ports.

Overview of IPv6 RA Guard

The IPv6 Router Advertisement (RA) guard feature enables the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network device platform. RAs are used by devices to announce themselves on the link. The RA Guard feature analyzes the RAs and filters out bogus RAs sent by unauthorized devices. In host mode, all router advertisement and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 device with the

information found in the received RA frame. Once the Layer 2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.

To debug RA guard packets, use the following privileged EXEC command.

debug ipv6 snooping raguard [**filter** <*name*>] [**interface** <*interface-id*>] [**vlan** <*vlanid*>]

Limitations and Restrictions of IPv6 RA Guard

- The IPv6 RA Guard feature does not offer protection in environments where IPv6 traffic is tunneled.
- 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 not supported on EtherChannel and EtherChannel port members.
- This feature is not supported on trunk ports with merge mode.
- 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.
- If the **platform ipv6 acl icmp optimize neighbor-discovery** command is configured, the IPv6 RA Guard feature cannot be configured and an error message will be displayed. This command adds default global Internet Control Message Protocol (ICMP) entries that will override the RA guard ICMP entries.

Configuring an IPv6 DHCP Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 DHCP (DHCPv6) Guard policy:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 dhcp guard policy** *policy-name* **Example:** `Router(config)# ipv6 dhcp guard policy policy1`	Specifies the DHCPv6 Guard policy name and enters DHCPv6 Guard Policy configuration mode.
Step 4	**device-role** {**client** | **monitor** | **server**} **Example:** `Router(config-dhcp-guard)# device-role server`	(Optional) Filters out DHCPv6 replies and DHCPv6 advertisements on the port that are not from a device of the specified role. Default is **client**. • **client**—Default value, specifies that the attached device is a client. Server messages are dropped on this port. • **server**—Specifies that the attached device is a DHCPv6 server. Server messages are allowed on this port.
Step 5	**trusted-port** **Example:** `Router(config-dhcp-guard)# trusted-port`	(Optional) **trusted-port**—Sets the port to a trusted mode. No further policing takes place on the port. **Note** If you configure a trusted po
Step 6	**end** **Example:** `Router(config-dhcp-guard)# end`	Exits DHCPv6 Guard Policy configuration mode and returns to privileged EXEC mode.
Step 7	**show ipv6 dhcp guard policy** *policy_name* **Example:** `Router# show ipv6 dhcp guard policy policy1`	(Optional) Displays the configuration of the IPv6 DHCP guard policy. Omitting the *policy_name* variable displays all DHCPv6 policies.

Attaching an IPv6 DHCP Guard Policy to an Interface or a VLAN

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/1/0`	Specifies an interface type and identifier, and enters interface configuration mode.
Step 4	**ipv6 dhcp guard** [**attach-policy** *policy_name*] **Example:** `Router(config-if)# ipv6 dhcp guard attach-policy policy1`	Attaches the DHCP Guard policy to the interface or the specified VLAN.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show ipv6 dhcp guard policy** *policy_name* **Example:** `Router# show ipv6 dhcp guard policy1`	(Optional) Displays the policy configuration as well as all the interfaces where the policy is applied.

Configuring an IPv6 Router Advertisement Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 Router Advertisement policy:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**ipv6 nd raguard policy** *policy-name* **Example:** `Router(config)# ipv6 nd raguard policy example_policy`	Specifies the RA guard policy name and enters RA guard policy configuration mode.

	Command or Action	Purpose
Step 4	**device-role {host | monitor | router | switch}** **Example:** `Router(config-nd-raguard)# device-role switch`	Specifies the role of the device attached to the port. The default is **host**. **Note** For a network with both hos with **device-role host** on ho **device-role router** on route
Step 5	**hop-limit** {**maximum** | **minimum**} *value* **Example:** `Router(config-nd-raguard)# hop-limit maximum 33`	Enables filtering of Router Advertisement messages by the Hop Limit value. A rogue RA message may have a low Hop Limit value (equivalent to the IPv4 Time to Live) that when accepted by the host, prevents the host from generating traffic to destinations beyond the rogue RA message generator. An RA message with an unspecified Hop Limit value is blocked. (1–255) Range for Maximum and Minimum Hop Limit values. If not configured, this filter is disabled. Configure **minimum** to block RA messages with Hop Limit values lower than the value you specify. Configure **maximum**to block RA messages with Hop Limit values greater than the value you specify.
Step 6	**managed-config-flag** {**off** | **on**} *value* **Example:** `Router(config-nd-raguard)# managed-config-flag on`	Enables filtering of Router Advertisement messages by the managed address configuration, or "M" flag field. A rouge RA message with an M field of 1 can cause a host to use a rogue DHCPv6 server. If not configured, this filter is disabled. **On**—Accepts and forwards RA messages with an M value of 1, blocks those with 0. **Off**—Accepts and forwards RA messages with an M value of 0, blocks those with 1.
Step 7	**other-config-flag** {**on** | **off**} **Example:** `Device(config-ra-guard)# other-config-flag on`	(Optional) Enables verification of the advertised “other” configuration parameter.
Step 8	**router-preference maximum** {**high** | **medium** | **low**} **Example:**	Enables filtering of Router Advertisement messages by the router preference flag. If not configured, this filter is disabled.

	Command or Action	Purpose
	`Router(config-nd-raguard)# router-preference maximum high`	• **high**—Accepts RA messages with the router preference set to high, medium, or low. • **medium**—Blocks RA messages with the router preference set to high. • **low**—Blocks RA messages with the router preference set to medium and high.
Step 9	**match** {**ipv6 access-list** *list* | **ra prefix-list** *list*} **Example:** `Router(config-nd-raguard)# match ipv6 access-list example_list`	Matches a specified prefix list or access list.
Step 10	**trusted-port** **Example:** `Router(config-nd-raguard)# trusted-port`	When configured as a trusted port, all attached devices are trusted, and no further message verification is performed.
Step 11	**end** **Example:** `Router(config-nd-raguard)# end`	Exits RA Guard policy configuration mode and returns to privileged EXEC mode.
Step 12	**show ipv6 nd raguard policy** *policy_name* **Example:** `Router# show ipv6 nd raguard policy example_policy`	(Optional)—Displays the ND guard policy configuration.

Attaching an IPv6 Router Advertisement Guard Policy to an Interface or a VLAN

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement policy to an interface or to VLANs on the interface :

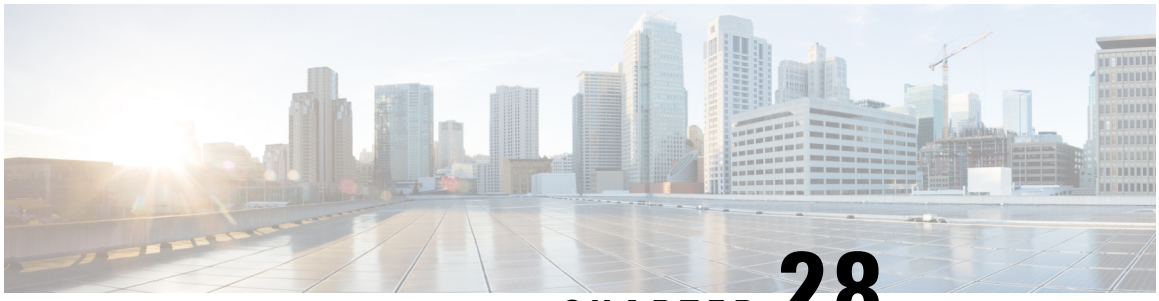
Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Specifies an interface type and identifier; enters the interface configuration mode.

Step 4	**ipv6 nd raguard** [**attach-policy** *policy_name*]
	Example:
	`Router(config-if)# ipv6 nd raguard attach-policy example_policy`
	`Router(config-vlan-config)# ipv6 nd raguard attach-policy example_policy`
		Attaches the Neighbor Discovery Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.



CHAPTER 28

Configuring IP Device Tracking

This chapter provides details about configuring IP Device Tracking (IPDT) on the IR8340 Router.

- Information About IP Device Tracking, on page 355
- Overview of SISF-Based Device Tracking, on page 356
- Options to Enable SISF-Based Device Tracking, on page 356
- How to Configure SISF-Based Device Tracking, on page 357

Information About IP Device Tracking

The main IPDT task is to keep track of connected hosts (association of MAC and IP address). In order to do this, it sends unicast Address Resolution Protocol (ARP) probes with a default interval of 30 seconds; these probes are sent to the MAC address of the host connected on the other side of the link, and use Layer 2 (L2) as the default source the MAC address of the physical interface out of which the ARP goes and a sender IP address of 0.0.0.0, based on the ARP Probe definition listed in RFC 5227.

In this document, the term 'ARP Probe' is used to refer to an ARP Request packet, broadcast on the local link, with an all-zero 'sender IP address'. The 'sender hardware address' MUST contain the hardware address of the interface sending the packet. The 'sender IP address' field MUST be set to all zeroes, to avoid polluting ARP caches in other hosts on the same link in the case where the address turns out to be already in use by another host. The 'target IP address' field MUST be set to the address being probed. An ARP Probe conveys both a question ("Is anyone using this address?") and an implied statement ("This is the address I hope to use.").

The purpose of IPDT is for the switch to obtain and maintain a list of devices that are connected to the switch via an IP address. The probe does not populate the tracking entry; it is simply used in order to maintain the entry in the table after it is learned through an ARP request/reply from the host.

IP ARP Inspection is enabled automatically when IPDT is enabled; it detects the presence of new hosts when it monitors ARP packets. If dynamic ARP inspection is enabled, only the ARP packets that it validates are used in order to detect new hosts for the Device Tracking table.

IP DHCP Snooping, if enabled, detects the presence or removal of new hosts when DHCP assigns or revokes their IP addresses.

IPDT is a feature that has always been available. However, on more recent Cisco IOS releases, its interdependencies are enabled by default (see Cisco bug ID CSCuj04986). It can be extremely useful when its database of IP/MAC hosts associations is used in order to populate the source IP of dynamic Access Control Lists (ACLs), or to maintain a binding of an IP address to a security group tag.

The ARP probe is sent under two circumstances:

- The link associated with a current entry in the IPDT database moves from a DOWN to an UP state, and the ARP entry has been populated.
- A link already in the UP state that is associated with an entry in the IPDT database has an expired probe interval.

Overview of SISF-Based Device Tracking

The Switch Integrated Security Features based (SISF-based) device tracking feature is part of the suite of first-hop security features.

The main role of the feature is to track the presence, location, and movement of end-nodes in the network. SISF snoops traffic received by the switch, extracts device identity (MAC and IP address), and stores them in a binding table. Many features, such as, Cisco TrustSec, IEEE 802.1X, LISP, and web authentication depend on the accuracy of this information to operate properly.

SISF-based device tracking supports both IPv4 and IPv6.

Even with the introduction of SISF-based device tracking, the legacy device tracking CLI (IP Device Tracking (IPDT) and IPv6 Snooping CLI) continues to be available. When you bootup the switch, the set of commands that is available depends on existing configuration, and only one of the following is available:

- SISF-based device tracking CLI, or
- IPDT and IPv6 Snooping CLI

SISF-based device tracking can be enabled manually (by using **device-tracking** commands), or programmatically (which is the case when providing device tracking services to other features).

Options to Enable SISF-Based Device Tracking

SISF-based device tracking is disabled by default.

You can enable it by defining a device tracking policy and attaching the policy to a specific target.

Note The target could be an interface or a VLAN.

Manually Enabling SISF-Based Device Tracking

- Option 1: Apply the **default** device tracking policy to a target.

  Enter the **device-tracking** command in the interface configuration mode or in the VLAN configuration mode. The system then attaches the **default** policy it to the interface or VLAN.

Note The **default** policy is a built-in policy with default settings; you cannot change any of the attributes of the **default** policy. In order to be able to configure device tracking policy attributes you must create a custom policy. See *Option 2: Create a custom policy with custom settings*.

- Option 2: Create a custom policy with custom settings.

  Enter the device-tracking policy command in global configuration mode and enter a custom policy name. The system creates a policy with the name you specify. You can then configure the available settings, in the device tracking configuration mode (config-device-tracking), and attach the policy to a specified target.

Programmatically Enabling SISF-Based Device Tracking

Some features rely on device tracking and utilize the trusted database of binding entries that SISF-based device tracking builds and maintains. These features, also called device tracking clients, enable device tracking programmatically (create and attach the device tracking policy).

Note The exceptions here are IEEE 802.1X, web authentication, Cisco TrustSec, and IP Source Guard (IPSG) - they also rely on device tracking, but they do not enable it. For these device tracking clients, you must enter the **ip dhcp snooping vlan** vlan command, to programmatically enable device tracking on a particular target.

Note the following about programmatically enabling SISF-based device tracking:

- A device tracking client *requires* device tracking to be enabled.

  There are several device tracking clients, therefore, multiple programmatic policies could be created. The settings of each policy differ depending on the device tracking client that creates the policy.

- The policy that is created, and its settings, are system-defined.

  Configurable policy attributes are available in the device tracking configuration mode (config-device-tracking) and vary from one release to another. If you try to modify an attribute that is not configurable, the configuration change is rejected and an error message is displayed.

How to Configure SISF-Based Device Tracking

Manually Enabling SISF-Based Device Tracking

Applying the Default Device Tracking Policy to a Target

Beginning in privileged EXEC mode, follow these steps to apply the default device tracking policy to an interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.

Step 3	Specify an interface or a VLAN.
	• **interface** *type number*
	• **vlan configuration** *vlan_list*
	Example:
	`Device(config)# interface gigabitethernet 0/1/0`
	OR
	`Device(config)# vlan configuration 100`
		interface *type number*—Specifies the interface and enters interface configuration mode. The device tracking policy will be attached to the specified interface.
		vlan configuration *vlan_list*—Specifies the VLANs and enters VLAN feature configuration mode. The device tracking policy will be attached to the specified VLAN.

Step 4	**device-tracking**
	Example:
	`Device(config-if)# device-tracking`
	OR
	`Device(config-vlan-config)# device-tracking`
		Enables SISF-based device tracking and attaches the default policy it to the interface or VLAN.
		The default policy is a built-in policy with default settings; none of the attributes of the default policy can be changed.

Step 5	**end**
	Example:
	`Device(config-if)# end`
	OR
	`Device(config-vlan-config)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.
		Exits VLAN feature configuration mode and returns to privileged EXEC mode.

Step 6	**show device-tracking policy** *policy-name* **Example:** `Device# show device-tracking policy default`	Displays device-tracking policy configuration, and all the targets it is applied to.

Creating a Custom Device Tracking Policy with Custom Settings

Beginning in privileged EXEC mode, follow these steps to create and configure a device tracking policy:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	[**no**] **device-tracking policy** *policy_name* **Example:** `Device(config)# device-tracking policy example_policy`	Creates the policy and enters device-tracking configuration mode.

Step 4	[**data-glean** | **default** | **destination-glean** | **device-role** | **distribution-switch** | **exit** | **limit** | **no** | **prefix-glean** | **protocol** | **security-level** | **tracking** | **trusted-port** | **vpc**]
	Example:
	`Device(config-device-tracking)# security-level glean`
		Enter the question mark (?) at the system prompt to obtain a list of available options in this mode. You can configure the following for both IPv4 and IPv6:
		• (Optional) **data-glean** —Enables learning of addresses from a data packet snooped from a source inside the network and populates the binding table with the data traffic source address. Enter one of these options:
		  • log-only —Generates a syslog message upon data packet notification
		  • recovery —Uses a protocol to enable binding table recovery. Enter **NDP** or **DHCP** .
		• (Optional) **default** —Sets the policy attribute to its default value. You can set these policy attributes to their default values: **data-glean** , **destination-glean** , **device-role** , **limit** , **prefix-glean** , **protocol** , **security-level** , **tracking** , **trusted-port** .
		• (Optional) **destination-glean** —Populates the binding table by gleaning data traffic destination address. Enter one of these options:
		  • **log-only** —Generates a syslog message upon data packet notification
		  • **recovery** —Uses a protocol to enable binding table recovery. Enter **DHCP** .

	Command or Action	Purpose

		• (Optional) **device-role** —Sets the role of the device attached to the port. It can be a node or a switch. Enter one of these options:
		  • node —Configures the attached device as a node. This is the default option.
		  • switch —Configures the attached device as a switch.
		• (Optional) **distribution-switch** —Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.
		• **exit** —Exits the device-tracking policy configuration mode.
		• **limit address-count** —Specifies an address count limit per port. The range is 1 to 32000.
		• **no** —Negates the command or sets it to defaults.
		• (Optional) **prefix-glean** —Enables learning of prefixes from either IPv6 Router Advertisements or from DHCP-PD. You have the following option:
		  • (Optional) **only** —Gleans only prefixes and not host addresses.
		• (Optional) **protocol** —Sets the protocol to glean; by default, all are gleaned. Enter one of these options:
		  • **arp** [**prefix-list** *name*] —Gleans addresses in ARP packets. Optionally, enter the name of prefix-list that is to be matched.
		  • **dhcp4** [**prefix-list** *name*] —Glean addresses in DHCPv4 packets. Optionally, enter the name of prefix-list that is to be matched.
		  • **dhcp6** [**prefix-list** *name*] —Glean addresses in DHCPv6 packets. Optionally, enter the name of prefix-list that is to be matched.

	Command or Action	Purpose
		• **ndp** [**prefix-list** *name*] —Glean addresses in NDP packets. Optionally, enter the name of prefix-list that is to be matched. • **udp** [**prefix-list** *name*] —Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect. • (Optional) **security-level** —Specifies the level of security enforced by the feature. Enter one of these options: • **glean** —Gleans addresses passively. • **guard** —Inspects and drops un-authorized messages. This is the default. • **inspect** —Gleans and validates messages. • (Optional) **tracking** —Specfies a tracking option. Enter one of these options: • **disable** [**stale-lifetime** [1-86400-seconds| **infinite**]] —Turns of device-tracking. Optionally, you can enter the duration for which the entry is kept inactive before deletion, or keep it permanently inactive. • **enable** [**reachable-lifetime** [1-86400-seconds| **infinite**]] —Turns on device-tracking. Optionally, you can enter the duration for which the entry is kept reachable, or keep it permanently reachable. • (Optional) **trusted-port** —Sets up a trusted port. Disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.

	Command or Action	Purpose
		• (Optional) **vpc** —Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.
Step 5	**end** **Example:** `Device(config-device-tracking)# end`	Exits device-tracking configuration mode and returns to privileged EXEC mode.
Step 6	**show device-tracking policy** *policy-name* **Example:** `Device# show device-tracking policy default`	Displays the device-tracking policy configuration.

Attaching a Device Tracking Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach a device tracking policy to an interface:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Specifies an interface and enters interface configuration mode.
Step 4	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-if)# device-tracking attach-policy example_policy`	Attaches the device tracking policy to the interface. Device tracking is also supported on EtherChannels. **Note** SISF based device-tracking poli created policies can be removed removed.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Exits device-tracking configuration mode and returns to privileged EXEC mode.
Step 6	**show device-tracking policies** [**interface***interface-id*] **Example:** `Device# show device-tracking policies interface gigabitethernet 0/1/0`	Displays device-tracking policy configuration, and all the targets it is applied to.

Attaching a Device Tracking Policy to a VLAN

Beginning in privileged EXEC mode, follow these steps to attach a device-tracking policy to VLANs across multiple interfaces:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**vlan configuration** *vlan_list* **Example:** `Device(config)# vlan configuration 100`	Specifies an interface and enters interface configuration mode.
Step 4	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-vlan-config)# device-tracking attach-policy example_policy`	Attaches the device tracking policy to the specified VLANs across all switch interfaces. **Note** SISF based device-tracking p created policies can be remo removed.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits device-tracking configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show device-tracking policies** [**vlan***vlan-id*] **Example:** `Device# show device-tracking policies vlan 100`	Verifies that the policy is attached to the specified VLAN, without exiting the VLAN interface configuration mode.

Configuring a Multi-Switch Network to Stop Creating Binding Entries from a Trunk Port

In a multi-switch network, SISF-based device tracking provides the capability to distribute binding table entries between switches running the feature. Binding entries are only created on the switches where the host appears on an access port. No entry is created for a host that appears over a trunk port. This is achieved by configuring a policy with the **trusted-port** and **device-role switch** options, and attaching it to the trunk port.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**device-tracking policy** *policy_name* **Example:** `Device(config)# device-tracking policy DT_trunk_policy`	Enters device-tracking policy configuration mode, for the specified policy.
Step 4	**device-role switch** **Example:** `Device(config-device-tracking)# device-role switch`	Specifies the role of the device attached to the port. Default is node. Enter the **device-role switch** option to stop the creation of binding entries for the port.
Step 5	**trusted-ports** **Example:** `Device(config-device-tracking)# trusted-port`	Sets up a trusted port. Disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.
Step 6	**end** **Example:** `Device(config-device-tracking)# end`	Exits device-tracking policy configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Specifies a trunk interface and enters interface configuration mode.
Step 8	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-if)# device-tracking attach-policy DT_trunk_policy`	Attaches a device tracking policy to the interface or the specified VLANs on the interface.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling SISF Syslogs

To enable syslogs of binding table events (such as create, delete, or modify entries), the following commands need to be executed:

```
device-tracking binding logging
```

If appropriate syslog level (6 - informational) need to be adjusted, execute:

```
logging console informational
```

To direct it to buffer:

```
logging buffered informational
```

to generate syslogs for MAC and/or IP theft events:

```
device-tracking logging theft
```

To generate syslogs for events when any of the SISF features decides to drop the packet for any reason:

```
device-tracking logging packet drop
```

to generate syslogs for events related to destination guard events:

```
device-tracking logging resolution-veto
```

the following command could be used to enable syslogs for all three event types listed above (but not to binding table events):

```
device-tracking logging
```

Example: DHCP Snooping Auto Enabling DT PROGRAMMATIC Policy

```
configure terminal
device-tracking policy Poo@12345
security-level glean
 device-role node
 limit address-count 10
tracking enable
  end
Switch(confg)#ip dhcp snooping
```

```
Switch(confg)#ip dhcp snooping vlan 100
Switch(confg)#end

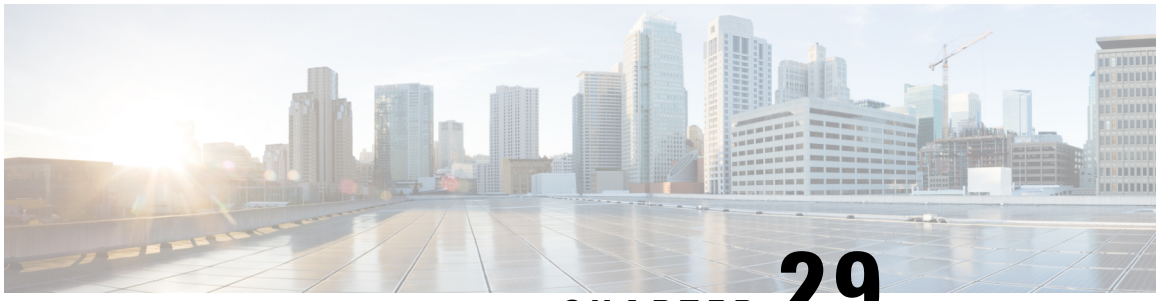
configure terminal
interface Gi0/1/0
device-tracking attach-policy Poo@12345
end
```

Use the following show commands to display the status of device tracking:

```
router#show device-tracking policies
Target               Type  Policy               Feature        Target range
Gi0/1/0              PORT  Poo@12345            Device-tracking vlan all
vlan 100             VLAN  DT-PROGRAMMATIC      Device-tracking vlan all
router#
router#show device-tracking policy DT-PROGRAMMATIC
Device-tracking policy DT-PROGRAMMATIC configuration:
security-level glean
device-role node
gleaning from Neighbor Discovery
gleaning from DHCP6
gleaning from ARP
 gleaning from DHCP4
 NOT gleaning from protocol unkn
limit address-count for IPv4 per mac 1
 tracking (downlink only) enable
Policy DT-PROGRAMMATIC is applied on the following targets:
Target               Type  Policy               Feature        Target range
vlan 100             VLAN  DT-PROGRAMMATIC      Device-tracking vlan all
router#
router#show device-tracking policy Poo@12345
Device-tracking policy Poo@12345 configuration:
 security-level glean
 device-role node
 gleaning from Neighbor Discovery
gleaning from DHCP6
  gleaning from ARP
 gleaning from DHCP4
  gleaning from protocol unkn
 limit address-count 10
 tracking enable
Policy Poo@12345 is applied on the following targets:
Target               Type  Policy               Feature        Target range
Gi0/1/0              PORT  Poo@12345            Device-tracking vlan all

Router#show device-tracking database
Binding Table has 11 entries, 11 dynamic (limit 100000)
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


 Network Layer Address                  Link Layer Address     Interface  vlan       prlvl
      age        state      Time left
ARP 100.1.1.1                               ac4a.6763.5a51         Gi0/1/10   100
0005       169s       REACHABLE  143 s try 0
DH4 100.0.0.14                              0013.0100.0004         Gi0/1/0    100
0024       53s        REACHABLE  259 s(31535947 s)
DH4 100.0.0.13                              0013.0100.0003         Gi0/1/0    100
0024       53s        REACHABLE  262 s(31535946 s)
DH4 100.0.0.12                              0013.0100.0002         Gi0/1/0    100
0024       52s        REACHABLE  250 s(31535947 s)
```



CHAPTER 29

Configuring Security for VPNs with IPsec

This chapter describes how to configure basic IPsec VPNs. IPsec is a framework of open standards developed by the IETF. It provides security for the transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (“peers”), such as Cisco routers.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

- Information About Configuring Security for VPNs with IPsec, on page 367
- How to Configure IPsec VPNs, on page 370

Information About Configuring Security for VPNs with IPsec

Supported Standards

Cisco implements the following standards with this feature:

- IPsec—IPsec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer; IPsec uses IKE to handle negotiation of protocols and algorithms based on the local policy, and generate the encryption and authentication keys to be used by IPsec. IPsec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Note The term IPsec is sometimes used to describe the entire protocol of IPsec data services and IKE security protocols, and is also sometimes used to describe only the data services.

- IKE (IKEv1 and IKEv2)—A hybrid protocol that implements Oakley and SKEME key exchanges inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. While IKE is

used with other protocols, its initial implementation is with the IPsec protocol. IKE provides authentication of IPsec peers, negotiates IPsec security associations, and establishes IPsec keys.

The component technologies implemented for IPsec include:

Note Cisco no longer recommends using DES, 3DES, MD5 (including HMAC variant), and Diffie-Hellman (DH) groups 1, 2 and 5; instead, you should use AES, SHA and DH Groups 14 or higher. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

- AES—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is a privacy transform for IPsec and IKE and has been developed to replace DES. AES is designed to be more secure than DES. AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.
- DES—Data Encryption Standard. An algorithm that is used to encrypt packet data. Cisco software implements the mandatory 56-bit DES-CBC with Explicit IV. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPsec packet. For backwards compatibility, Cisco IOS IPsec also implements the RFC 1829 version of ESP DES-CBC.

Cisco IOS also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Cisco no longer recommends Triple DES (3DES).

Note Cisco IOS images with strong encryption (including, but not limited to 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send an e-mail to export@cisco.com.

- SHA-2 and SHA-1 family (HMAC variant)—Secure Hash Algorithm (SHA) 1 and 2. Both SHA-1 and SHA-2 are hash algorithms used to authenticate packet data and verify the integrity verification mechanisms for the IKE protocol. HMAC is a variant that provides an additional level of hashing. SHA-2 family adds the SHA-256 bit hash algorithm and SHA-384 bit hash algorithm. This functionality is part of the Suite-B requirements that comprises four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support. SHA-2 for ISAKMP is supported in Cisco IOS XE 15.3(3)S and later.
- Diffie-Hellman—A public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. It supports 768-bit (the default), 1024-bit, 1536-bit, 2048-bit, 3072-bit, and 4096-bit DH groups. It also supports a 2048-bit DH group with a 256-bit subgroup, and 256-bit and 384-bit elliptic curve DH (ECDH). Cisco recommends using 2048-bit or larger DH key exchange, or ECDH key exchange.
- MD5 (Hash-based Message Authentication Code (HMAC) variant)—Message digest algorithm 5 (MD5) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.

IPsec as implemented in Cisco software supports the following additional standards:

- AH—Authentication Header. A security protocol, which provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram).
- ESP—Encapsulating Security Payload. A security protocol, which provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected.

Supported Encapsulation

IPsec works with the following serial encapsulations: Frame Relay, High-Level Data-Links Control (HDLC), and PPP.

IPsec also works with Generic Routing Encapsulation (GRE) and IPinIP Layer 3, Data Link Switching+ (DLSw+), and Source Route Bridging (SRB) tunneling protocols; however, multipoint tunnels are not supported. Other Layer 3 tunneling protocols may not be supported for use with IPsec.

IPsec Functionality Overview

IPsec provides the following network security services. (In general, the local security policy dictates the use of one or more of these services.)

- Data confidentiality—The IPsec sender can encrypt packets before transmitting them across a network.
- Data integrity—The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.
- Data origin authentication—The IPsec receiver can authenticate the source of the sent IPsec packets. This service is dependent upon the data integrity service.
- Anti-replay—The IPsec receiver can detect and reject replayed packets.

IPsec provides secure *tunnels* between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets by specifying the characteristics of these tunnels. When the IPsec peer recognizes a sensitive packet, the peer sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer. (The use of the term *tunnel* in this chapter does not refer to using IPsec in tunnel mode.)

More accurately, these *tunnels* are sets of security associations (SAs) that are established between two IPsec peers. The SAs define the protocols and algorithms to be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (AH or ESP).

Once established, the set of SAs (outbound to the peer) is then applied to the triggering packet and to subsequent applicable packets as those packets exit the router. “Applicable” packets are packets that match the same access list criteria that the original packet matched. For example, all applicable packets could be encrypted before being forwarded to the remote peer. The corresponding inbound SAs are used when processing the incoming traffic from that peer.

Multiple IPsec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams only need to be authenticated, while other data streams must both be encrypted and authenticated.

IKEv1 Transform Sets

An Internet Key Exchange version 1 (IKEv1) transform set represents a certain combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

IKEv2 Transform Sets

An Internet Key Exchange version 2 (IKEv2) proposal is a set of transforms used in the negotiation of IKEv2 SA as part of the IKE_SA_INIT exchange. An IKEv2 proposal is regarded as complete only when it has at least an encryption algorithm, an integrity algorithm, and a Diffie-Hellman (DH) group configured. If no proposal is configured and attached to an IKEv2 policy, then the default proposal is used in the negotiation. The default proposal is a collection of commonly used algorithms which are as follows:

```
encryption aes-cbc-128 3des
integrity sha1 md5
group 5 2
```

Although the **crypto ikev2 proposal** command is similar to the **crypto isakmp policy priority** command, the IKEv2 proposal differs as follows:

- An IKEv2 proposal allows configuration of one or more transforms for each transform type.
- An IKEv2 proposal does not have any associated priority.

Note To use IKEv2 proposals in negotiation, they must be attached to IKEv2 policies. If a proposal is not configured, then the default IKEv2 proposal is used with the default IKEv2 policy.

How to Configure IPsec VPNs

Creating Crypto Access Lists

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination*	Specifies conditions to determine which IP packets are protected.

	Command or Action	Purpose

	destination-wildcard [**log**] or **ip access-list extended** *name*
	Example:
	`Router(config)# access-list 100 permit ip 10.0.68.0 0.0.0.255 10.1.1.0 0.0.0.255`
	`Router(config)# ip access-list extended vpn-tunnel`
		• You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered extended access list; the **ip access-list extended** command designates a named access list.
		• Enable or disable crypto for traffic that matches these conditions.
		Tip Cisco recommends that you using the **any** keyword.

Step 4	Repeat Step 3 for each crypto access list you want to create.	

What to do next

After at least one crypto access list is created, a transform set needs to be defined as described in Configuring Transform Sets for IKEv1 and IKEv2 Proposals, on page 371.

Next the crypto access lists need to be associated to particular interfaces when you configure and apply crypto map sets to the interfaces. (Follow the instructions in Creating Crypto Map Sets, on page 375 and Applying Crypto Map Sets to Interfaces, on page 384).

Configuring Transform Sets for IKEv1 and IKEv2 Proposals

Perform this task to define a transform set that is to be used by the IPsec peers during IPsec security association negotiations with IKEv1 and IKEv2 proposals.

Restrictions

If you are specifying SEAL encryption, note the following restrictions:

- Your router and the other peer must not have a hardware IPsec encryption.
- Your router and the other peer must support IPsec.
- Your router and the other peer must support the k9 subsystem.
- SEAL encryption is available only on Cisco equipment. Therefore, interoperability is not possible.
- Unlike IKEv1, the authentication method and SA lifetime are not negotiable in IKEv2, and because of this, these parameters cannot be configured under the IKEv2 proposal.

Configuring Transform Sets for IKEv1

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]] **Example:** `Router(config)# crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac`	Defines a transform set and enters crypto transform configuration mode. • There are complex rules defining the entries that you can use for transform arguments. These rules are explained in the command description for the **crypto ipsec transform-set** command, and the table in "About Transform Sets" section provides a list of allowed transform combinations.
Step 4	**mode** [**tunnel** | **transport**] **Example:** `Router(cfg-crypto-tran)# mode transport`	(Optional) Changes the mode associated with the transform set. • The mode setting is applicable only to traffic whose source and destination addresses are the IPsec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)
Step 5	**end** **Example:** `Router(cfg-crypto-tran)# end`	Exits crypto transform configuration mode and enters privileged EXEC mode.
Step 6	**clear crypto sa** [**peer** {*ip-address* | *peer-name*} | **sa map** *map-name* | **sa entry** *destination-address protocol spi*] **Example:** `Router# clear crypto sa`	(Optional) Clears existing IPsec security associations so that any changes to a transform set takes effect on subsequently established security associations. Manually established SAs are reestablished immediately. • Using the **clear crypto sa** command without parameters clears out the full SA database, which clears out active security sessions.

	Command or Action	Purpose
		• You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database.
Step 7	**show crypto ipsec transform-set** [**tag** *transform-set-name*] **Example:** `Router# show crypto ipsec transform-set`	(Optional) Displays the configured transform sets.

What to do next

After you have defined a transform set, you should create a crypto map as specified in Creating Crypto Map Sets, on page 375.

Configuring Transform Sets for IKEv2

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 proposal** *proposal-name* **Example:** `Router(config)# crypto ikev2 proposal proposal-1`	Specifies the name of the proposal and enters crypto IKEv2 proposal configuration mode. • The proposals are referred in IKEv2 policies through the proposal name.
Step 4	**encryption** *transform1* [*transform2*] ... **Example:** `Router(config-ikev2-proposal)# encryption aes-cbc-128`	(Optional) Specifies one or more transforms of the following encryption type: • AES-CBC 128—128-bit AES-CBC • AES-CBC 192—192-bit AES-CBC • AES-CBC 256—256-bit AES-CBC • 3DES—168-bit DES (No longer recommended. AES is the recommended encryption algorithm).

	Command or Action	Purpose
Step 5	**integrity** *transform1* [*transform2*] ... **Example:** `Router(config-ikev2-proposal)# integrity sha1`	(Optional) Specifies one or more transforms of the following integrity type: • The **sha256** keyword specifies SHA-2 family 256-bit (HMAC variant) as the hash algorithm. • The **sha384** keyword specifies SHA-2 family 384-bit (HMAC variant) as the hash algorithm. • The **sha512** keyword specifies SHA-2 family 512-bit (HMAC variant) as the hash algorithm • The **sha1** keyword specifies the SHA-1 (HMAC variant) as the hash algorithm. • The **md5** keyword specifies MD5 (HMAC variant) as the hash algorithm. (No longer recommended. SHA-1 is the recommended replacement.)
Step 6	**group** *transform1* [*transform2*] ... **Example:** `Router(config-ikev2-proposal)# group 14`	(Optional) Specifies one or more transforms of the possible DH group type: • **1**—768-bit DH (No longer recommended.) • **2**—1024-bit DH (No longer recommended) • **5**—1536-bit DH (No longer recommended) • **14**—Specifies the 2048-bit DH group. • **15**—Specifies the 3072-bit DH group. • **16**—Specifies the 4096-bit DH group. • **19**—Specifies the 256-bit elliptic curve DH (ECDH) group. • **20**—Specifies the 384-bit ECDH group. • **24**—Specifies the 2048-bit DH/DSA group.
Step 7	**end** **Example:** `Router(config-ikev2-proposal)# end`	Exits crypto IKEv2 proposal configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show crypto ikev2 proposal** **Example:** `Router# show crypto ikev2 proposal`	(Optional) Displays the parameters for each IKEv2 proposal.

Creating Crypto Map Sets

Creating Static Crypto Maps

When IKE is used to establish SAs, the IPsec peers can negotiate the settings they use for the new security associations. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

Perform this task to create crypto map entries that use IKE to establish SAs. To create IPv6 crypto map entries, you must use the **ipv6** keyword with the **crypto map** command. For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** [**ipv6**] *map-name seq-num* [**ipsec-isakmp**] **Example:** `Router(config)# crypto map static-map 1 ipsec-isakmp`	Creates or modifies a crypto map entry, and enters crypto map configuration mode. • For IPv4 crypto maps, use the command without the **ipv6** keyword.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address vpn-tunnel`	Names an extended access list. • This access list determines the traffic that should be protected by IPsec and the traffic that should not be protected by

	Command or Action	Purpose
		IPsec security in the context of this crypto map entry.
Step 5	**set-peer** {*hostname* | *ip-address*} **Example:** `Router(config-crypto-m)# set-peer 192.168.101.1`	Specifies a remote IPsec peer—the peer to which IPsec protected traffic can be forwarded. • Repeat for multiple remote peers.
Step 6	**crypto ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*} **Example:** `Router(config-crypto-m)# set security-association dummy seconds 5`	Enables generating dummy packets. These dummy packets are generated for all flows created in the crypto map.
Step 7	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set aesset`	Specifies the transform sets that are allowed for this crypto map entry. • List multiple transform sets in the order of priority (highest priority first).
Step 8	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**} **Example:** `Router(config-crypto-m)# set security-association lifetime seconds 2700`	(Optional) Specifies a SA lifetime for the crypto map entry. • By default, the SAs of the crypto map are negotiated according to the global lifetimes, which can be disabled.
Step 9	**set security-association level per-host** **Example:** `Router(config-crypto-m)# set security-association level per-host`	(Optional) Specifies that separate SAs should be established for each source and destination host pair. • By default, a single IPsec "tunnel" can carry traffic for multiple source hosts and multiple destination hosts. ⚠ **Caution** Use this command with care bec
Step 10	**set pfs [group1 | group14 | group15 | group16 | group19 | group2 | group20 | group24 | group5]** **Example:** `Router(config-crypto-m)# set pfs group14`	(Optional) Specifies that IPsec either should ask for password forward secrecy (PFS) when requesting new SAs for this crypto map entry or should demand PFS in requests received from the IPsec peer. • Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended).

	Command or Action	Purpose
		• Group 2 specifies the 1024-bit DH identifier. (No longer recommended). • Group 5 specifies the 1536-bit DH identifier. (No longer recommended) • Group 14 specifies the 2048-bit DH identifier. • Group 15 specifies the 3072-bit DH identifier. • Group 16 specifies the 4096-bit DH identifier. • Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier. • Group 20 specifies the 384-bit ECDH identifier. • Group 24 specifies the 2048-bit DH/DSA identifier • By default, PFS is not requested. If no group is specified with this command, group 1 is used as the default.
Step 11	**end** **Example:** `Router(config-crypto-m)# end`	Exits crypto map configuration mode and returns to privileged EXEC mode.
Step 12	**show crypto map** [**interface** *interface*| **tag** *map-name*] **Example:** `Router# show crypto map`	Displays your crypto map configuration.

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPsec traffic, clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the full SA database should be reserved for large-scale changes, or when the router is processing very little other IPsec traffic.

To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears out the full SA database, which clears active security sessions.)

What to do next

After you have successfully created a static crypto map, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see Applying Crypto Map Sets to Interfaces, on page 384.

Creating Dynamic Crypto Maps

Dynamic crypto map entries specify crypto access lists that limit traffic for which IPsec SAs can be established. A dynamic crypto map entry that does not specify an access list is ignored during traffic filtering. A dynamic crypto map entry with an empty access list causes traffic to be dropped. If there is only one dynamic crypto map entry in the crypto map set, it must specify the acceptable transform sets.

Perform this task to create dynamic crypto map entries that use IKE to establish the SAs.

Note IPv6 addresses are not supported on dynamic crypto maps.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Router(config)# crypto dynamic-map test-map 1`	Creates a dynamic crypto map entry and enters crypto map configuration mode.
Step 4	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set aesset`	Specifies the transform sets allowed for the crypto map entry. • List multiple transform sets in the order of priority (highest priority first). This is the only configuration statement required in dynamic crypto map entries.

	Command or Action	Purpose
Step 5	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 101`	(Optional) Specifies the list number or name of an extended access list. • This access list determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec security in the context of this crypto map entry. **Note** Although access lists are c recommended. • If an access list is configured, the data flow identity proposed by the IPsec peer must fall within a **permit** statement for this crypto access list. • If an access list is not configured, the device accepts any data flow identity proposed by the IPsec peer. However, if an access list is configured but the specified access list does not exist or is empty, the device drops all packets. This is similar to static crypto maps, which require access lists to be specified. • Care must be taken if the **any** keyword is used in the access list, because the access list is used for packet filtering as well as for negotiation. • You must configure a match address; otherwise, the behavior is not secure, and you cannot enable TED because packets are sent in the clear (unencrypted.)
Step 6	**set-peer** {*hostname* | *ip-address*} **Example:** `Router(config-crypto-m)# set-peer 192.168.101.1`	(Optional) Specifies a remote IPsec peer. Repeat this step for multiple remote peers. **Note** This is rarely configured in unknown remote peers.
Step 7	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**} **Example:**	(Optional) Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IP Security SAs.

	Command or Action	Purpose
	`Router(config-crypto-m)# set security-association lifetime seconds 720`	**Note** To minimize the possibility of p the rekey request triggered by a
Step 8	**set pfs [group1 | group14 | group15 | group16 | group19 | group2 | group20 | group24 | group5]** **Example:** `Router(config-crypto-m)# set pfs group14`	(Optional) Specifies that IPsec should ask for PFS when requesting new security associations for this crypto map entry or should demand PFS in requests received from the IPsec peer. • Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended). • Group 2 specifies the 1024-bit DH identifier. (No longer recommended). • Group 5 specifies the 1536-bit DH identifier. (No longer recommended) • Group 14 specifies the 2048-bit DH identifier. • Group 15 specifies the 3072-bit DH identifier. • Group 16 specifies the 4096-bit DH identifier. • Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier. • Group 20 specifies the 384-bit ECDH identifier. • Group 24 specifies the 2048-bit DH/DSA identifier • By default, PFS is not requested. If no group is specified with this command, **group1** is used as the default.
Step 9	**exit** **Example:** `Router(config-crypto-m)# exit`	Exits crypto map configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 11	**show crypto dynamic-map** [**tag** *map-name*] **Example:** `Router# show crypto dynamic-map`	(Optional) Displays information about dynamic crypto maps.
Step 12	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 13	**crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name* [**discover**] **Example:** `Router(config)# crypto map static-map 1 ipsec-isakmp dynamic test-map discover`	(Optional) Adds a dynamic crypto map to a crypto map set. • You should set the crypto map entries referencing dynamic maps to the lowest priority entries in a crypto map set. **Note** You must enter the **discover**
Step 14	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPsec traffic, clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the entire SA database must be reserved for large-scale changes, or when the router is processing minimal IPsec traffic.

To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears the full SA database, which clears active security sessions.)

What to do next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see Applying Crypto Map Sets to Interfaces, on page 384.

Creating Crypto Map Entries to Establish Manual SAs

Perform this task to create crypto map entries to establish manual SAs (that is, when IKE is not used to establish the SAs). To create IPv6 crypto maps entries, you must use the **ipv6** keyword with the **crypto map** command. For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** [**ipv6**] *map-name seq-num* [**ipsec-manual**] **Example:** `Router(config)# crypto map mymap 10 ipsec-manual`	Specifies the crypto map entry to be created or modified and enters crypto map configuration mode. • For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 102`	Names an IPsec access list that determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec in the context of this crypto map entry. • The access list can specify only one **permit** entry when IKE is not used.
Step 5	**set peer** {*hostname* | *ip-address*} **Example:** `Router(config-crypto-m)# set peer 10.0.0.5`	Specifies the remote IPsec peer. This is the peer to which IPsec protected traffic should be forwarded. • Only one peer can be specified when IKE is not used.
Step 6	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-m)# set transform-set someset`	Specifies which transform set should be used. • This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. **Note** Only one transform set can be s
Step 7	Do one of the following: **set session-key inbound ah** *spi hex-key-string* or **set session-key outbound ah** *spi hex-key-string* **Example:**	Sets the AH security parameter indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the AH protocol.

	Command or Action	Purpose
	`Router(config-crypto-m)# set session-key inbound ah 256 9876543210987654987654321098765 4` `Router(config-crypto-m)# set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc`	• This manually specifies the AH security association to be used with protected traffic.

Step 8	Do one of the following: **set session-key inbound ah esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*] or **set session-key outbound ah esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]
	Example:
	`Router(config-crypto-m)# set session-key inbound esp 256 cipher 0123456789012345`
	`Router(config-crypto-m)# set session-key outbound esp 256 cipher abcdefabcdefabcd`
		Sets the Encapsulating Security Payload (ESP) Security Parameter Indexes (SPI) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the ESP protocol.
		Or
		Specifies the cipher keys if the transform set includes an ESP cipher algorithm. Specifies the authenticator keys if the transform set includes an ESP authenticator algorithm.
		• This manually specifies the ESP security association to be used with protected traffic.

Step 9	**exit** **Example:** `Router(config-crypto-m)# exit`	Exits crypto map configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto map** [**interface** *interface*| **tag** *map-name*] **Example:** `Router# show crypto map`	Displays your crypto map configuration.

For manually established SAs, you must clear and reinitialize the SAs for the changes to take effect. To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears the entire SA database, which clears active security sessions.)

What to do next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see Applying Crypto Map Sets to Interfaces, on page 384.

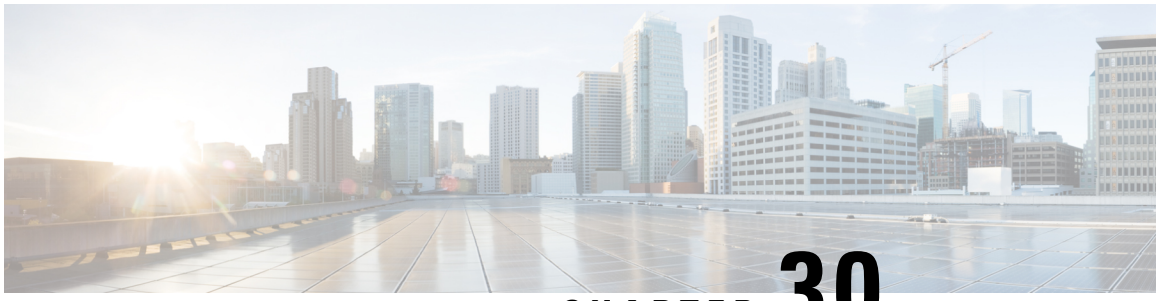
Applying Crypto Map Sets to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic flows. Applying the crypto map set to an interface instructs the device to evaluate the interface's traffic against the crypto map set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by the crypto map.

Perform this task to apply a crypto map to an interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type/number* **Example:** `Router# (config)# interface Gi 0/0/1`	Configures an interface and enters interface configuration mode.
Step 4	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map mymap`	Applies a crypto map set to an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**crypto map** *map-name* **local-address** *interface-id* **Example:** `Router(config)# crypto map mymap local-address loopback0`	(Optional) Permits redundant interfaces to share the same crypto map using the same local identity.
Step 7	**exit** **Example:** `Router(config)# exit`	(Optional) Exits global configuration mode.
Step 8	**show crypto map** [**interface** *interface*] **Example:** `Router# show crypto map`	Displays your crypto map configuration.



CHAPTER 30

Configuring High-availability Seamless Redundancy (HSR)

This chapter provides details about configuring High-availability Seamless Redundancy (HSR) on the Cisco IR8340 Router.

- Information About HSR, on page 385
- Configuring an HSR Ring, on page 388
- Clearing All Node Table and VDAN Table Dynamic Entries , on page 389
- Verifying Configuration, on page 389

Information About HSR

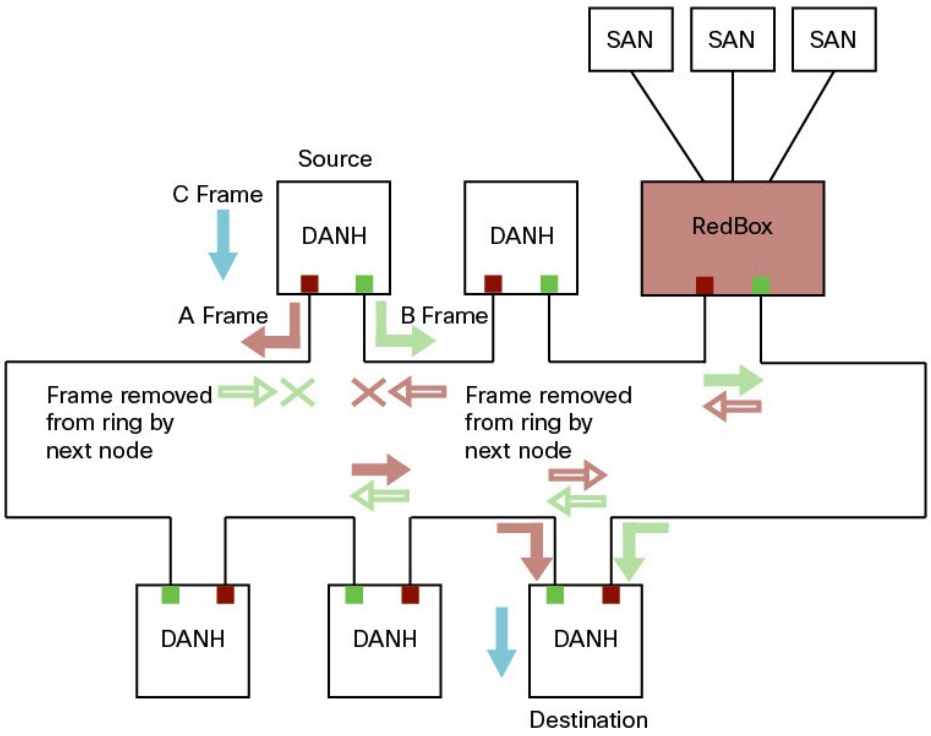
High-availability Seamless Redundancy (HSR) is defined in International Standard IEC 62439-3-2016 clause 5. HSR is similar to Parallel Redundancy Protocol (PRP) but is designed to work in a ring topology. Instead of two parallel independent networks of any topology (LAN-A and LAN-B), HSR defines a ring with traffic in opposite directions. Port-A sends traffic counter clockwise in the ring, and Port-B sends traffic clockwise in the ring.

The HSR packet format is also different from PRP. To allow the router to determine and discard duplicate packets, additional protocol specific information is sent with the data frame. For PRP, this is sent as part of a trailer called the redundancy control trailer (RCT), whereas for HSR this is sent as part of the header called the HSR header. Both the RCT and HSR header contain a sequence number, which is the primary data used to determine if the received frame is the first instance or a duplicate instance.

The non-switching nodes with two interfaces attached to the HSR ring are referred to as Doubly Attached Nodes implementing HSR (DANHs). Similar to PRP, Singly Attached Nodes (SANs) are attached to the HSR ring through a device called a RedBox (Redundancy Box). The RedBox acts as a DANH for all traffic for which it is the source or the destination. The router implements RedBox functionality using Gigabit Ethernet port connections to the HSR ring.

The following figure shows an example of an HSR ring as described in IEC 62439-3. In this example, the RedBox is an IR8340.

Figure 7: Example of HSR Ring Carrying Unicast Traffic



Devices that do not support HSR out of the box (for example, laptops and printers) cannot be attached to the HSR ring directly because all HSR capable devices must be able to process the HSR header on packets received from the ring and add the HSR header to all packets sent into the ring. These nodes are attached to the HSR ring through a RedBox. As shown in the figure above, the RedBox has two ports on the DANH side. Non-HSR SAN devices are attached to the upstream switch ports. The RedBox generates the supervision frames on behalf of these devices so that they are seen as DANH devices on the ring. Because the RedBox emulates these as DANH, they are called Virtual Doubly Attached Nodes (VDAN).

Loop Avoidance

To avoid loops and use network bandwidth effectively, the RedBox does not transmit frames that are already transmitted in same direction. When a node injects a packet into the ring, the packet is handled as follows to avoid loops:

- Unicast packet with destination inside the ring: When the unicast packet reaches the destination node, the packet is consumed by the respective node and is not forwarded.
- Unicast packet with destination not inside the ring: Because this packet does not have a destination node in the ring, it is forwarded by every node in the ring until it reaches the originating node. Because every node has a record of the packet it sent, along with the direction in which it was sent, the originating node detects that packet has completed the loop and drops the packet.
- Multicast packet: A multicast packet is forwarded by each node because there can be more than one consumer of this packet. For this reason a multicast packet always reaches the originating node. However, every node will check whether it has already forwarded the received packet through its outgoing interface.

Once the packet reaches the originating node, the originating node determines that it already forwarded this packet and drops the packet instead of forwarding it again.

HSR RedBox Modes of Operation

An HSR RedBox can operate in one of the following modes that define how HSR handles packets in different scenarios:

- HSR-SAN—This is the most basic mode. In this mode, the RedBox connects SAN devices to an HSR Ring. No other PRP or HSR network is involved in this configuration. In this mode, the traffic on the upstream switch port does not have HSR/PRP tags, and the RedBox represents the SAN device as a VDAN in the ring.
- HSR-PRP—This configuration is used to bridge HSR and PRP networks. The RedBox extracts the data from the PRP frame and generates the HSR frame using this data, and it performs the reverse operation for packets in the opposite direction.

  More than one PRP network can be bridged to one HSR ring and vice versa.
- HSR-HSR—In this mode, two HSR rings are connected through a four-port device called a QuadBox. In this mode, two of the ports on the switch are associated with one HSR ring, and the other two ports are associated with the second HSR ring. The remaining ports on the switch are shut down.



Note Only HSR-SAN mode is supported on IR8340 in Cisco IOS XE Release 17.8.x.

HSR-SAN Mode

In HSR-SAN mode, the RedBox inserts the HSR tag on behalf of the host and forwards the ring traffic, except for frames sent by the node itself, duplicate frames, and frames for which the node is the unique destination. In this mode, packets are handled as follows:

- A source DANH sends a frame passed from its upper layers ("C" frame), prefixes it with an HSR tag to identify frame duplicates, and sends the frame over each port ("A" frame and "B" frame).
- A destination DANH receives two identical frames from each port within a certain interval. The destination DANH removes the HSR tag of the first frame before passing it to its upper layers and discards any duplicate.
- Each node in the HSR ring forwards frames received from one port to the other port of the HSR pair. A node will not forward frames received on one port to the other under the following conditions:
  - The received frame returns to the originating node in the ring.
  - The frame is a unicast frame with a destination MAC address of a node upstream of the receiving node.
  - The node had already sent the same frame in the same direction. This rule prevents a frame from spinning in the ring in an infinite loop.

Note Maximum of 512 entries of node table and 512 entries of VDAN table is supported.

HSR-SAN Interfaces

One HSR ring can be configured on the following specific port pairs:

- Gi0/1/4, Gi0/1/5
- Gi0/1/6, Gi0/1/7

Configuring an HSR Ring

Use the **hsr-ring** command to configure an HSR ring.

```
Router(config)#hsr-ring 1 ?
  entryForgetTime                    Time to clear an entry from duplicate discard table
  fpgamode-DualUplinkEnhancement     Set FPGA register for source mac filtering
  nodeForgetTime                     Time to clear node entry from node table
  nodeRebootInterval                 Time to remain silent after reboot
  pauseFrameTime                     HSR pause frame time
  proxyNodeTableForgetTime           Time to clear node entry from vdan table
  supervisionFrameLifeCheckInterval  HSR supervision frame life check interval
  supervisionFrameOption             HSR supervision frame option
  supervisionFrameRedboxMacaddress   HSR Supervision Frame Redbox mac address
  supervisionFrameTime               HSR Inter supervision frame time

Router(config)#hsr-ring 1 nodeforgetTime ?
  <0-65535>  Node Forget time value

Router(config)#hsr-ring 1 proxyNodeTableForgetTime ?
  <0-65535>  Proxy Node table Forget time value

Router(config)#hsr-ring 1 supervisionFrameOption ?
  mac-da       MAC DA last two bit
  vlan-cfi     Enable CFI for the VLAN tagged frame
  vlan-cos     COS value for the VLAN tag
  vlan-tagged  Enable VLAN tagging

Router(config)#hsr-ring 1 supervisionFrameLifeCheckInterval ?
  <0-65535>  life check interval value for supervision frames

Router(config)#hsr-ring 1 supervisionFrameTime ?
  <0-65535>  time value between supervision frames

Router(config)#hsr-ring 1 supervisionFrameRedboxMacaddress ?
  H.H.H  48 bit redbox mac address
```

Configuring Interface Sub-Mode

Use the following commands to configure the physical ports, which are part of the HSR ring. Once you configure a physical port to be part of an HSR ring, a new interface “HSR-ring” will be created.

```
Router(config)#interface GigabitEthernet 1/1
Router(config-if)#hsr-ring 1
```

Clearing All Node Table and VDAN Table Dynamic Entries

To clear all dynamic entries in the node table, enter

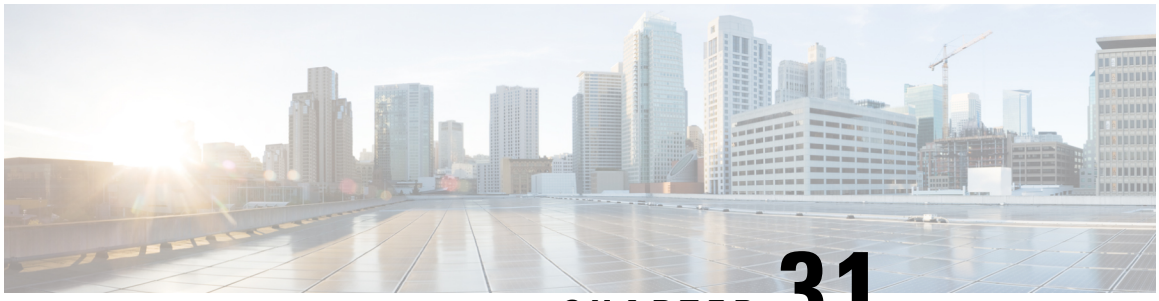
clear hsr node-table

To clear all dynamic entries in the VDAN table, enter

clear hsr vdan-table

Verifying Configuration

Command	Purpose
show hsr ring	Displays HSR ring information.
show hsr statistics	Displays statistics for HSR components. **Note** To clear HSR statistics information, enter the command **clear hsr statistics**.
show hsr node-table	Displays HSR node table.
show hsr vdan-table	Displays HSR Virtual Doubly Attached Node (VDAN) table.
show hsr supervisionFrameLifeCheckInterval	Displays supervision frame life check interval.
show hsr supervisionFrameOption	Displays supervision frame option.
show hsr supervisionFrameRedboxMacaddress	Displays supervision Redbox MAC Address.
show hsr supervisionFrameTime	Displays supervision frame time.



CHAPTER 31

Configuring Parallel Redundancy Protocol (PRP)

This chapter provides details about configuring Parallel Redundancy Protocol (PRP) on the Cisco IR8340 Router.

- Information About PRP, on page 391
- Creating a PRP Channel and Group, on page 392
- Clearing All Node Table and VDAN Table Dynamic Entries, on page 394
- Disabling the PRP Channel and Group, on page 394
- PRP Mode LED, on page 395
- Verifying Configuration, on page 395

Information About PRP

Parallel Redundancy Protocol (PRP) is defined in the International Standard IEC 62439-3. PRP is designed to provide hitless redundancy (zero recovery time after failures) in Ethernet networks.

To recover from network failures, redundancy can be provided by network elements connected in mesh or ring topologies using protocols like RSTP, REP, or MRP, where a network failure causes some reconfiguration in the network to allow traffic to flow again (typically by opening a blocked port). These schemes for redundancy can take between a few milliseconds to a few seconds for the network to recover and traffic to flow again.

PRP uses a different scheme, where the end nodes implement redundancy (instead of network elements) by connecting two network interfaces to two independent, disjointed, parallel networks (LAN-A and LAN-B). Each of these Dually Attached Nodes (DANs) then have redundant paths to all other DANs in the network.

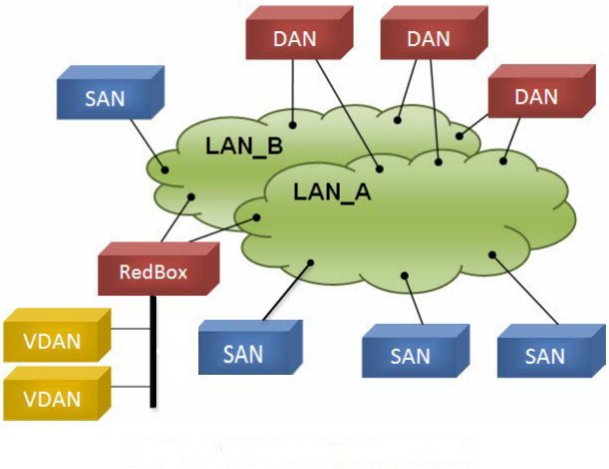
The DAN sends two packets simultaneously through its two network interfaces to the destination node. A redundancy control trailer (RCT), which includes a sequence number, is added to each frame to help the destination node distinguish between duplicate packets. When the destination DAN receives the first packet successfully, it removes the RCT and consumes the packet. If the second packet arrives successfully, it is discarded. If a failure occurs in one of the paths, traffic continues to flow over the other path uninterrupted, and zero recovery time is required.

Non-redundant endpoints in the network that attach only to either LAN-A or LAN-B are known as Singly Attached Nodes (SANs).

A Redundancy Box (RedBox) is used when an end node that does not have two network ports and does not implement PRP needs to implement redundancy. Such an end node can connect to a RedBox, which provides connectivity to the two different networks on behalf of the device. Because a node behind a RedBox appears

for other nodes like a DAN, it is called a Virtual DAN (VDAN). The RedBox itself is a DAN and acts as a proxy on behalf of its VDANs.

Figure 8: PRP Redundant Network



PRP Channels

PRP channel or channel group is a logical interface that aggregates two Gigabit Ethernet interfaces (access, trunk, or routed) into a single link. In the channel group, the lower numbered Gigabit Ethernet member port is the primary port and connects to LAN_A. The higher numbered port is the secondary port and connects to LAN_B. The PRP channel remains up as long as at least one of these member ports remains up and sends traffic. When both member ports are down, the channel is down. The total number of supported PRP channel groups is 2 per router, and the interfaces that can be utilized for each group are fixed.

- PRP channel group 1 always uses Gi0/1/4 for LAN_A and Gi0/1/5 for LAN_B
- PRP channel group 2 always uses Gi0/1/6 for LAN_A and Gi0/1/7 for LAN_B

Creating a PRP Channel and Group

To create and enable a PRP channel and group, follow these steps:

Procedure

Step 1 Enter global configuration mode:

configure terminal

Step 2 Assign two Gigabit Ethernet interfaces to the PRP channel group:

interface {**GigabitEthernet 0/1/4** | **GigabitEthernet 0/1/5**}

Use the **no interface prp-channel 1|2** command to disable PRP on the defined interfaces and shut down the interfaces.

Note You must apply the Gi 0/1/4 interface before the Gi 0/1/5 interface. So, we recommend using the **interface range** command. Similarly, you must apply the Gi 0/1/6 interface before the Gi 0/1/7 interface.

Step 3 (Optional) For Layer 2 traffic, enter **switchport**. (Default):

switchport

Step 4 (Optional) Set a non-trunking, non-tagged single VLAN Layer 2 (access) interface:

switchport mode access

Step 5 (Optional) Create a VLAN for the Gi 0/1/4 and Gi 0/1/5 interfaces:

switchport access vlan <*value*>

Note Only required for Layer 2 traffic.

Step 6 (Optional) Disable Precision Time Protocol (PTP) on the switch:

no ptp enable

PTP is enabled by default. You can disable it if you do not need to run PTP.

Step 7 Disable loop detection for the redundancy channel:

no keepalive

Step 8 Disable UDLD for the redundancy channel:

udld port disable

Step 9 Enter sub-interface mode and create a PRP channel group:

prp-channel-group *prp-channel group*

prp-channel group—Value of 1 or 2

The two interfaces that you assigned in step 2 are assigned to this channel group.

The **no** form of this command is not supported.

Step 10 Bring up the PRP channel:

no shutdown

Step 11 Specify the PRP interface and enter interface mode:

interface prp-channel *prp-channel-number*

prp-channel-number—Value of 1 or 2

Step 12 Configure bpdufilter on the prp-channel interface:

spanning-tree bpdufilter enable

Spanning-tree BPDU filter drops all ingress/egress BPDU traffic. This command is required to create independent spanning-tree domains (zones) in the network.

Step 13 (Optional) Configure LAN-A/B ports to quickly get to FORWARD mode:

spanning-tree portfast edge trunk

This command is optional but highly recommended. It improves the spanning-tree convergence time on PRP RedBoxes and LAN-A and LAN-B switch edge ports. It is also highly recommended to configure this command on the LAN_A/LAN_B ports directly connected to a RedBox PRP interface.

Clearing All Node Table and VDAN Table Dynamic Entries

To clear all dynamic entries in the node table, enter

clear prp node-table [**channel-group** *group*]

To clear all dynamic entries in the VDAN table, enter

clear prp vdan-table [**channel-group** *group*]

If you do not specify a channel group, the dynamic entries are cleared for all PRP channel groups.



Note The **clear prp node-table** and **clear prp vdan-table** commands clear only dynamic entries. To clear static entries, use the **no** form of the **nodeTableMacaddress** or **vdanTableMacaddress** commands.

Disabling the PRP Channel and Group

Procedure

Step 1 Enter global configuration mode:

configure terminal

Step 2 Disable the PRP channel:

no interface prp-channel *prp-channel-number*

prp-channel number— Value of 1 or 2

Step 3 Exit interface mode:

exit

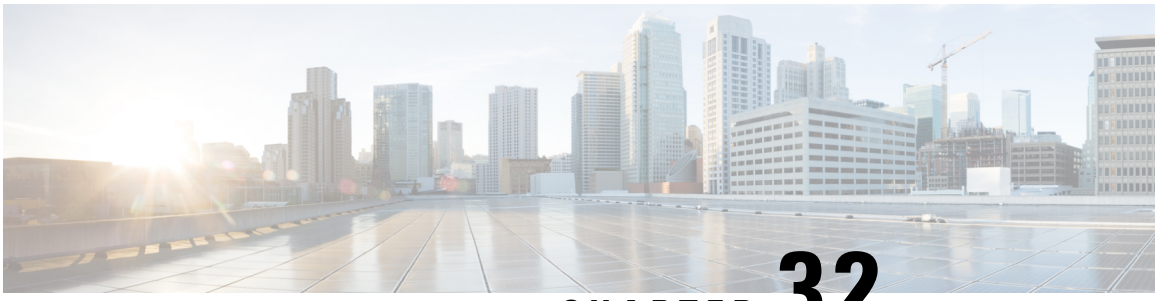
PRP Mode LED

The REDUN (Redundancy status) LED is on the faceplate. The router supports the following states.

Label Description	Color and State	Description
REDUN (Redundancy status)	Green (solid)	Redundancy protocols are configured and active.
	Amber (solid)	Redundancy fault detected.

Verifying Configuration

Command	Purpose
show prp control {**ptpLanOption** | **ptpProfile** | **supervisionFrameLifeCheckInterval** | **supervisionFrameOption** | **supervisionFrameRedboxMacaddress** | **supervisionFrameTime** | **nodeForgetTime** | **entryForgetTime** | **nodeRebootIntervalTime** | **pauseFrameTime**}	Displays PRP control information and supervision frame information.
show prp statistics {**egressPacketStatistics** | **ingressPacketStatistics** | **nodeTableStatistics** | **pauseFrameStatistics** | **ptpPacketStatistics**}	Displays statistics for PRP components.



CHAPTER 32

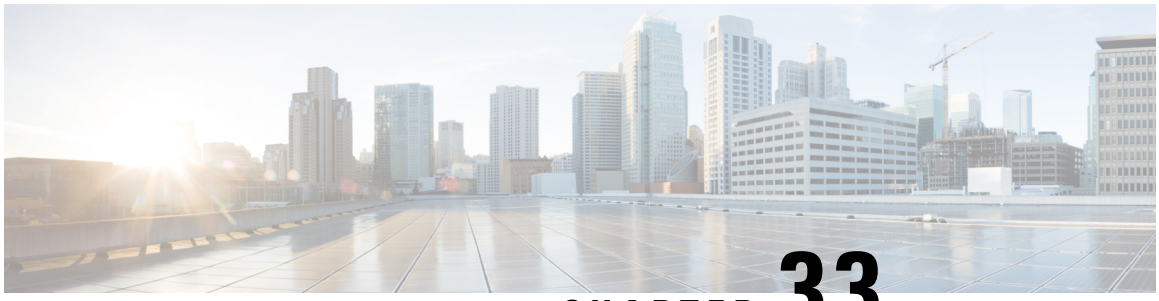
Configuring Resilient Ethernet Protocol (REP)

Resilient Ethernet Protocol (REP) is a Cisco proprietary protocol that provides an alternative to Spanning Tree Protocol (STP) to control network loops, handle link failures, and improve convergence time.

For detailed information on configuring REP, see the Configuring Resilient Ethernet Protocol chapter in *Redundancy Protocol Configuration Guide, Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches*.

Note The following features are not supported on IR8340:

- REP Fast
- REP Day Zero
- REP Segment Id Auto Discovery
- REP Negotiated



CHAPTER 33

System Messages

This section lists and explains the system messages that occur during device operation.

- Process management systems, on page 399
- Error message details, on page 399

Process management systems

You can access system messages by logging in to the console through Telnet protocol and monitoring your system components remotely from any workstation that supports the Telnet protocol.

Starting and monitoring software is referred to as process management. The process management infrastructure for a router is platform independent, and error messages are consistent across platforms running on Cisco IOS XE. You do not have to be directly involved in process management, but we recommend that you read the system messages that refer to process failures and other issues.

Error message details

To show further details about a process management or a syslog error message, enter the error message into the Error Message Decoder tool at: https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi . For example, enter the message `%PMAN-0-PROCESS_NOTIFICATION` into the tool to view an explanation of the error message and the recommended action to be taken.

Table 25: Error Message Details

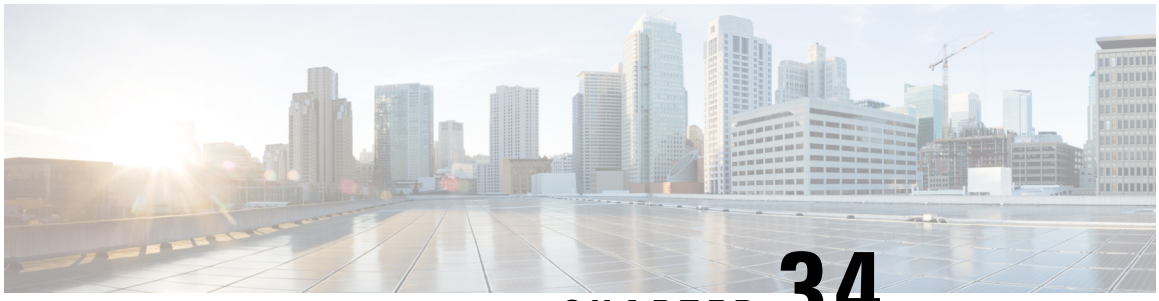
Error Message	Explanation	Recommended Action
`%PMAN-0-PROCESS_NOTIFICATION : The process lifecycle notification component failed because [chars]`	The process lifecycle notification component failed, preventing proper detection of a process start and stop. This problem is likely the result of a software defect in the software subpackage.	Note the time of the message and investigate the kernel error message logs to learn more about the problem and see if it is correctable. If the problem cannot be corrected or the logs are not helpful, copy the error message exactly as it appears on the console along with the output of the **show tech-support** command and provide the gathered information to a Cisco technical support representative.
`%PMAN-0-PROCFAILCRIT A critical process [chars] has failed (rc [dec])`	A process important to the functioning of the router has failed.	Note the time of the message and investigate the error message logs to learn more about the problem. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac . Search for resolved software issues using the Bug Search Tool at: https://bst.cloudapps.cisco.com/bugsearch . If you still require assistance, open a case with the Technical Assistance Center or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message	Explanation	Recommended Action
`%PMAN-3-PROCFAILOPT An optional process [chars] has failed (rc [dec])`	A process that does not affect the forwarding of traffic has failed.	Note the time of the message and investigate the kernel error message logs to learn more about the problem. Although traffic will still be forwarded after receiving this message, certain functions on the router may be disabled because of this message and the error should be investigated. If the logs are not helpful or indicate a problem you cannot correct, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at http://www.cisco.com/tac. Search for resolved software issues using the Bug Search Tool at: https://bst.cloudapps.cisco.com/bugsearch . If you still require assistance, open a case with the Technical Assistance Center or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message	Explanation	Recommended Action
`%PMAN-3-PROCFAIL The process [chars] has failed (rc [dec])`	The process has failed as the result of an error.	This message will appear with other messages related to the process. Check the other messages to determine the reason for the failures and see if corrective action can be taken. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac . Search for resolved software issues using the Bug Search Tool at: https://bst.cloudapps.cisco.com/bugsearch . If you still require assistance, open a case with the Technical Assistance Center or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.
`%PMAN-3-PROCFAIL_IGNORE [chars] process exits and failures are being ignored due to debug settings.`	A process failure is being ignored due to the user-configured debug settings.	If this behavior is desired and the debug settings are set according to a user's preference, no action is needed. If the appearance of this message is viewed as a problem, change the debug settings. The router is not expected to behave normally with this debug setting. Functionalities such as SSO switchover, router reloads, FRU resets, and so on will be affected. This setting should only be used in a debug scenario.

Error Message	Explanation	Recommended Action
%PMAN-3-PROCHOLDDOWN The process [chars] has been helddown (rc [dec])	The process was restarted too many times with repeated failures and has been placed in the hold-down state.	This message will appear with other messages related to the process. Check the other messages to determine the reason for the failures and see if corrective action can be taken. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac . Search for resolved software issues using the Bug Search Tool at: https://bst.cloudapps.cisco.com/bugsearch . If you still require assistance, open a case with the Technical Assistance Center or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.
%PMAN-3-RELOAD_RP_SB_NOT_READY : Reloading: [chars]	The route processor is being reloaded because there is no ready standby instance.	Ensure that the reload is not due to an error condition.
%PMAN-3-RELOAD_RP : Reloading: [chars]	The RP is being reloaded.	Ensure that the reload is not due to an error condition. If it is due to an error condition, collect information requested by the other log messages.
%PMAN-3-RELOAD_SYSTEM : Reloading: [chars]	The system is being reloaded.	Ensure that the reload is not due to an error condition. If it is due to an error condition, collect information requested by the other log messages.
%PMAN-3-PROC_BAD_EXECUTABLE : Bad executable or permission problem with process [chars]	The executable file used for the process is bad or has permission problem.	Ensure that the named executable is replaced with the correct executable.

Error Message	Explanation	Recommended Action
%PMAN-3-PROC_BAD_COMMAND:Non-existent executable or bad library used for process <process name>	The executable file used for the process is missing, or a dependent library is bad.	Ensure that the named executable is present and the dependent libraries are good.
%PMAN-3-PROC_EMPTY_EXEC_FILE : Empty executable used for process [chars]	The executable file used for the process is empty.	Ensure that the named executable is non-zero in size.
%PMAN-5-EXITACTION : Process manager is exiting: [chars]	The process manager is exiting.	Ensure that the process manager is not exiting due to an error condition. If it is due to an error condition, collect information requested by the other log messages.
%PMAN-6-PROCSHUT : The process [chars] has shutdown	The process has gracefully shut down.	No user action is necessary. This message is provided for informational purposes only.
%PMAN-6-PROCSTART : The process [chars] has started	The process has launched and is operating properly.	No user action is necessary. This message is provided for informational purposes only.
%PMAN-6-PROCSTATELESS : The process [chars] is restarting stateless	The process has requested a stateless restart.	No user action is necessary. This message is provided for informational purposes only.



CHAPTER 34

Environmental Monitoring

This chapter contains the following sections:

- Environmental monitoring, on page 405
- Environmental Monitoring and Reporting Functions, on page 405
- Environmental Monitoring Functions, on page 406
- Environmental Reporting Functions, on page 407
- Additional References, on page 407

Environmental monitoring

The router provides a robust environment-monitoring system with several sensors that monitor the system temperatures. The following are some of the key functions of the environmental monitoring system:

- Monitoring temperature of CPUs and Motherboard
- Recording abnormal events and generating notifications
- Monitoring Simple Network Management Protocol (SNMP) traps
- Generating and collecting Onboard Failure Logging (OBFL) data
- Sending call home event notifications
- Logging system error messages
- Displaying present settings and status

Environmental Monitoring and Reporting Functions

Monitoring and reporting functions allow you to maintain normal system operation by identifying and resolving adverse conditions prior to loss of operation.

- Environmental Monitoring Functions, on page 406
- Environmental Reporting Functions, on page 407

Environmental Monitoring Functions

Environmental monitoring functions use sensors to monitor the temperature of the cooling air as it moves through the chassis.

The router is expected to meet the following environmental operating conditions

- Operating Temperature: -40°F to 140°F (-40°C to 60°C)
- Operating Humidity: 5% to 95% relative humidity (non-condensing)
- Operating Altitude: Up to 10,000 ft (3048 m)



Note Refer to the *Cisco Catalyst IR8340 Rugged Series Router Harware Installation Guide* for restricted ranges of the environmental operating conditions for exceptions which is specially applicable for harsh industrial deployments.

The following table displays the levels of status conditions used by the environmental monitoring system.

Table 26: Levels of Status Conditions Used by the Environmental Monitoring System

Status Level	Description
Normal	All monitored parameters are within normal tolerance.
Warning	The system has exceeded a specified threshold. The system continues to operate, but operator action is recommended to bring the system back to a normal state.
Critical	An out-of-tolerance temperature or voltage condition exists. Although the system continues to operate, it is approaching shutdown. Immediate operator action is required.

The environmental monitoring system sends system messages to the console, for example, when the conditions described here are met:

Temperature and Voltage Exceed Max/Min Thresholds

The following example shows the warning messages indicating the maximum and minimum thresholds of the temperature or voltage:

```
Warnings :
--------
For all the temperature sensors (name starting with “Temp:”) above,
the critical warning threshold is 100C (100C and higher)
the warning threshold is 80C (range from 80C to 99C)
the low warning threshold is 1C (range from -inf to 1C).

For all voltage sensors (names starting with "V:"),
the high warning threshold starts at that voltage +10%. (voltage + 10% is warning)
the low warning threshold starts at the voltage -10%. (voltage - 10% is warning)
```

Environmental Reporting Functions

You can retrieve and display environmental status reports using the following commands:

- **show diag all eeprom**
- **show environment**
- **show environment all**
- **show inventory**
- **show platform**
- **show platform diag**
- **show platform software status control-processor**
- **show diag slot R0 eeprom detail**
- **show version**
- **show power**

These commands show the current values of parameters such as temperature and voltage.

The environmental monitoring system updates the values of these parameters every 60 seconds.

Additional References

The following sections provide references related to the power efficiency management feature.

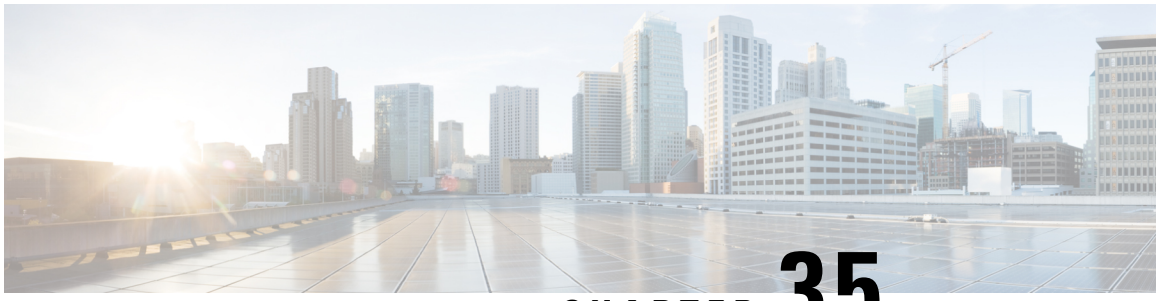
MIBs

MIBs	MIBs Link
CISCO-ENTITY-FRU-CONTROL-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use the Cisco MIB Locator at: http://www.cisco.com/go/mibs.

Technical assistance

This reference provides the primary link and description for accessing Cisco technical support resources and subscription services.

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 35

IOx Application Hosting

This section contains the following topics:

- Application Hosting, on page 409
- Application hosting, on page 409
- Application hosting on the IR8340 router, on page 411
- Application hosting configuration, on page 414
- Install and uninstall apps, on page 419
- Override the app resource configuration, on page 420
- Override the app resource configuration, on page 421
- Configure IOx with ERSPAN, on page 422
- Configuration examples for application hosting, on page 423
- Signed application support, on page 424
- Cisco Cyber Vision, on page 425
- Cisco ThousandEyes Enterprise Agent, on page 425

Application Hosting

A hosted application is a software as a service solution that runs remotely using commands.

- Provides a platform for administrators to leverage their own tools and utilities.

Application hosting

Application hosting is a platform capability that:

- enables deployment and management of third-party applications on network devices,
- provides virtualized runtime environments for applications, and
- extends device functionality beyond traditional networking capabilities.

Application hosting

Application hosting is a platform that:

- enables administrators to leverage their own tools and utilities on network devices,
- provides reusable, portable, and scalable applications for virtual environments, and
- supports third-party off-the-shelf applications built using Linux tool chains.

Application hosting capabilities

Applications hosted on network devices can serve various purposes:

- Automation.
- Configuration management monitoring.
- Integration with existing tool chains.

Cisco devices support custom applications cross-compiled with the software development kit that Cisco provides.

IOx

IOx is a Cisco-developed end-to-end application framework that provides application hosting capabilities for different application types on Cisco network platforms.

Storage device requirements

From Cisco IOS-XE Release 17.8.1, IOx installation on IR8340 requires Cisco supported mSATA to be the storage device. There are two partitions on the mSATA. One for IOS, and the other for IOx. The IOS partition will be mounted on **/mnt/msata** and IOx partition will be mounted on **/vol/harddisk**. OIR is not supported for mSATA device. When mSATA is inserted to the router, the router needs to be reloaded to have two partitions. If mSATA is not present, bootflash is used for application hosting.

In Cisco IOS XE Release 17.7.x, SD card is used as storage device for IOx. Any upgrade from 17.7.x to 17.8.x requires all applications to be reinstalled.

Cisco application hosting

Cisco application hosting is a virtualization infrastructure that:

- launches designated applications in containers,
- checks available resources (memory, CPU, and storage), and allocates and manages them,
- provides support for console logging and CLI endpoint access, and
- provides an application hosting infrastructure referred to as Cisco Application Framework (CAF).

Application hosting components

Application hosting provides these additional services:

- Helps in the setup of platform-specific networking (packet-path) via VirtualPortGroup and management interfaces.

The container is referred to as the virtualization environment provided to run the guest application on the host operating system. The Cisco IOS-XE virtualization services provide manageability and networking models for running guest applications. The virtualization infrastructure allows the administrator to define a logical interface that specifies the connectivity between the host and the guest. IOx maps the logical interface into the Virtual Network Interface Card (vNIC) that the guest application uses.

Applications to be deployed in the containers are packaged as TAR files. The configuration that is specific to these applications is also packaged as part of the TAR file.

The management interface on the device connects the application hosting network to the IOS management interface. The Layer 3 interface of the application receives the Layer 2 bridged traffic from the IOS management interface. The management interface connects through the management bridge to the container/application interface. The IP address of the application must be on the same subnet as the management interface IP address.

IOXMAN

IOXMAN is a process that:

- establishes a tracing infrastructure to provide logging or tracing services for guest applications, except Libvirt, that emulates serial devices,
- operates based on the lifecycle of the guest application to enable and disable the tracing service,
- sends logging data to IOS syslog, saves tracing data to IOx tracelog, and maintains IOx tracelog for each guest application.

Application hosting on the IR8340 router

Application hosting on the IR8340 router is a deployment capability that:

- enables applications to be deployed using application hosting CLI commands,
- supports deployment through Local Manager WebUI interface, and
- requires WebUI to be enabled for Local Manager access.

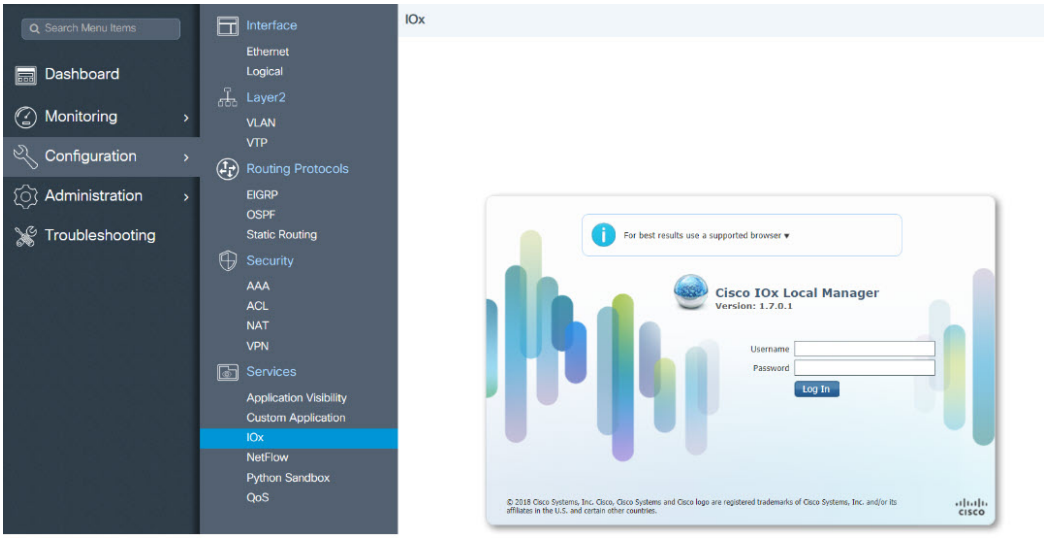
Application deployment methods

Application hosting can be achieved using two primary methods:

- Application hosting CLI commands.
- Local Manager through WebUI.

To deploy applications using Local Manager, enable WebUI and then log in to Local Manager.

Figure 9: Local Manager



For Local Manager deployment you can:

1. Navigate to **Configuration > Services > IOx** from WebUI.
2. Log in using the configured username and password.
3. Follow the application lifecycle management steps. For more details, see Cisco IOx Local Manager Reference Guide.

The next section explains the deployment of an application using the application hosting CLI commands.

Application hosting on Layer 2 and Layer 3 interfaces

Application hosting on Layer 2 and Layer 3 interfaces is a network configuration approach that:

- uses two interface types to support L2 and L3 traffic from LAN and WAN ports respectively,
- configures L2 interfaces with AppGigabitEthernet and VLAN with IP address in the same VLAN network for forwarding L2 app traffic, and,
- uses L3 gateway interfaces configured with Virtual port group and IP address in the same network as VPG for forwarding L3 traffic to applications.

Configuration requirements

For application hosting, you can configure the L2 and L3 interfaces as given:

- L2 interfaces are configured with AppGigabitEthernet and VLAN with IP address in the same VLAN network, which are used or forwarding the L2 app traffic. Dedicated VLAN range 2340 - 2349 must be used for configuring L2 interfaces of application and to communicate the application for L2 traffic.

  You should configure the AppGigEthernet interface as a trunk interface.

- L3 interfaces or gateway interfaces are configured with Virtual port group, and IP address in the same network as VPG, which are used for forwarding the L3 traffic to applications.

VirtualPortGroup

A VirtualPortGroup is a software construct on Cisco IOS that:

- maps to a Linux bridge IP address,
- represents the switch virtual interface (SVI) of the Linux container,
- connects the application hosting network to the IOS routing domain.

VirtualPortGroup configuration and networking details

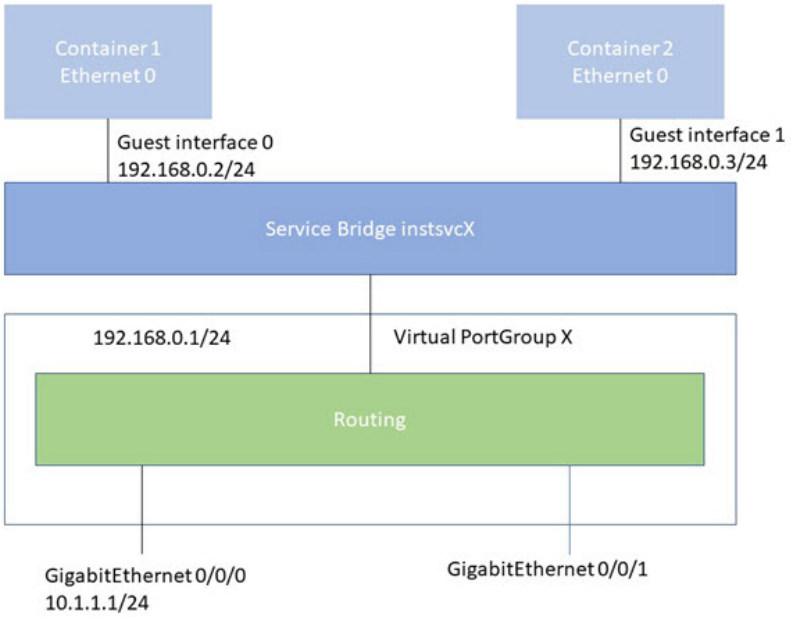
Each bridge can contain multiple interfaces; each mapping to a different container. Each container can also have multiple interfaces.

VirtualPortGroup interfaces are configured by using the interface virtualportgroup command. Once these interfaces are created, IP address and other resources are allocated.

The VirtualPortGroup interface connects the application hosting network to the IOS routing domain. The Layer 3 interface of the application receives routed traffic from IOS. The VirtualPortGroup interface connects through the SVC Bridge to the container/application interface.

The following graphic helps to understand the relationship between the VirtualPortGroup and other interfaces.

Figure 10: Virtual Port Group Mapping



vNICs

A vNIC is a virtual Ethernet interface that:

- connects application containers to the platform dataplane for sending and receiving packets,
- forms part of a virtual Ethernet pair where the other interface (vpgX) belongs to the host system, and

- operates as a standard Ethernet interface inside the container with gateway, IP address, and unique MAC address assigned by IOx.

vNIC implementation details

The Layer 3 routing model supports one container per internal logical interface for container lifecycle management. IOx is responsible for the gateway (VirtualPortGroup interface), IP address, and unique MAC address assignment for each vNIC in the container.

Application hosting configuration

The following sections provide information about the various tasks that comprise the configuration of application hosting.

Enable IOx

This task enables access to the IOx Local Manager. The IOx Local Manager provides a web-based user interface that you can use to manage, administer, monitor, and troubleshoot apps on the host system, and to perform a variety of related activities.

Note In the steps that follow, IP HTTP commands do not enable IOx, but allow the user to access the WebUI to connect the IOx Local Manager.

Procedure

Step 1 Use the **enable** command to enter privileged EXEC mode.

Example:

```
Router> enable
                          Router#
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
                          Router(config)#
```

Step 3 Use the **iox** command to enable the IOx framework.

Example:

```
Router(config)# iox
```

Step 4 Use the **ip http server** command to enable the HTTP server.

Example:

```
Router(config)# ip http server
```

Step 5 Use the **ip http secure-server** command to enable the HTTPS server.

Example:

```
Router(config)# ip http secure-server
```

Note The IP HTTP commands do not enable IOx itself; they allow you to access the WebUI to connect to the IOx Local Manager.

Step 6 Use the **username** *name* **privilege** *level***secret** {**0** | **7** | *user-password*} *encrypted-password* command to establish a username-based authentication system.

Example:

```
Router(config)# username cisco privilege 15 secret 0 cisco
```

Note The username privilege level must be configured as 15 to ensure full administrative access.

Step 7 Use the **end** command to exit configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# end
```

Configure application hosting to Layer 2 interfaces

Use this task to configure the AppGigabitEthernet interface and associate it with an application for Layer 2 connectivity.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
                        Router(config)#
```

Step 2 Use the **interface AppGigabitEthernet** *number* command to configure the interface and enter interface configuration mode.

Example:

```
Router(config)# interface AppGigabitEthernet 0/1/1
```

Step 3 Use the **switchport mode trunk** command to set the interface to trunking mode.

Example:

```
Router(config-if)# switchport mode trunk
```

Step 4 Use the **exit** command to return to global configuration mode.

Example:

```
Router(config-if)# exit
```

Step 5 Use the **app-hosting appid** *name* command to configure the application and enter application hosting configuration mode.

Example:

```
Router(config)# app-hosting appid iperf_3
```

Step 6 Use the **app-vnic AppGigabitEthernet trunk** command to configure a trunk port for the application.

Example:

```
Router(config-app-hosting)# app-vnic AppGigabitEthernet trunk
```

Step 7 Use the **vlan** *vlan-id* **guest-interface** *interface-number* command to configure a VLAN guest interface.

Example:

```
Router(config-app-hosting-trunk)# vlan 2340 guest-interface 1
```

Step 8 Use the **guest-ipaddress** *ip-address* **netmask** *mask* command to configure a static IP address for the guest interface.

Example:

```
Router(config-app-hosting-vlan-access-ip)# guest-ipaddress 20.1.1.2 netmask 255.255.255.0
```

Step 9 Use the **end** command to exit to privileged EXEC mode.

Example:

```
Router(config-config-app-hosting-vlan-access-ip)# end
```

Configure a VirtualPortGroup to a Layer 3 data port

Multiple Layer 3 data ports can be routed to one or more VirtualPortGroups or containers. VirtualPortGroups and Layer 3 data ports must be on different subnets.

Note The **ip routing** command must be enabled to allow external routing on Layer 3 data ports.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
                        Router(config)#
```

a) Use the **ip routing** command to enable IP routing.

Example:

```
Router(config)# ip routing
```

b) Use the **interface** *type number* command to configure the interface.

Example:

```
Router(config)# interface gigabitethernet 0/0/0
```

c) Use the **no switchport** command to place the interface in Layer 3 mode.

Example:

```
Router(config-if)# no switchport
```

d) Use the **ip address** *ip-address mask* command to configure an IP address for the interface.

Example:

```
Router(config-if)# ip address 10.1.1.1 255.255.255.0
```

e) Use the **exit** command to return to global configuration mode.

Example:

```
Router(config-if)# exit
```

Step 2 Use the **interface virtualportgroup** *number* command to configure the interface.

Example:

```
Router(config)# interface virtualportgroup 0
```

a) Use the **ip address** *ip-address mask* command to configure an IP address for the interface.

Example:

```
Router(config-if)# ip address 20.1.2.1 255.255.255.0
```

b) Use the **end** command to return to privileged EXEC mode.

Example:

```
Router(config-if)# end
```

Step 3 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

a) Use the **interface vlan** *vlan-id* command to configure the SVI interface.

Example:

```
Router(config)# interface vlan 2340
```

b) Use the **ip address** *ip-address mask* command to configure an IP address and subnet mask.

Example:

```
Router(config-if)# ip address 20.1.1.1 255.255.255.0
```

c) Use the **end** command to return to privileged EXEC mode.

Example:

```
Router(config-if)# end
```

Step 4 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

a) Use the **app-hosting appid** *name* command to configure the application.

Example:

```
Router(config)# app-hosting appid iperf_3
```

b) Use the **app-vnic gateway2 virtualportgroup** *id* **guest-interface** *number* command to configure the application interface and gateway.

Example:

```
Router(config-app-hosting)# app-vnic gateway2 virtualportgroup 0 guest-interface 2
```

c) Use the **guest-ipaddress** *ip-address* **netmask** *mask* command to configure the application IP address.

Example:

```
Router(config-app-hosting-gateway0)# guest-ipaddress 20.1.2.2 netmask 255.255.255.0
```

d) Use the **app-default-gateway** *ip-address* **guest-interface** *number* command to configure the default gateway.

Example:

```
Router(config-app-hosting-gateway0)# app-default-gateway 20.1.2.1 guest-interface 2
```

e) Use the **end** command to return to privileged EXEC mode.

Example:

```
Router# end
```

Configure Docker run time options

Use this task to specify Docker run time options for hosted applications.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
                        Router(config)#
```

Step 2 Use the **app-hosting appid** *name* command to enable application hosting and enter application hosting configuration mode.

Example:

```
Router(config)# app-hosting appid app1
```

Step 3 Use the **app-resource docker** command to enter application-hosting docker-configuration mode.

Example:

```
Router(config-app-hosting)# app-resource docker
```

Note Application start-up scripts are activated.

Step 4 Use the **run-opts** *id options* command to specify the Docker run time options.

Example:

```
Router(config-app-hosting-docker)# run-opts 1 "-v $(APP_DATA):/data"
```

Step 5 Use the **end** command to exit to privileged EXEC mode.

Example:

```
Router(config-app-hosting-docker)# end
```

Install and uninstall apps

Use this task to install, activate, start, stop, deactivate, and uninstall applications on the host system.

Procedure

Step 1 Use the **enable** command to enter privileged EXEC mode.

Example:

```
Router> enable
                          Router#
```

Step 2 Use the **app-hosting install appid** *application-name* **package** *package-path* command to install an application.

Example:

```
Router# app-hosting install appid lxc_app package flash:my_iox_app.tar
```

Step 3 Use the **app-hosting activate appid** *application-name* command to activate the application.

Example:

```
Router# app-hosting activate appid app1
```

Note This command validates all application resource requests. If all resources are available, the application is activated; otherwise, the activation fails.

Step 4 Use the **app-hosting start appid** *application-name* command to start the application.

Example:

```
Router# app-hosting start appid app1
```

Step 5 Use the **app-hosting stop appid** *application-name* command to stop the application.

Example:

```
Router# app-hosting stop appid app1
```

Step 6 Use the **app-hosting deactivate appid** *application-name* command to deactivate all resources allocated for the application.

Example:

```
Router# app-hosting deactivate appid app1
```

Step 7 Use the **app-hosting uninstall appid** *application-name* command to uninstall the application.

Example:

```
Router# app-hosting uninstall appid app1
```

Note Uninstalling the application removes all packaging, images, and stored changes. If you reinstall the app with the same IP addresses, traffic to VirtualPortGroup interfaces may be blocked because the ARP entry is not updated. You must manually clear the ARP cache to refresh the entries.

Override the app resource configuration

Use this task to define custom resource profiles for your applications. Resource changes take effect only after the **app-hosting activate** command is configured.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
            Router(config)#
```

Step 2 Use the **app-hosting appid** *name* command to enable application hosting and enter application hosting configuration mode.

Example:

```
Router(config)# app-hosting appid app1
```

Step 3 Use the **app-resource profile custom** command to configure the custom application resource profile.

Example:

```
Router(config-app-hosting)# app-resource profile custom
```

Note Only the custom profile name is supported.

Step 4 Use the **cpu** *unit* command to change the default CPU allocation.

Example:

```
Router(config-app-resource-profile-custom)# cpu 800
```

Note Resource values are application-specific. Ensure that the application can run reliably with the adjusted values.

Step 5 Use the **memory** *memory* command to change the default memory allocation.

Example:

```
Router(config-app-resource-profile-custom)# memory 512
```

Step 6 Use the **vcpu** *number* command to change the virtual CPU (vCPU) allocation.

Example:

```
Router(config-app-resource-profile-custom)# vcpu 2
```

Step 7 Use the **end** command to exit configuration mode and return to privileged EXEC mode.

Example:

```
Router(config-app-resource-profile-custom)# end
```

Override the app resource configuration

Use this task to define custom resource profiles for your applications. Resource changes take effect only after the **app-hosting activate** command is configured.

Procedure

Step 1 Use the **enable** command to enter privileged EXEC mode.

Example:

```
Router> enable
            Router#
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
            Router(config)#
```

Step 3 Use the **app-hosting appid** *name* command to enable application hosting and enter application hosting configuration mode.

Example:

```
Router(config)# app-hosting appid app1
```

Step 4 Use the **app-resource profile custom** command to configure the custom application resource profile.

Example:

```
Router(config-app-hosting)# app-resource profile custom
```

Note Only the custom profile name is supported.

Step 5 Use the **cpu** *unit* command to change the default CPU allocation.

Example:

```
Router(config-app-resource-profile-custom)# cpu 800
```

Note Resource values are application-specific. Ensure that the application can run reliably with the adjusted values.

Step 6 Use the **memory** *memory* command to change the default memory allocation.

Example:

```
Router(config-app-resource-profile-custom)# memory 512
```

Step 7 Use the **vcpu** *number* command to change the virtual CPU (vCPU) allocation.

Example:

```
Router(config-app-resource-profile-custom)# vcpu 2
```

Step 8 Use the **end** command to exit configuration mode and return to privileged EXEC mode.

Example:

```
Router(config-app-resource-profile-custom)# end
```

Configure IOx with ERSPAN

Use this task to span traffic to IOx applications. Access Control Lists (ACLs) can be applied to ERSPAN traffic similarly to FSPAN.

Procedure

Step 1 Use the **ip access-list extended** *name* command to create an extended access list.

Example:

```
Router(config)# ip access-list extended ACL120
                        Router(config-ext-nacl)# 10 permit ip host 120.1.1.1 host
120.120.120.120
```

Step 2 Use the **monitor session 1** *session-id* **type erspan-source** command for configuring ERSPAN session for LAN or WAN ports to span data to the application.

- Configure ERSPAN session for LAN ports to span data to the application.

Note ERSPAN Session ID 1 - 4 are only supported on LAN ports.

```
monitor session 1 type erspan-source
source interface Gi0/1/10 rx
filter access-group ACL120
destination
erspan-id 1
ip address 20.1.2.2 <== Ip address of L2/VLAN interface on APP
origin ip address 68.68.68.68
```

- configuring ERSPAN session for WAN ports to span data to the application.

```
monitor session 1 type erspan-source
source interface Gi0/0/0 rx
filter access-group ACL120
destination
erspan-id 1
ip address 20.1.1.2 <== Ip address of L3 interface on APP
origin ip address 68.68.68.68
```

Configuration examples for application hosting

See the examples:

Example: Enable IOx

This example demonstrates the command sequence required to enable IOx functionality on a Cisco device, including the necessary HTTP server configuration and user account setup.

```
Device> enable
Device# configure terminal
Device(config)# iox
Device(config)# ip http server
Device(config)# ip http secure-server
Device(config)# username cisco privilege 15 secret 0 cisco
Device(config)# end
```

VirtualPortGroup to Layer 3 data port configuration example

This example demonstrates the configuration steps for setting up a VirtualPortGroup interface and connecting it to a Layer 3 data port with IP addressing.

```
Device> enable
Device# configure terminal
Device(config)# ip routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# no switchport
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# exit
```

```
Device(config)# interface virtualportgroup 0
Device(config-if)# ip address 192.168.0.1 255.255.255.0
Device(config-if)# end
```

App installation and uninstallation examples

This reference provides command-line examples that demonstrate the complete lifecycle of app management, including installation, activation, start, stop, deactivation, and uninstallation of IOx applications.

```
Device> enable
Device# app-hosting install appid app1 package flash:my_iox_app.tar
Device# app-hosting activate appid app1
Device# app-hosting start appid app1
Device# app-hosting stop appid app1
Device# app-hosting deactivate appid app1
Device# app-hosting uninstall appid app1
```

App resource configuration override example

This example demonstrates how to override the default app resource configuration by creating a custom profile with specific CPU, memory, and virtual CPU settings for an application.

```
Device# configure terminal
Device(config)# app-hosting appid app1
Device(config-app-hosting)# app-resource profile custom
Device(config-app-resource-profile-custom)# cpu 800
Device(config-app-resource-profile-custom)# memory 512
Device(config-app-resource-profile-custom)# vcpu 2
Device(config-app-resource-profile-custom)# end
```

Signed application support

Signed application support is an IOX security feature that:

- requires signed verification to be enabled on the device to install signed applications,
- prevents unsigned applications from being activated when verification is enabled, and
- allows signed applications to move to different states regardless of the verification status.

Signed verification commands

To install a signed application, signed verification has to be enabled on the device.

You can enable or disable signed verification using the **app-hosting verification** {**enable**|**disable**} command.

The signed verification enabled or disabled status can be verified by the **show app-hosting infra** command:

```
# show app-hosting infra
IOX version: 2.7.0.0
App signature verification: disabled
Internal working directory: /vol/harddisk/iox

Application Interface Mapping
AppGigabitEthernet Port # Interface Name Port Type Bandwidth
```

```
1 AppGigabitEthernet0/1/1 KR Port - Internal 10G


CPU:
Quota: 99(Percentage)
Available: 99(Percentage)
Quota: 3465(Units)
Available: 0(Units)
```

When signed verification is enabled, any unsigned app can not be activated, and signed app can move to different states irrespective of the app sign verification enabled or disabled.

After enabling the signed verification, follow the instructions in Install and uninstall apps, on page 419 to install the application.

Cisco Cyber Vision

Cisco Cyber Vision Center (CVC) is a network security solution that:

- gives more visibility into Industrial IoT networks across Industrial Control Systems (ICS) with real-time monitoring of control and data networks,
- supports integration on IoT IOS-XE platforms beginning with release 17.4 by deploying IOX Cyber Vision sensor, and
- enables platforms to forward traffic from IOX applications to Cyber Vision Center for real-time monitoring and captured PCAP files to Vision center from IOX application.

Additional information

The minimum Cyber Vision release is 4.1.1 to work with the IR8340. For more information about CVC, see the release notes:

https://www.cisco.com/c/en/us/support/security/cyber-vision/products-release-notes-list.html

For more information about CVC installation and ERSPAN with CVC, see the Cisco Cyber Vision Network Sensor Installation Guide for Cisco IR8340.

Cisco ThousandEyes Enterprise Agent

A Cisco ThousandEyes Enterprise Agent is a network monitoring solution that

- runs a variety of tests using agents
- provides real-time monitoring of network and application performance, and
- offers multidimensional insights with routing and device data for end-to-end visibility.

Additional information

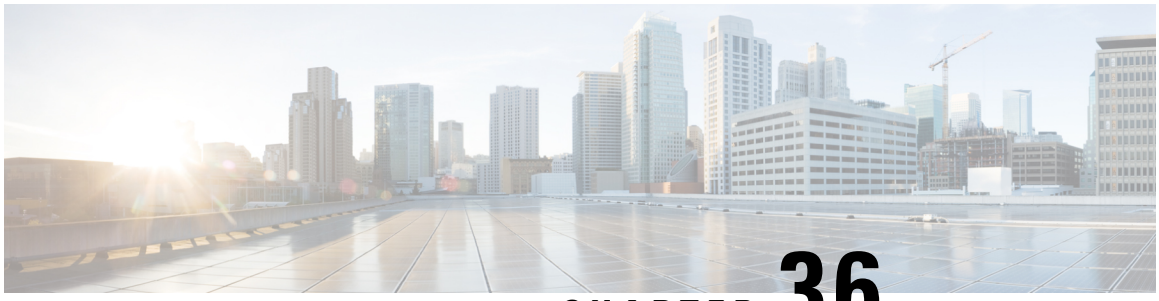
The Cisco ThousandEyes Enterprise Agent enables you to view end-to-end paths across networks and services affecting your business. It actively monitors network traffic paths across internal, external, and internet networks, helping analyze network performance and application availability.

You can use application-hosting features to deploy the Cisco ThousandEyes Enterprise Agent as a container application on Cisco Industrial IoT Routers. The agent runs as a Docker container using the Cisco IOx.

Starting with Cisco IOS XE Release 17.18.2, you can configure Cisco ThousandEyes Enterprise Agent in Controller mode.

For details about configuring Cisco ThousandEyes Enterprise Agent in controller mode, see Cisco SD-WAN Systems and Interfaces Configuration Guide.

For more information about Cisco ThousandEyes Enterprise Agent, see ThousandEyes Documentation.



CHAPTER 36

ROM Monitor Overview

- ROM Monitor, on page 427
- Access ROM Monitor mode, on page 428
- Display the Configuration Register Setting, on page 431
- Environment variable settings, on page 432
- Exit ROM Monitor Mode, on page 434

ROM Monitor

A ROM Monitor is a bootstrap program that

- initializes the hardware and boots the Cisco IOS XE software when you power on or reload a router
- displays the ROM Monitor (rommon 1>) prompt when you connect a terminal to the router in ROM Monitor mode, and
- operates as a separate program from the Cisco IOS XE software for special circumstances such as reinstalling software, resetting passwords, or specifying startup configuration files.

The ROM Monitor software is known by many names. It is sometimes called *ROMMON* because of the CLI prompt in ROM Monitor mode. The ROM Monitor software is also called the *boot software*, *boot image*, or *boot helper*.

ROM Monitor operation and configuration

During normal operation, users do not use ROM Monitor mode. ROM Monitor mode is used only in special circumstances, such as reinstalling the entire software set, resetting the router password, or specifying a configuration file to use at startup.

Although it is distributed with routers that use the Cisco IOS XE software, ROM Monitor is a separate program from the Cisco IOS XE software. During normal startup, the ROM Monitor initializes the router, and then control passes to the Cisco IOS XE software. After the Cisco IOS XE software takes over, the ROM Monitor is no longer in use.

Two primary connections exist between ROM Monitor and the Cisco IOS XE software: the ROM Monitor environment variables and the configuration register.

The ROM Monitor environment variables define the location of the Cisco IOS XE software and describe how to load it. After the ROM Monitor has initialized the router, it uses the environment variables to locate and load the Cisco IOS XE software.

The *configuration register* is a software setting that controls how a router starts up. One of the primary uses of the configuration register is to control whether the router starts in ROM Monitor mode or Administration EXEC mode. The configuration register is set in either ROM Monitor mode or Administration EXEC mode as needed. Typically, you set the configuration register using the Cisco IOS XE software prompt when you need to use ROM Monitor mode. When the maintenance in ROM Monitor mode is complete, you change the configuration register so the router reboots with the Cisco IOS XE software.

When the router is in ROM Monitor mode, you can access the ROM Monitor software only from a terminal connected directly to the console port of the card. Because the Cisco IOS XE software (EXEC mode) is not operating, nonmanagement interfaces are not accessible. Basically, all Cisco IOS XE software resources are unavailable. The hardware is available, but no configuration exists to make use of the hardware.

It is important to remember that ROM Monitor mode is a router mode, not a mode within the Cisco IOS XE software. It is best to remember that ROM Monitor software and the Cisco IOS XE software are two separate programs that run on the same router. At any given time, the router runs only one of these programs.

One area that can be confusing when using ROM Monitor and the Cisco IOS XE software is the area that defines the IP configuration for the Management Ethernet interface. Most users are comfortable with configuring the Management Ethernet interface in the Cisco IOS XE software. When the router is in ROM Monitor mode, however, the router does not run the Cisco IOS XE software, so that Management Ethernet interface configuration is not available.

When you want to access other devices, such as a TFTP server, while in ROM Monitor mode on the router, you must configure the ROM Monitor variables with IP access information.



Note TFTP access variables are currently not supported on the IR8340 platform.

Access ROM Monitor mode

This reference provides the necessary steps to enter ROM Monitor (ROMMON) mode on your device.

Check the Current ROMMON Version

Perform this task to determine the current ROMmon version and review configured variables for troubleshooting, compatibility verification, or system maintenance.

ROMmon (ROM Monitor) is the bootstrap program that initializes the router hardware and loads the Cisco IOS software.

Before you begin

The device must be powered on and you must have access to the CLI in either privileged EXEC mode or ROMmon mode.

Follow these steps to check the ROMmon version and variables:

Procedure

Step 1 Use the **show rom-monitor** *slot* command to display the current ROMmon version.

Example:

```
Router# show rom-monitor r0
============================================================

System Bootstrap, Version v0.33, DEVELOPMENT SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.
Compiled Mon Jul 12 18:53:57 2021 by gilchen
```

Step 2 Use the **show romvar** command to display all configured ROMmon variables in privileged EXEC mode.

Example:

```
Router# show romvar
ROMMON variables:
PS1 = rommon ! >
DEVICE_MANAGED_MODE = autonomous
CRYPTO_BI_THPUT = 50000
TEMPLATE = advanced
DISABLE_USB_FLASH = no
DISABLE_CC_AUTH = 1
DISABLE_SD_FLASH = no
PASSWD_RECOVERY = no
ENABLE_BREAK = no
ENABLE_FLASH_PRIMARY_BOOT = no
BAUD = 9600
LICENSE_BOOT_LEVEL =
SKIP_CHIPGUARD = 1
RET_2_RTS =
BOOT = flash:ir8340-universalk9.BLD_POLARIS_DEV_LATEST_20210322_153228.SSA.bin,12;
BSI = 0
RET_2_RCALTS =
RANDOM_NUM = 404654831
```

If the configuration register is set to `0x0` or `0x1820`, a reload operation brings the router to the ROMmon mode prompt (`rommon 1>`).

Step 3 Alternatively, if the router is in ROMmon mode, use the **set** command to display the variables.

Example:

```
rommon 2 > set
BOOT=flash:ir8340-universalk9.BLD_POLARIS_DEV_LATEST_20210710_191629_V17_7_0_60.SSA.bin,12;
BSI=0
CRYPTO_BI_THPUT=50000
DEVICE_MANAGED_MODE=autonomous
LICENSE_BOOT_LEVEL=
PS1=rommon ! >
RANDOM_NUM=1822515242
RET_2_RCALTS=1630484680
RET_2_RTS=
SKIP_CHIPGUARD=1
TEMPLATE=advanced
rommon 3 >
```

The output provides comprehensive system bootstrap details, including the software version and environmental variables used during the boot process.

ROM Monitor commands

The following table summarizes the commands commonly used in ROM Monitor. For specific instructions on using these commands, refer to the relevant procedure in this document.

Table 27: Commonly Used ROM Monitor Commands

ROMMON Command	Description
boot image	Manually boots a Cisco IOS XE software image.
boot image –o config-file-path	Manually boots the Cisco IOS XE software with a temporary alternative administration configuration file.
confreg	Changes the config-register setting.
dev	Displays the available local storage devices.
dir	Displays the files on a storage device.
reset	Resets the node.
set	Displays the currently set ROM Monitor environmental settings.
sync	Saves the new ROM Monitor environmental settings.
unset	Removes an environmental variable setting.

Examples

The example displays what appears when you enter the **?** command on a router:

```
rommon 3 >?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
alias set and/or display command aliases
boot boot up an external process
cpuinfo display processor information
confreg configuration register utility
ctchallenge Generate a challenge for dev key install
ctinstkey Parse the response and install dev key
ctiskeyinstalled checks if devkey installed and display it
ctremkey Remove the devkey from Aikido
ctrespsave Store parts of incoming response
dev display the device table
dir list files in a file system
efi_shell launch the UEFI SHELL enviroment
flwr burn new BIOS onto the bootrom
help monitor built-in command(s) help
history display monitor command history
meminfo display main memory information
repeat repeat a monitor command
reset system reset
set display the monitor environment
showmon display currently selected ROM monitor
sync write monitor environment to NVRAM
```

```
token display board's unique token identifier
unalias unset an alias
unset unset a variable from the monitor environment
rommon 4 >
```

Change the ROM Monitor prompt

Changing the prompt is useful if you are working with multiple routers in ROM Monitor at the same time.

Follow these steps to change the ROM Monitor prompt:

Procedure

Use the **PS1=** command to specify a custom prompt format.

Example:

```
rommon 4 >PS1="IR8340 rommon ! >"
IR8340 rommon 5 >
```

This example specifies that the prompt should be "IR8340 rommon ", followed by the line number, and then followed by " > " by the line number.

Display the Configuration Register Setting

Use this task to verify the current configuration register setting and display the current environment variable settings.

Before you begin

The router must be in ROMmon mode to perform this task.

Procedure

Use the **showmon** command without parameters to display the system bootstrap information and configuration register.

Example:

```
IR8340 rommon 8 >showmon

System Bootstrap, Version v0.33, DEVELOPMENT SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.
Compiled Mon Jul 12 18:53:57 2021 by root


!!! DEBUG SECURE-BOOT CPLD Version Installed. For INTERNAL USE ONLY !!!

Current image running : Boot ROM0

Last reset cause (0x00000002): LocalSoft
IR8340-K9 platform with 8388608 Kbytes of main memory
```

```
IR8340 rommon 9 >
```

The configuration register setting is typically labeled as **Virtual Configuration Register** in the output. You can use the **no** command if you need to exit a prompt without changing the configuration register settings.

Environment variable settings

Environment variable settings are ROM Monitor configuration parameters that:

- define the attributes of the ROM Monitor,
- are entered like commands and are always followed by the equal sign (=), and
- use capital letters for the variable name, followed by a definition.

Environment variable characteristics

Under normal operating conditions, you do not need to modify these variables. They are cleared or set only when you need to make changes to the way ROM Monitor operates.

Environment variable settings are entered in capital letters, followed by a definition. For example:

```
IP_ADDRESS=10.0.0.2
```

Frequently used environmental variables

The following table shows the main ROM Monitor environmental variables. For instructions on how to use these variables, see the relevant instructions in this document. The IR8340 boot loader does not support netboot, so any setting like environment variables IP_ADDRESS, IP_SUBNET_MASK, DEFAULT_GATEWAY, TFTP_SERVER, TFTP_FILE are not used.

Table 28: Frequently Used ROM Monitor Environmental Variables

Environmental variable	Description
BOOT=path/file	Identifies the boot software for a node. This variable is usually set automatically when the router boots.

Display environment variable settings

Procedure

Enter the **showmon** command to display the current environment variable settings.

Example:

```
IR8340 rommon 8 >showmon


System Bootstrap, Version v0.33, DEVELOPMENT SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.
Compiled Mon Jul 12 18:53:57 2021 by root


!!! DEBUG SECURE-BOOT CPLD Version Installed. For INTERNAL USE ONLY !!!

Current image running : Boot ROM0

Last reset cause (0x00000002): LocalSoft
IR8340-K9 platform with 8388608 Kbytes of main memory

IR8340 rommon 9 >
```

Environment variable settings

Environment variable settings are configuration parameters that:

- are entered in capital letters followed by a definition,
- control system boot behavior and hardware configurations, and
- are configured in ROMmon mode.

The following example shows the environmental variables that can be configured in ROMmon mode:

```
rommon 1 > confreg 0x0
rommon 1> BOOT_WDOG = DISABLE
rommon 1> BOOT = IR8340-K9_image_name
```

Save Environment Variable settings

Use this task to save environment variable settings to ensure that your current configuration is preserved and will not be lost when the system is reset or booted.

Environment variables that are not explicitly saved are temporary and will be discarded whenever the system undergoes a reset or boot cycle.

Before you begin

The router must be in ROMmon mode to perform this task.

Procedure

Use the **sync** command to save the current environment variable settings.

Example:

```
rommon > sync
```

Environmental values that are not saved with the **sync** command are discarded whenever the system is reset or booted.

The current environment variable settings are now saved and will persist through system resets and boots.

Exit ROM Monitor Mode

Use this this task to return the router to normal operational mode from the ROMmon prompt.

To exit ROM Monitor mode, you must change the configuration register to a valid boot value and then reset the router.

Before you begin

The router must be in ROMmon mode (indicated by the `rommon >` prompt).

Procedure

Step 1 Use the **confreg** command to initiate the configuration register configuration prompts.

Example:

```
rommon 1> confreg
```

Respond to each prompt as instructed by the system to set the desired boot behavior.

Step 2 Use the **reset** command to reset and initialize the router.

Example:

```
rommon 2> reset
```

The router will reboot and attempt to load the Cisco IOS software image based on the new configuration register settings.

Configuration example

This example demonstrates the interactive configuration process using the confreg command in ROM Monitor mode, showing the configuration summary and available options for modifying system boot characteristics.

```
rommon 3 > confreg
           Configuration Summary
   (Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:  y
 enable  "diagnostic mode"? y/n  [n]:
 enable  "use net in IP bcast address"? y/n  [n]:
 enable  "load rom after netboot fails"? y/n  [n]:
```

```
 enable  "use all zero broadcast"? y/n  [n]:
 disable "break/abort has effect"? y/n  [n]:
 enable  "ignore system config info"? y/n  [n]:
 change console baud rate? y/n  [n]:
change the boot characteristics? y/n  [n]:
          Configuration Summary
   (Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:
```

Upgrade the ROMmon for a Router

Use this task to ensure the latest ROMmon version is installed and operational on the IR8340-K9 router.

ROMmon upgrades on the IR8340-K9 router are performed automatically when the image is booted because the latest ROMmon version is bundled with the Cisco IOS XE image. An internal algorithm detects if the current running version is older than the bundled version; if it is, the system upgrades it automatically. If the versions are equal, no upgrade is executed. After a successful upgrade, the router reboots automatically to load and execute the new version.

Before you begin

Ensure the desired Cisco IOS XE image is available on the router's storage.

Procedure

Step 1 (Optional) Use the **show rom-monitor** *slot* command to view the current ROMmon release numbers on the hardware.

Example:

```
Router# show rom-monitor r0
```

Step 2 If autoboot is not enabled, use the **boot** *filesystem:/file-location* command at the ROMmon prompt to boot the Cisco IOS XE image.

Example:

```
rommon 1> boot flash:ir8340-universalk9.bin
```

The ROMmon upgrade is not permanent until the Cisco IOS XE image is successfully booted. If the configuration register is not set to `0x2102`, manual booting is required.

Step 3 After the boot is complete, use the **enable** command to enter privileged EXEC mode.

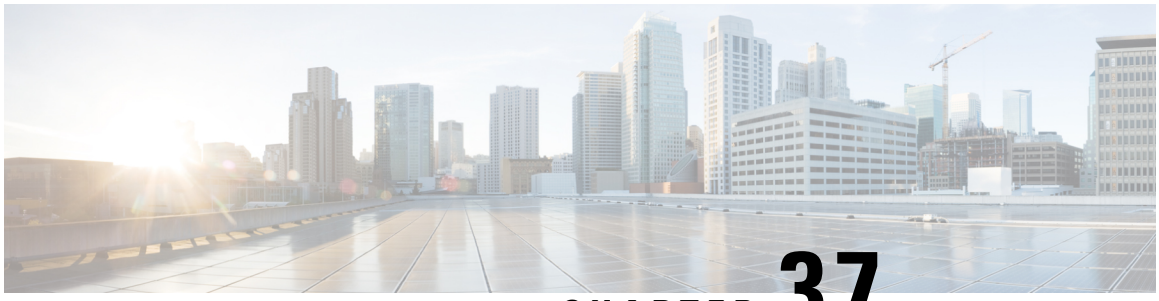
Example:

```
Router> enable
```

Step 4 Use the **show rom-monitor** *slot* command to verify that the ROMmon has been upgraded to the expected version.

Example:

```
Router# show rom-monitor r0
```



CHAPTER 37

Process Health Monitoring

This chapter describes how to manage and monitor the health of various components of your router. It contains the following sections:

- Monitoring Control Plane Resources, on page 437
- Monitoring Hardware Using Alarms, on page 449

Monitoring Control Plane Resources

The following sections explain the details of memory and CPU monitoring from the perspective of the Cisco IOS process and the overall control plane:

- Avoiding Problems Through Regular Monitoring, on page 437
- Cisco IOS Process Resources, on page 437
- Overall Control Plane Resources, on page 447

Avoiding Problems Through Regular Monitoring

Processes should provide monitoring and notification of their status/health to ensure correct operation. When a process fails, a syslog error message is displayed and either the process is restarted or the router is rebooted. A syslog error message is displayed when a monitor detects that a process is stuck or has crashed. If the process can be restarted, it is restarted; else, the router is restarted.

Monitoring system resources enables you to detect potential problems before they occur, thus avoiding outages. It also establishes a baseline for a normal system load. You can use this information as a basis for comparison, when you upgrade hardware or software to see if the upgrade has affected resource usage.

Cisco IOS Process Resources

You can view CPU utilization statistics on active processes and see the amount of memory being used in these processes using the **show memory** command and the **show process cpu** command. These commands provide a representation of memory and CPU utilization from the perspective of only the Cisco IOS process; they do not include information for resources on the entire platform. When the **show memory** command is used in a system with 4 GB RAM running a single Cisco IOS process, the following memory usage is displayed:

```
Router# show memory
Tracekey : 1#ac03ca5b9748c51baa2971d7a417e55e

Head Total(b) Used(b) Free(b) Lowest(b) Largest(b)
Processor 7F74107C4048 3807080396 239658308 3567422088 661095820 3145727908
reserve P 7F74107C40A0 102404 92 102312 102312 102312
lsmpi_io 7F740259B1A8 6295128 6294304 824 824 412
Dynamic heap limit(MB) 3000 Use(MB) 0



Processor memory

Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F74107C4048 0000102408 00000000 7F74107DD0A8 001 -------- -------- *Init*
:559FB88BD000+5E73061
7F74107DD0A8 0000000056 7F74107C4048 7F74107DD138 001 -------- -------- *Init*
:559FB88BD000+5E7307F
7F74107DD138 0000008224 7F74107DD0A8 7F74107DF1B0 001 -------- -------- *Init*
:559FB88BD000+5E7309A
7F74107DF1B0 0000000296 7F74107DD138 7F74107DF330 001 -------- -------- *Init*
:559FB88BD000+9093511
7F74107DF330 0000000568 7F74107DF1B0 7F74107DF5C0 001 -------- -------- *Init*
:559FB88BD000+9099C4A
7F74107DF5C0 0000032776 7F74107DF330 7F74107E7620 001 -------- -------- Managed Chunk Q
:559FB88BD000+90832BF
7F74107E7620 0000000056 7F74107DF5C0 7F74107E76B0 001 -------- -------- *Init*
:559FB88BD000+5EC98DD
7F74107E76B0 0000032776 7F74107E7620 7F74107EF710 001 -------- -------- Queue Pair - Q
:559FB88BD000+90B02EF
7F74107EF710 0000012808 7F74107E76B0 7F74107F2970 001 -------- -------- *Init*
:559FB88BD000+13BF988B
7F74107F2970 0000032776 7F74107EF710 7F74107FA9D0 001 -------- -------- List Elements
:559FB88BD000+904DB29
7F74107FA9D0 0000032776 7F74107F2970 7F7410802A30 001 -------- -------- List Headers
:559FB88BD000+904DB77
7F7410802A30 0000032776 7F74107FA9D0 7F741080AA90 001 -------- -------- IOSXE Process S
:559FB88BD000+A664348
7F741080AA90 0000032776 7F7410802A30 7F7410812AF0 001 -------- -------- IOSXE Queue Pro
:559FB88BD000+A66438A
7F7410812AF0 0000065544 7F741080AA90 7F7410822B50 001 -------- -------- IOSXE Queue Bal
:559FB88BD000+A6643C5
7F7410822B50 0000000328 7F7410812AF0 7F7410822CF0 001 -------- -------- *Init*
:559FB88BD000+13BF1567
7F7410822CF0 0000000328 7F7410822B50 7F7410822E90 001 -------- -------- *Init*
:559FB88BD000+13BF1567
7F7410822E90 0000000192 7F7410822CF0 7F7410822FA8 001 -------- -------- *Init*
:559FB88BD000+90C3F74
7F7410822FA8 0000036872 7F7410822E90 7F741082C008 001 -------- -------- *Init*
:559FB88BD000+A65A3A8
7F741082C008 0000010008 7F7410822FA8 7F741082E778 001 -------- -------- Platform VM Pag
:559FB88BD000+A68A8A3
7F741082E778 0000002008 7F741082C008 7F741082EFA8 001 -------- -------- *Init*
iosd_crb_crankshaft_unix:7F7460E22000+7D8CB
7F741082EFA8 0000200712 7F741082E778 7F7410860008 001 -------- -------- Interrupt Stack
:559FB88BD000+A65A3A8
7F7410860008 0000000328 7F741082EFA8 7F74108601A8 001 -------- -------- *Init*
:559FB88BD000+13BF1567
7F74108601A8 0000003008 7F7410860008 7F7410860DC0 001 -------- -------- Watched Semapho
:559FB88BD000+90BBCE8
7F7410860DC0 0000000400 7F74108601A8 7F7410860FA8 001 -------- -------- *Init*
:559FB88BD000+13BF1567
7F7410860FA8 0000036872 7F7410860DC0 7F741086A008 001 -------- -------- *Init*
```

```
:559FB88BD000+A65A3A8
7F741086A008 0000000328 7F7410860FA8 7F741086A1A8 001 -------- -------- *Init*
:559FB88BD000+13BF1567
7F741086A1A8 0000000184 7F741086A008 7F741086A2B8 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F741086A2B8 0000000184 7F741086A1A8 7F741086A3C8 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F741086A3C8 0000000184 7F741086A2B8 7F741086A4D8 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F741086A4D8 0000000184 7F741086A3C8 7F741086A5E8 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F741086A5E8 0000000184 7F741086A4D8 7F741086A6F8 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F741086A6F8 0000000096 7F741086A5E8 7F741086A7B0 001 -------- -------- *Init*
:559FB88BD000+90A4AA4
7F741086A7B0 0000000152 7F741086A6F8 7F741086A8A0 001 -------- -------- Crypto CA
:559FB88BD000+98E58BF
7F741086A8A0 0000000152 7F741086A7B0 7F741086A990 001 -------- -------- Crypto CA
:559FB88BD000+98E58BF
7F741086A990 0000000152 7F741086A8A0 7F741086AA80 001 -------- -------- Crypto CA
:559FB88BD000+98E58BF
7F741086AA80 0000000152 7F741086A990 7F741086AB70 001 -------- -------- Crypto CA
:559FB88BD000+98E58BF
7F741086AB70 0000000176 7F741086AA80 7F741086AC78 001 -------- -------- Crypto CA
:559FB88BD000+98E58BF
7F741086AC78 0000000272 7F741086AB70 7F741086ADE0 001 -------- -------- Crypto CA
:559FB88BD000+98E5875
7F741086ADE0 0000000096 7F741086AC78 7F741086AE98 000 7F741C306070 7F7415EC2CE8 (fragment)
 :559FB88BD000+98E5875
7F741086AE98 0000000184 7F741086ADE0 7F741086AFA8 001 -------- -------- Init
:559FB88BD000+52443C9
7F741086AFA8 0000036872 7F741086AE98 7F7410874008 001 -------- -------- *Init*
:559FB88BD000+A65A3A8
7F7410874008 0000000328 7F741086AFA8 7F74108741A8 001 -------- -------- *Init*
:559FB88BD000+13BF1567
7F74108741A8 0000000184 7F7410874008 7F74108742B8 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F74108742B8 0000000064 7F74108741A8 7F7410874350 001 -------- -------- Parser Linkage
:559FB88BD000+5D90EDA
7F7410874350 0000000216 7F74108742B8 7F7410874480 001 -------- -------- IPv4 FIB subblo
:559FB88BD000+5A2947A
7F7410874480 0000000224 7F7410874350 7F74108745B8 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F74108745B8 0000000224 7F7410874480 7F74108746F0 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F74108746F0 0000000328 7F74108745B8 7F7410874890 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F7410874890 0000000328 7F74108746F0 7F7410874A30 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F7410874A30 0000000328 7F7410874890 7F7410874BD0 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F7410874BD0 0000000896 7F7410874A30 7F7410874FA8 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F7410874FA8 0000200712 7F7410874BD0 7F74108A6008 001 -------- -------- Interrupt Stack
:559FB88BD000+A65A3A8
7F74108A6008 0000000968 7F7410874FA8 7F74108A6428 001 -------- -------- *Init*
iosd_crb_crankshaft_unix:7F7460E22000+3AC76
7F74108A6428 0000002008 7F74108A6008 7F74108A6C58 001 -------- -------- Watcher Message
:559FB88BD000+90BBD7B
7F74108A6C58 0000000360 7F74108A6428 7F74108A6E18 001 -------- -------- Process Events
:559FB88BD000+90B6840
7F74108A6E18 0000000096 7F74108A6C58 7F74108A6ED0 001 -------- -------- SWIDB_SB_PTP
:559FB88BD000+5324CAA
7F74108A6ED0 0000000128 7F74108A6E18 7F74108A6FA8 001 -------- -------- *Init*
```

```
:559FB88BD000+9060A7C
7F74108A6FA8 0000036872 7F74108A6ED0 7F74108B0008 001 -------- -------- *Init*
:559FB88BD000+A65A3A8
7F74108B0008 0000002840 7F74108A6FA8 7F74108B0B78 001 -------- -------- *Init*
:559FB88BD000+B529B9D
7F74108B0B78 0000000984 7F74108B0008 7F74108B0FA8 001 -------- -------- Watched Message
:559FB88BD000+90BBD4A
7F74108B0FA8 0000200712 7F74108B0B78 7F74108E2008 001 -------- -------- Interrupt Stack
:559FB88BD000+A65A3A8
7F74108E2008 0000002336 7F74108B0FA8 7F74108E2980 001 -------- -------- Process Array
:559FB88BD000+90C3DC0
7F74108E2980 0000000360 7F74108E2008 7F74108E2B40 001 -------- -------- Process Events
:559FB88BD000+90B6840
7F74108E2B40 0000000328 7F74108E2980 7F74108E2CE0 001 -------- -------- *Init*
:559FB88BD000+13BF1567
7F74108E2CE0 0000000120 7F74108E2B40 7F74108E2DB0 001 -------- -------- *Init*
:559FB88BD000+90A4347
7F74108E2DB0 0000000184 7F74108E2CE0 7F74108E2EC0 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F74108E2EC0 0000000144 7F74108E2DB0 7F74108E2FA8 001 -------- -------- *Init*
:559FB88BD000+4FFE35D
7F74108E2FA8 0000036872 7F74108E2EC0 7F74108EC008 001 -------- -------- *Init*
:559FB88BD000+A65A3A8
7F74108EC008 0000001232 7F74108E2FA8 7F74108EC530 001 -------- -------- Process
:559FB88BD000+90C3F05
7F74108EC530 0000001232 7F74108EC008 7F74108ECA58 001 -------- -------- Process
:559FB88BD000+90C3F05
7F74108ECA58 0000000096 7F74108EC530 7F74108ECB10 001 -------- -------- Init
:559FB88BD000+8FF0694
7F74108ECB10 0000000184 7F74108ECA58 7F74108ECC20 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F74108ECC20 0000000184 7F74108ECB10 7F74108ECD30 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F74108ECD30 0000000184 7F74108ECC20 7F74108ECE40 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F74108ECE40 0000000272 7F74108ECD30 7F74108ECFA8 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F74108ECFA8 0000200712 7F74108ECE40 7F741091E008 001 -------- -------- Interrupt Stack
:559FB88BD000+A65A3A8
7F741091E008 0000000184 7F74108ECFA8 7F741091E118 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F741091E118 0000003008 7F741091E008 7F741091ED30 001 -------- -------- Reg Function Li
:559FB88BD000+905FC68
7F741091ED30 0000000064 7F741091E118 7F741091EDC8 001 -------- -------- Parser Linkage
:559FB88BD000+5D90C9A
7F741091EDC8 0000000064 7F741091ED30 7F741091EE60 001 -------- -------- Parser Linkage
:559FB88BD000+5D90EDA
7F741091EE60 0000000064 7F741091EDC8 7F741091EEF8 001 -------- -------- Parser Linkage
:559FB88BD000+5D90C9A
7F741091EEF8 0000000088 7F741091EE60 7F741091EFA8 001 -------- -------- *Init*
:559FB88BD000+5EC98DD
7F741091EFA8 0000036872 7F741091EEF8 7F7410928008 001 -------- -------- *Init*
:559FB88BD000+A65A3A8
7F7410928008 0000000184 7F741091EFA8 7F7410928118 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F7410928118 0000001504 7F7410928008 7F7410928750 001 -------- -------- Reg Function Se
:559FB88BD000+905FCC2
7F7410928750 0000001504 7F7410928118 7F7410928D88 001 -------- -------- Reg Function Ca
:559FB88BD000+905FCEF
7F7410928D88 0000000064 7F7410928750 7F7410928E20 001 -------- -------- Parser Linkage
:559FB88BD000+5D90C9A
7F7410928E20 0000000064 7F7410928D88 7F7410928EB8 001 -------- -------- Parser Linkage
:559FB88BD000+5D90EDA
7F7410928EB8 0000000152 7F7410928E20 7F7410928FA8 001 -------- -------- Init
```

```
:559FB88BD000+500508B
7F7410928FA8 0000200712 7F7410928EB8 7F741095A008 001 -------- -------- Interrupt Stack
:559FB88BD000+A65A3A8
7F741095A008 0000006888 7F7410928FA8 7F741095BB48 001 -------- -------- TTY data
:559FB88BD000+8F75806
7F741095BB48 0000004104 7F741095A008 7F741095CBA8 001 -------- -------- TTY Input Buf
:559FB88BD000+8F77B6F
7F741095CBA8 0000004104 7F741095BB48 7F741095DC08 001 -------- -------- TTY Output Buf
:559FB88BD000+8F77BC7
7F741095DC08 0000024584 7F741095CBA8 7F7410963C68 001 -------- -------- proc_hist_lmt_v
:559FB88BD000+D06DC00
7F7410963C68 0000008200 7F741095DC08 7F7410965CC8 001 -------- -------- proc_hist_lmt_v
:559FB88BD000+D06DC48
7F7410965CC8 0000008200 7F7410963C68 7F7410967D28 001 -------- -------- proc_hist_lmt_v
:559FB88BD000+D06DC82
7F7410967D28 0000005008 7F7410965CC8 7F7410969110 001 -------- -------- messages
:559FB88BD000+90BBBB5
7F7410969110 0000005008 7F7410967D28 7F741096A4F8 001 -------- -------- Watched message
:559FB88BD000+90BBBED
7F741096A4F8 0000020008 7F7410969110 7F741096F378 001 -------- -------- Watched Queue
:559FB88BD000+90BBC1E
7F741096F378 0000065544 7F741096A4F8 7F741097F3D8 001 -------- -------- Watched Queue I
:559FB88BD000+90BBC55
7F741097F3D8 0000020008 7F741096F378 7F7410984258 001 -------- -------- Watched Boolean
:559FB88BD000+90BBC86
7F7410984258 0000020008 7F741097F3D8 7F74109890D8 001 -------- -------- Watched Bitfiel
:559FB88BD000+90BBCB7
7F74109890D8 0000010008 7F7410984258 7F741098B848 001 -------- -------- Watcher Info
:559FB88BD000+90BBD19
7F741098B848 0000010008 7F74109890D8 7F741098DFB8 001 -------- -------- Read/Write Lock
:559FB88BD000+90BBDAC
7F741098DFB8 0000000184 7F741098B848 7F741098E0C8 001 -------- -------- *Init*
:559FB88BD000+90A4B10
7F741098E0C8 0000000064 7F741098DFB8 7F741098E160 001 -------- -------- Init
:559FB88BD000+98AD5C3
7F741098E160 0000000576 7F741098E0C8 7F741098E3F8 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F741098E3F8 0000000400 7F741098E160 7F741098E5E0 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F741098E5E0 0000001240 7F741098E3F8 7F741098EB10 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F741098EB10 0000000488 7F741098E5E0 7F741098ED50 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F741098ED50 0000000072 7F741098EB10 7F741098EDF0 001 -------- -------- Init
:559FB88BD000+98DB044
7F741098EDF0 0000000056 7F741098ED50 7F741098EE80 001 -------- -------- Init
:559FB88BD000+5D94FF6
7F741098EE80 0000000208 7F741098EDF0 7F741098EFA8 001 -------- -------- *Init*
:559FB88BD000+C1E0292
7F741098EFA8 0000028104 7F741098EE80 7F7410995DC8 001 -------- -------- Process Stack
:559FB88BD000+A65A3A8
7F7410995DC8 0000000152 7F741098EFA8 7F7410995EB8 001 -------- -------- *Init*
:559FB88BD000+90C3F74
7F7410995EB8 0000000152 7F7410995DC8 7F7410995FA8 001 -------- -------- *Init*
:559FB88BD000+9060A7C
7F7410995FA8 0000016104 7F7410995EB8 7F7410999EE8 001 -------- -------- Process Stack
:559FB88BD000+A65A3A8
7F7410999EE8 0000032776 7F7410995FA8 7F74109A1F48 001 -------- -------- List Elements
:559FB88BD000+904DF98
7F74109A1F48 0000032776 7F7410999EE8 7F74109A9FA8 001 -------- -------- List Elements
:559FB88BD000+904DF98
7F74109A9FA8 0000032776 7F74109A1F48 7F74109B2008 001 -------- -------- List Elements
:559FB88BD000+904DF98
7F74109B2008 0000032776 7F74109A9FA8 7F74109BA068 001 -------- -------- List Elements
```

```
:559FB88BD000+904DF98
7F74109BA068 0000032776 7F74109B2008 7F74109C20C8 001 -------- -------- List Elements
:559FB88BD000+904DF98
7F74109C20C8 0000032776 7F74109BA068 7F74109CA128 001 -------- -------- List Elements
:559FB88BD000+904DF98
7F74109CA128 0000032776 7F74109C20C8 7F74109D2188 001 -------- -------- List Element
```

The **show process cpu** command displays Cisco IOS CPU utilization average:

```
Router# show process cpu
CPU utilization for five seconds: 1%/0%; one minute: 1%; five minutes: 1%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
1 3 21 142 0.00% 0.00% 0.00% 0 Chunk Manager
2 166 399 416 0.00% 0.00% 0.00% 0 Load Meter
3 0 1 0 0.00% 0.00% 0.00% 0 PKI Trustpool
4 0 1 0 0.00% 0.00% 0.00% 0 Retransmission o
5 0 1 0 0.00% 0.00% 0.00% 0 IPC ISSU Dispatc
6 17 14 1214 0.00% 0.00% 0.00% 0 RF Slave Main Th
7 0 1 0 0.00% 0.00% 0.00% 0 EDDRI_MAIN
8 0 1 0 0.00% 0.00% 0.00% 0 RO Notify Timers
9 1017 304 3345 0.47% 0.06% 0.03% 0 Check heaps
10 5 34 147 0.00% 0.00% 0.00% 0 Pool Manager
11 0 1 0 0.00% 0.00% 0.00% 0 DiscardQ Backgro
12 1 2 500 0.00% 0.00% 0.00% 0 Timers
13 0 53 0 0.00% 0.00% 0.00% 0 WATCH_AFS
14 0 1 0 0.00% 0.00% 0.00% 0 MEMLEAK PROCESS
15 4 18 222 0.00% 0.00% 0.00% 0 ARP Input
16 37 2132 17 0.00% 0.00% 0.00% 0 ARP Background
17 0 2 0 0.00% 0.00% 0.00% 0 ATM Idle Timer
18 0 1 0 0.00% 0.00% 0.00% 0 ATM ASYNC PROC
19 0 1 0 0.00% 0.00% 0.00% 0 CEF MIB API
20 0 1 0 0.00% 0.00% 0.00% 0 AAA_SERVER_DEADT
21 0 1 0 0.00% 0.00% 0.00% 0 Policy Manager
22 0 2 0 0.00% 0.00% 0.00% 0 DDR Timers
23 130 50 2600 0.00% 0.00% 0.00% 0 Entity MIB API
24 179 83 2156 0.00% 0.00% 0.00% 0 PrstVbl
25 2 247 8 0.00% 0.00% 0.00% 0 Serial Backgroun
26 0 1 0 0.00% 0.00% 0.00% 0 RMI RM Notify Wa
27 0 2 0 0.00% 0.00% 0.00% 0 ATM AutoVC Perio
28 0 2 0 0.00% 0.00% 0.00% 0 ATM VC Auto Crea
29 10 1000 10 0.00% 0.00% 0.00% 0 IOSXE heartbeat
30 31 1023 30 0.00% 0.00% 0.00% 0 DB Lock Manager
31 33 1989 16 0.00% 0.00% 0.00% 0 GraphIt
32 0 1 0 0.00% 0.00% 0.00% 0 DB Notification
33 0 1 0 0.00% 0.00% 0.00% 0 IPC Apps Task
34 0 1 0 0.00% 0.00% 0.00% 0 ifIndex Receive
35 5 401 12 0.00% 0.00% 0.00% 0 IPC Event Notifi
36 32 1952 16 0.00% 0.00% 0.00% 0 IPC Mcast Pendin
37 0 1 0 0.00% 0.00% 0.00% 0 Platform appsess
38 0 34 0 0.00% 0.00% 0.00% 0 IPC Dynamic Cach
39 1 401 2 0.00% 0.00% 0.00% 0 IPC Service NonC
40 0 1 0 0.00% 0.00% 0.00% 0 IPC Zone Manager
41 16 1952 8 0.00% 0.00% 0.00% 0 IPC Periodic Tim
42 19 1952 9 0.00% 0.00% 0.00% 0 IPC Deferred Por
43 0 1 0 0.00% 0.00% 0.00% 0 IPC Process leve
44 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat Manager
45 0 115 0 0.00% 0.00% 0.00% 0 IPC Check Queue
46 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat RX Cont
47 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat TX Cont
48 10 201 49 0.00% 0.00% 0.00% 0 IPC Keep Alive M
49 33 401 82 0.00% 0.00% 0.00% 0 IPC Loadometer
50 0 1 0 0.00% 0.00% 0.00% 0 IPC Session Deta
51 0 1 0 0.00% 0.00% 0.00% 0 SENSOR-MGR event
52 2 201 9 0.00% 0.00% 0.00% 0 Compute SRP rate
```

```
53 0 1 0 0.00% 0.00% 0.00% 0 IFS Agent Manage
54 0 1 0 0.00% 0.00% 0.00% 0 License IPC stat
55 0 1 0 0.00% 0.00% 0.00% 0 License IPC serv
56 0 7 0 0.00% 0.00% 0.00% 0 Net Input
57 0 2 0 0.00% 0.00% 0.00% 0 Dialer event
58 0 1 0 0.00% 0.00% 0.00% 0 SERIAL A'detect
59 0 1 0 0.00% 0.00% 0.00% 0 IOSXE signals IO
60 0 1 0 0.00% 0.00% 0.00% 0 client_entity_se
61 1 1 1000 0.00% 0.00% 0.00% 0 RF SCTPthread
62 0 1 0 0.00% 0.00% 0.00% 0 CHKPT RG SCTPthr
63 0 2 0 0.00% 0.00% 0.00% 0 XML Proxy Client
64 0 1 0 0.00% 0.00% 0.00% 0 ARP Snoop
65 28 1999 14 0.00% 0.00% 0.00% 0 Dynamic ARP Insp
66 1216 110 11054 0.00% 0.00% 0.00% 0 crypto sw pk pro
67 0 2 0 0.00% 0.00% 0.00% 0 License Client N
68 0 1 0 0.00% 0.00% 0.00% 0 Image License br
69 24 133 180 0.00% 0.00% 0.00% 0 SAEventLog
70 11 2 5500 0.00% 0.00% 0.00% 0 SAStorage
71 1 4 250 0.00% 0.00% 0.00% 0 SASConnect
72 41 1165 35 0.00% 0.00% 0.00% 0 SASRcvWQ
73 2 5 400 0.00% 0.00% 0.00% 0 SACConnect
74 30 1183 25 0.00% 0.00% 0.00% 0 SACRcvWQ
75 8 68 117 0.00% 0.00% 0.00% 0 Licensing Auto U
76 0 1 0 0.00% 0.00% 0.00% 0 License HA Consi
77 0 1 0 0.00% 0.00% 0.00% 0 Token Daemon
78 0 1 0 0.00% 0.00% 0.00% 0 Critical Bkgnd
79 221 1374 160 0.00% 0.00% 0.00% 0 Net Background
80 0 3 0 0.00% 0.00% 0.00% 0 IDB Work
81 8 101 79 0.00% 0.00% 0.00% 0 Logger
82 71 1985 35 0.00% 0.00% 0.00% 0 TTY Background
83 53 24 2208 0.00% 0.00% 0.00% 0 CTS CORE
84 1 3 333 0.00% 0.00% 0.00% 0 SXP CORE
85 56 1164 48 0.00% 0.00% 0.00% 0 SASRcvWQWrk1
86 3 14 214 0.00% 0.00% 0.00% 0 IF-MGR control p
87 0 28 0 0.00% 0.00% 0.00% 0 IF-MGR event pro
88 0 2 0 0.00% 0.00% 0.00% 0 CTS HA
89 0 2 0 0.00% 0.00% 0.00% 0 CTS HA IPC flow
90 0 1 0 0.00% 0.00% 0.00% 0 CTS HA operation
91 5357 18052 296 0.15% 0.20% 0.20% 0 IOSD ipc task
92 213 5786 36 0.00% 0.00% 0.00% 0 IOSD chasfs task
93 0 2 0 0.00% 0.00% 0.00% 0 Crimson interfac
94 274 19 14421 0.00% 0.00% 0.00% 0 Crimson Database
95 681 2548 267 0.00% 0.02% 0.00% 0 Crimson flush tr
96 4 287 13 0.00% 0.00% 0.00% 0 REDUNDANCY FSM
97 0 1 0 0.00% 0.00% 0.00% 0 Punt FP Stats Du
98 171 953 179 0.00% 0.00% 0.00% 0 PuntInject Keepa
99 0 16 0 0.00% 0.00% 0.00% 0 ESG MATM Learnin
100 141 401 351 0.00% 0.00% 0.00% 0 CMAN RP Msg Proc
101 118 1998 59 0.00% 0.00% 0.00% 0 Environmental Mo
102 42 1998 21 0.00% 0.00% 0.00% 0 RP HA Periodic
103 0 2 0 0.00% 0.00% 0.00% 0 cpf_msg_holdq_pr
104 0 1 0 0.00% 0.00% 0.00% 0 cpf_msg_rcvq_pro
105 0 1 0 0.00% 0.00% 0.00% 0 cpf_process_tpQ
106 0 1 0 0.00% 0.00% 0.00% 0 CEF RRP RF waite
107 0 1 0 0.00% 0.00% 0.00% 0 CONSOLE helper p
108 175 50 3500 0.00% 0.00% 0.00% 0 DBAL EVENTS
109 0 1 0 0.00% 0.00% 0.00% 0 XDR RRP RF waite
110 50 1999 25 0.00% 0.00% 0.00% 0 REDUNDANCY peer
111 337 19959 16 0.00% 0.00% 0.00% 0 100ms check
112 0 1 0 0.00% 0.00% 0.00% 0 CWAN APS HA Proc
113 1 34 29 0.00% 0.00% 0.00% 0 RF CWAN HA Proce
114 0 1 0 0.00% 0.00% 0.00% 0 CWAN IF EVENT HA
115 0 4 0 0.00% 0.00% 0.00% 0 ANCP HA
116 0 2 0 0.00% 0.00% 0.00% 0 ANCP HA IPC flow
```

```
117 0 1 0 0.00% 0.00% 0.00% 0 QoS HA ID RETAIN
118 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
119 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
120 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
121 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
122 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
123 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
124 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
125 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
126 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
127 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
128 0 1 0 0.00% 0.00% 0.00% 0 CHKPT Test clien
129 0 1 0 0.00% 0.00% 0.00% 0 DHCPC HA
130 0 1 0 0.00% 0.00% 0.00% 0 DHCPD HA
131 0 1 0 0.00% 0.00% 0.00% 0 DHCPv6 Relay HA
132 0 1 0 0.00% 0.00% 0.00% 0 DHCPv6 Server HA
133 0 2 0 0.00% 0.00% 0.00% 0 SISF HA Process
134 0 5 0 0.00% 0.00% 0.00% 0 ARP HA
135 0 1 0 0.00% 0.00% 0.00% 0 IOSXE-RP Punt Se
136 0 1 0 0.00% 0.00% 0.00% 0 IOSXE-RP Punt IP
137 0 1 0 0.00% 0.00% 0.00% 0 ACL Log Punt Ser
138 0 1 0 0.00% 0.00% 0.00% 0 ACL deny punt se
139 1576 124342 12 0.07% 0.06% 0.07% 0 L2 LISP Punt Pro
140 0 1 0 0.00% 0.00% 0.00% 0 OFSDN Punject Pr
141 1653 124341 13 0.15% 0.08% 0.07% 0 SIS Punt Process
142 0 1 0 0.00% 0.00% 0.00% 0 IOSXE-RP SPA TSM
143 0 1 0 0.00% 0.00% 0.00% 0 IOSXE-RP QFP HA
144 0 1 0 0.00% 0.00% 0.00% 0 Network-rf Notif
145 0 1 0 0.00% 0.00% 0.00% 0 DHCP Snooping cl
146 1 18 55 0.00% 0.00% 0.00% 0 DHCP Snooping
147 0 1 0 0.00% 0.00% 0.00% 0 DHCP Snooping db
148 0 1 0 0.00% 0.00% 0.00% 0 IKE HA Mgr
149 0 1 0 0.00% 0.00% 0.00% 0 IPSEC HA Mgr
150 0 1 0 0.00% 0.00% 0.00% 0 Crypto PKI-HA
151 3 12 250 0.00% 0.00% 0.00% 0 RF Master Main T
152 0 7 0 0.00% 0.00% 0.00% 0 RF Master Status
153 39 1009 38 0.00% 0.00% 0.00% 0 SACRcvWQWrk1
154 28 1009 27 0.00% 0.00% 0.00% 0 SACRcvWQWrk2
155 103 1185 86 0.00% 0.00% 0.00% 0 SACRcvWQWrk3
156 0 1 0 0.00% 0.00% 0.00% 0 BACK CHECK
157 323 210 1538 0.00% 0.00% 0.00% 0 SAMsgThread
158 25 202 123 0.00% 0.00% 0.00% 0 Compute load avg
159 515 107 4813 0.00% 0.02% 0.00% 0 Per-minute Jobs
160 167 2009 83 0.00% 0.00% 0.00% 0 Per-Second Jobs
161 0 1 0 0.00% 0.00% 0.00% 0 Transport Port A
162 167 9 18555 0.00% 0.00% 0.00% 0 ACT2 Crypto Engi
163 0 1 0 0.00% 0.00% 0.00% 0 AggMgr Process
164 0 4 0 0.00% 0.00% 0.00% 0 EEM ED MAT
165 4 73 54 0.00% 0.00% 0.00% 0 EEM ED ND
166 0 1 0 0.00% 0.00% 0.00% 0 MACSEC POST rest
167 0 1 0 0.00% 0.00% 0.00% 0 MACSEC POST hand
168 0 1 0 0.00% 0.00% 0.00% 0 MACSEC POST hand
169 0 1 0 0.00% 0.00% 0.00% 0 IOSXE-RP FastPat
170 0 6 0 0.00% 0.00% 0.00% 0 NGIO_BRI_POLL_DE
171 0 1 0 0.00% 0.00% 0.00% 0 ASYNC Input
172 21 2001 10 0.00% 0.00% 0.00% 0 IR8340 Alarm Con
173 0 2 0 0.00% 0.00% 0.00% 0 dialer isdn sess
174 0 1 0 0.00% 0.00% 0.00% 0 DSX3MIB ll handl
175 29 1961 14 0.00% 0.00% 0.00% 0 fanrp_l2fib
176 0 1 0 0.00% 0.00% 0.00% 0 POS APS Event Pr
177 0 2 0 0.00% 0.00% 0.00% 0 netclk_process
178 0 1 0 0.00% 0.00% 0.00% 0 netclk_ha_proces

Router#show process cpu platform sorted
```

```
CPU utilization for five seconds: 21%, one minute: 22%, five minutes: 22%
Core 0: CPU utilization for five seconds: 4%, one minute: 5%, five minutes: 5%
Core 1: CPU utilization for five seconds: 2%, one minute: 5%, five minutes: 5%
Core 2: CPU utilization for five seconds: 4%, one minute: 6%, five minutes: 6%
Core 3: CPU utilization for five seconds: 4%, one minute: 6%, five minutes: 6%
Core 4: CPU utilization for five seconds: 5%, one minute: 5%, five minutes: 5%
Core 5: CPU utilization for five seconds: 2%, one minute: 2%, five minutes: 2%
Core 6: CPU utilization for five seconds: 41%, one minute: 42%, five minutes: 43%
Core 7: CPU utilization for five seconds: 100%, one minute: 100%, five minutes: 100%
Pid PPid 5Sec 1Min 5Min Status Size Name
------------------------------------------------------------------------------
15435 15419 158% 158% 157% S 226748 ucode_pkt_PPE0
16998 16972 7% 6% 7% S 14252 btman
56 2 3% 5% 5% S 0 ksmd
15179 15139 2% 2% 2% S 171524 fman_fp_image
3821 3807 2% 3% 3% S 584524 linux_iosd-imag
26565 26536 1% 1% 1% S 52764 fman_cc
29044 2 0% 0% 0% I 0 kworker/u32:0-events
26536 16605 0% 0% 0% S 3788 pman
25912 25906 0% 0% 0% S 116208 fed main event
25906 16605 0% 0% 0% S 3796 pman
25689 25684 0% 0% 0% S 5036 nginx
25688 25684 0% 0% 0% S 5868 nginx
25684 25678 0% 0% 0% S 9980 nginx
25678 2784 0% 0% 0% S 3792 pman
25359 25349 0% 0% 0% S 8448 ngiolite
25349 16605 0% 0% 0% S 3792 pman
25048 25043 0% 0% 0% S 9876 ngiolite
25043 16605 0% 0% 0% S 3792 pman
24784 24779 0% 0% 0% S 12352 ngiolite
24779 16605 0% 0% 0% S 3824 pman
24713 2 0% 0% 0% I 0 kworker/1:2H
24688 24446 0% 0% 0% S 432 sleep
24446 1 0% 0% 0% S 2100 memory_monitor.
24341 24333 0% 0% 0% S 12300 ngiolite
24333 16605 0% 0% 0% S 3824 pman
24164 2 0% 0% 0% S 0 SarIosdMond
22339 2 0% 0% 0% I 0 gobisetpower-2-
22338 2 0% 0% 0% I 0 gobireadcb-2-2-
22337 2 0% 0% 0% I 0 gobiprobe-2-2-3
22139 22129 0% 0% 0% S 62172 iomd
22129 16605 0% 0% 0% S 3768 pman
21734 21723 0% 0% 0% S 62064 iomd
21723 16605 0% 0% 0% S 3768 pman
21566 2 0% 0% 0% I 0 kworker/1:3-gobiread
21466 21458 0% 0% 0% S 61992 iomd
21458 16605 0% 0% 0% S 3768 pman
21135 2 0% 0% 0% I 0 gobisetpower-1-
21131 2 0% 0% 0% I 0 gobireadcb-1-2-
21129 2 0% 0% 0% I 0 gobiprobe-1-2-2
21128 2 0% 0% 0% I 0 gobisetpower-0-
21127 2 0% 0% 0% I 0 gobireadcb-0-2-
21126 2 0% 0% 0% I 0 gobiprobe-0-2-2
21120 21113 0% 0% 0% S 74960 iomd
21113 16605 0% 0% 0% S 3820 pman
20788 20780 0% 0% 0% S 63312 iomd
20780 16605 0% 0% 0% S 3768 pman
19604 19582 0% 0% 0% S 18360 btman
19582 16605 0% 0% 0% S 3772 pman
19136 2 0% 0% 0% I 0 kworker/7:2H
18966 18953 0% 0% 0% S 23472 cmcc
18953 16605 0% 0% 0% S 3768 pman
18687 18674 0% 0% 0% S 12420 hman
18674 16605 0% 0% 0% S 3768 pman
```

```
18341 18320 0% 0% 0% S 7020 pttcd
18320 2784 0% 0% 0% S 3776 pman
18046 18031 0% 0% 0% S 78104 pubd
18031 2784 0% 0% 0% S 3776 pman
17396 2 0% 0% 0% I 0 kworker/5:0H
17301 16605 0% 0% 0% S 712 inotifywait
17057 1 0% 0% 0% S 3356 rotee
16972 14118 0% 0% 0% S 3768 pman
16605 1 0% 0% 0% S 8020 pvp.sh
16567 16545 0% 0% 0% S 19688 cman_fp
16545 14118 0% 0% 0% S 3768 pman
16331 16317 0% 0% 0% S 170672 cpp_cp_svr
16317 14118 0% 0% 0% S 3768 pman
16108 16095 0% 0% 0% S 66440 cpp_driver
16095 14118 0% 0% 0% S 3768 pman
15888 15873 0% 0% 0% S 70160 cpp_ha_top_leve
15873 14118 0% 0% 0% S 3772 pman
15660 15646 0% 0% 0% S 80592 cpp_sp_svr
15646 14118 0% 0% 0% S 3768 pman
15419 14118 0% 0% 0% S 3772 pman
15139 14118 0% 0% 0% S 3768 pman
14861 14844 0% 0% 0% S 12420 hman
14844 14118 0% 0% 0% S 3768 pman
14745 14731 0% 0% 0% S 1676 sort_files_by_i
14731 2784 0% 0% 0% S 3756 pman
14405 14118 0% 0% 0% S 712 inotifywait
14282 1 0% 0% 0% S 3356 rotee
14118 1 0% 0% 0% S 7988 pvp.sh
13169 13160 0% 0% 0% S 4224 flash_check.sh
13160 2784 0% 0% 0% S 3764 pman
12444 12435 0% 0% 0% S 14176 lman
12435 2784 0% 0% 0% S 3772 pman
12306 5095 0% 0% 0% S 3108 journalctl
10703 2784 0% 0% 0% S 712 inotifywait
10439 10345 0% 0% 0% S 712 inotifywait
10431 1 0% 0% 0% S 3360 rotee
10347 1 0% 0% 0% S 1180 xinetd
10346 1 0% 0% 0% S 1188 xinetd
10345 1 0% 0% 0% S 12168 rollback_timer.
10343 1 0% 0% 0% S 2068 auxinit.sh
9415 1 0% 0% 0% S 1012 xinetd
9412 1 0% 0% 0% S 1176 xinetd
9150 2 0% 0% 0% I 0 kworker/2:2H-kblockd
8448 1 0% 0% 0% S 6240 dhcpd
8198 8176 0% 0% 0% S 6784 tam_svcs_esg_cf
8176 2784 0% 0% 0% S 3776 pman
7929 7914 0% 0% 0% S 8432 tamd_proc
7914 2784 0% 0% 0% S 3776 pman
7673 7658 0% 0% 0% S 7904 tams_proc
7658 2784 0% 0% 0% S 3776 pman
7422 7405 0% 0% 0% S 32684 btman
7405 2784 0% 0% 0% S 3772 pman
7127 7106 0% 0% 0% S 26272 cli_agent
7106 2784 0% 0% 0% S 3780 pman
6878 6862 0% 1% 1% S 27788 cmand
6862 2784 0% 0% 0% S 3776 pman
6600 6583 0% 0% 0% S 92300 dbm
6583 2784 0% 0% 0% S 3772 pman
6059 6025 0% 0% 0% S 84992 fman_rp
6025 2784 0% 0% 0% S 3772 pman
5779 2 0% 0% 0% S 0 lfts_sar_aux
5778 5763 0% 0% 0% R 15868 hman
5763 2784 0% 0% 0% S 3772 pman
5568 5103 0% 0% 0% S 712 inotifywait
```

```
5560 1 0% 0% 0% S 3356 rotee
5412 5388 0% 0% 0% S 11452 keyman
5388 2784 0% 0% 0% S 3772 pman
5287 1 0% 0% 0% S 3360 rotee
5103 1 0% 0% 0% S 5640 iptbl.sh
5095 5086 0% 0% 0% S 10900 plogd
5086 2784 0% 0% 0% S 3780 pman
5054 13169 0% 0% 0% S 440 sleep
5043 4729 0% 0% 0% S 716 inotifywait
5030 14745 0% 0% 0% S 432 sleep
4940 1 0% 0% 0% S 3424 rotee

Router#
```

Overall Control Plane Resources

Control plane memory and CPU utilization on each control processor allows you to keep a tab on the overall control plane resources. You can use the **show platform software status control-processor brief** command (summary view) or the **show platform software status control-processor command** (detailed view) to view control plane memory and CPU utilization information.

All control processors should show status, Healthy. Other possible status values are Warning and Critical. Warning indicates that the router is operational, but that the operating level should be reviewed. Critical implies that the router is nearing failure.

If you see a Warning or Critical status, take the following actions:

- Reduce the static and dynamic loads on the system by reducing the number of elements in the configuration or by limiting the capacity for dynamic services.
- Reduce the number of routes and adjacencies, limit the number of ACLs and other rules, and so on.

The following sections describe the fields in the **show platform software status control-processor** command output.

Load Average

Load average represents the process queue or process contention for CPU resources. For example, on a single-core processor, an instantaneous load of 7 would mean that seven processes are ready to run, one of which is currently running. On a dual-core processor, a load of 7 would mean that seven processes are ready to run, two of which are currently running.

Memory Utilization

Memory utilization is represented by the following fields:

- Total—Total system memory
- Used—Consumed memory
- Free—Available memory
- Committed—Virtual memory committed to processes

CPU Utilization

CPU utilization is an indication of the percentage of time the CPU is busy, and is represented by the following fields:

- CPU—Allocated processor
- User—Non-Linux kernel processes
- System—Linux kernel process
- Nice—Low-priority processes
- Idle—Percentage of time the CPU was inactive
- IRQ—Interrupts
- SIRQ—System Interrupts
- IOwait—Percentage of time CPU was waiting for I/O

Example: show platform software status control-processor Command

The following are some examples of using the **show platform software status control-processor** command:

```
Router# show platform software status control-processor
RP0: online, statistics updated 2 seconds ago
Load Average: healthy
1-Min: 2.07, status: healthy, under 9.30
5-Min: 2.03, status: healthy, under 9.30
15-Min: 1.92, status: healthy, under 9.30
Memory (kb): healthy
Total: 8000724
Used: 2565652 (32%), status: healthy
Free: 5435072 (68%)
Committed: 3263176 (41%), under 90%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
User: 0.80, System: 4.00, Nice: 0.00, Idle: 95.19
IRQ: 0.00, SIRQ: 0.00, IOwait: 0.00
CPU1: CPU Utilization (percentage of time spent)
User: 2.50, System: 1.60, Nice: 0.00, Idle: 95.90
IRQ: 0.00, SIRQ: 0.00, IOwait: 0.00
CPU2: CPU Utilization (percentage of time spent)
User: 2.90, System: 1.80, Nice: 0.00, Idle: 95.30
IRQ: 0.00, SIRQ: 0.00, IOwait: 0.00
CPU3: CPU Utilization (percentage of time spent)
User: 1.49, System: 7.79, Nice: 0.00, Idle: 90.70
IRQ: 0.00, SIRQ: 0.00, IOwait: 0.00
CPU4: CPU Utilization (percentage of time spent)
User: 4.60, System: 1.80, Nice: 0.00, Idle: 93.59
IRQ: 0.00, SIRQ: 0.00, IOwait: 0.00
CPU5: CPU Utilization (percentage of time spent)
User: 0.99, System: 1.89, Nice: 0.00, Idle: 97.10
IRQ: 0.00, SIRQ: 0.00, IOwait: 0.00
CPU6: CPU Utilization (percentage of time spent)
User: 24.10, System: 18.30, Nice: 0.00, Idle: 57.60
IRQ: 0.00, SIRQ: 0.00, IOwait: 0.00
CPU7: CPU Utilization (percentage of time spent)
User: 92.50, System: 7.50, Nice: 0.00, Idle: 0.00
IRQ: 0.00, SIRQ: 0.00, IOwait: 0.00
```

```
Router#

Router# show platform software status control-processor brief
Load Average
Slot Status 1-Min 5-Min 15-Min
RP0 Healthy 2.01 2.01 1.91

Memory (kB)
Slot Status Total Used (Pct) Free (Pct) Committed (Pct)
RP0 Healthy 8000724 2565240 (32%) 5435484 (68%) 3263124 (41%)

CPU Utilization
Slot CPU User System Nice Idle IRQ SIRQ IOwait
RP0 0 0.70 3.30 0.00 96.00 0.00 0.00 0.00
1 4.39 7.39 0.00 88.11 0.00 0.09 0.00
2 4.80 9.30 0.00 85.80 0.00 0.10 0.00
3 3.39 8.69 0.00 87.91 0.00 0.00 0.00
4 4.40 1.80 0.00 93.80 0.00 0.00 0.00
5 1.00 1.90 0.00 97.10 0.00 0.00 0.00
6 24.64 23.54 0.00 51.80 0.00 0.00 0.00
7 92.60 7.40 0.00 0.00 0.00 0.00 0.00

Router#
```

Monitoring Hardware Using Alarms

Router Design and Monitoring Hardware

The router sends alarm notifications when problems are detected, allowing you to monitor the network remotely. You do not need to use **show** commands to poll devices on a routine basis; however, you can perform onsite monitoring if you choose.

BootFlash Disk Monitoring

The bootflash disk must have enough free space to store two core dumps. This condition is monitored, and if the bootflash disk is too small to store two core dumps, a syslog alarm is generated, as shown in the following example:

```
Oct  6 14:10:56.292: %FLASH_CHECK-3-DISK_QUOTA: R0/0: flash_check: Flash disk quota exceeded

[free space is 1429020 kB] - Please clean up files on bootflash.
```

Approaches for Monitoring Hardware Alarms

Viewing the Console or Syslog for Alarm Messages

The network administrator can monitor alarm messages by reviewing alarm messages sent to the system console or to a system message log (syslog).

Enabling the logging alarm Command

The **logging alarm** command must be enabled for the system to send alarm messages to a logging device, such as the console or a syslog. This command is not enabled by default.

You can specify the severity level of the alarms to be logged. All the alarms at and above the specified threshold generate alarm messages. For example, the following command sends only critical alarm messages to logging devices:

```
Router(config)# logging alarm critical
```

If alarm severity is not specified, alarm messages for all severity levels are sent to logging devices.

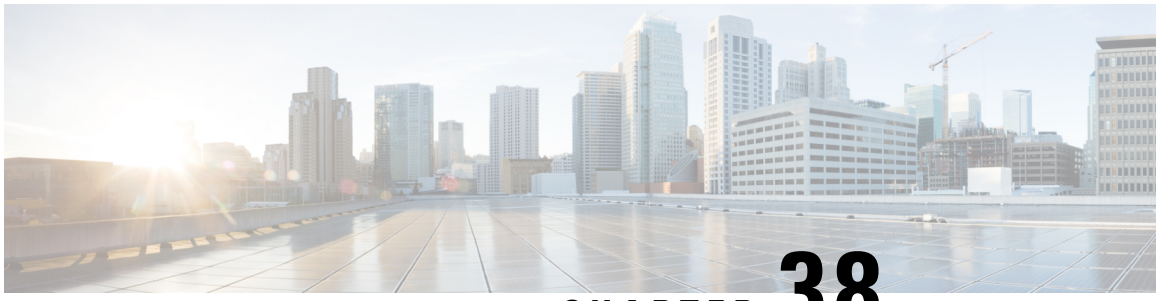
Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP

The SNMP is an application-layer protocol that provides a standardized framework and a common language used for monitoring and managing devices in a network.

SNMP provides notification of faults, alarms, and conditions that might affect services. It allows a network administrator to access router information through a network management system (NMS) instead of reviewing logs, polling devices, or reviewing log reports.

To use SNMP to get alarm notification, use the following MIBs:

- ENTITY-MIB, RFC4133 (required for the CISCO-ENTITY-ALARM-MIB, ENTITY-STATE-MIB and CISCO-ENTITY-SENSOR-MIB to work)
- CISCO-ENTITY-ALARM-MIB
- ENTITY-STATE-MIB
- CISCO-ENTITY-SENSOR-MIB (for transceiver environmental alarm information, which is not provided through the CISCO-ENTITY-ALARM-MIB)



CHAPTER 38

Device Sensors

A device sensor is a network feature that

- gathers raw endpoint data from network devices using protocols such as Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), Dynamic Host Configuration Protocol (DHCP), DHCP version 6, and multicast DNS (mDNS),
- makes the collected endpoint data available to registered clients, and
- operates in the context of an access session.

By aggregating endpoint information, device sensors enable network administrators to gain visibility into devices connected to the network, enhancing security and compliance.

Table 29: Feature History Table

Feature name	Release information	Feature description
Device Sensors	Release 26.1.1	You can use the device sensor network feature to gather raw endpoint data from network devices using protocols such as: • Cisco Discovery Protocol (CDP), • Link Layer Discovery Protocol (LLDP), • Dynamic Host Configuration Protocol (DHCP), • DHCP version 6, • Multicast DNS (mDNS). The collected endpoint data is made available to registered clients and operates within the context of an access session.

• Restrictions for device sensor configuration, on page 452

• Information about device sensors, on page 452
• Device sensor configuration, on page 454
• Configuration examples of device sensor, on page 461

Restrictions for device sensor configuration

These restrictions apply when configuring device sensors:

- Only Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), Dynamic Host Configuration Protocol (DHCP), Dynamic Host Configuration Protocol version 6 (DHCPv6), and multicast DNS (mDNS) protocols are supported.
- The session limit for profiling ports is 32.
- The length of a single Type-Length-Value (TLV) must not exceed 1024 bytes, and the combined length of TLVs from all protocols must not exceed 4096 bytes.
- The sensor profiles devices that are only one hop away.

Information about device sensors

Device sensors

A device sensor is an embedded network feature that

- collects raw endpoint data from network devices,
- provides detailed information to support device profiling, and
- enables secure client notifications and accounting for profiling events.

The profiling capability consists of two parts:

- Collector: Gathers endpoint data from network devices.
- Analyzer: Processes the data and determines the type of device.

Device sensor capabilities

Device sensors gather endpoint data using protocols such as Cisco Discovery Protocol, LLDP, DHCPv6, mDNS, and DHCP. This data helps determine the endpoint type as part of device profiling.

Device sensors support both internal clients and external clients:

- Internal clients: Embedded Device Classifier (local analyzer), ATM switch processor (ASP), MSI-Proxy, and EnergyWise (EW)
- External clients: Cisco ISE analyzers

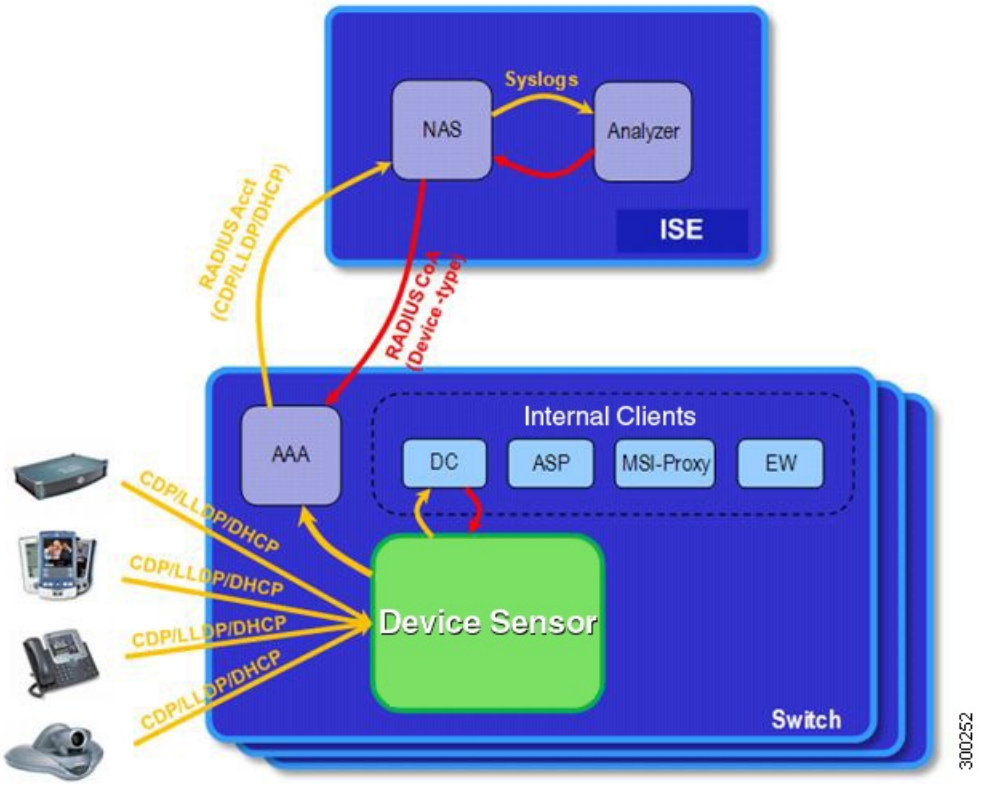
Client notifications and accounting messages containing profiling data along with the session events and other session-related data, such as the MAC address and the ingress port, are generated and sent to the internal and external clients (like Cisco ISE). By default, for each supported peer protocol, client notifications and accounting

events are only generated where an incoming packet includes a TLV that has not previously been received in the context of a given session. You can enable client notifications and accounting events for all TLV changes, where either a new TLV has been received or a previously received TLV has been received with a different value using CLI commands.

The device sensor's port security protects the switch from consuming memory and crashing during deliberate or unintentional denial-of-service (DoS) type attacks. The sensor limits the maximum device monitoring sessions to 32 per port. In case of lack of activity from hosts, the idle session time is 12 hours.

The device sensor represents the embedded collector functionality. The illustration shows a Cisco sensor in the context of the profiling system and also features other possible clients of the sensor.

Figure 11: Cisco sensor in a profiling system and clients of sensors



Device sensor profiling example

An access switch uses its device sensor to provide endpoint profiling data to Cisco ISE for authentication and authorization decisions.

Device sensor limitation example

A network endpoint without device sensor capability cannot provide detailed profiling data for security analysis.

Device sensor configuration

The device sensor is enabled by default.

The tasks in this chapter help you configure the sensor to meet specific requirements. If you do not perform these configuration tasks, these TLVs are included by default:

- Cisco Discovery Protocol filter: secondport-status-type and powernet-event-type (type 28 and 29).
- LLDP filter: organizationally-specific (type 127).
- DHCP filter: message-type (type 53).

Enable accounting augmentation

Perform this task to add device sensor protocol data to accounting records.

Before you begin

- The device must be in IBNS 2.0 mode before performing this task.
- For the sensor protocol data to be added to the accounting messages, you must enable session accounting by using the standard authentication, authorization, and accounting (AAA), and RADIUS configuration commands.

Follow these steps to enable accounting augmentation:

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode.

Example:

```
Router> enable
```

Enter your password if prompted.

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **device-sensor accounting** command to enable the addition of sensor protocol data to accounting records and also enable the generation of additional accounting events when new sensor data is detected.

Example:

```
Router(config)# device-sensor accounting
```

Step 4 Use the **end** command to exit global configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# end
```

What to do next

After enabling accounting augmentation, configure protocol attributes in access and accounting requests.

Configure protocol attributes in access and accounting requests

Perform this task to create an attribute filter-list and to bind it with authentication and accounting requests.

Before you begin

The device must be in IBNS 2.0 mode before performing this task.

Follow these steps to configure protocol attributes in access and accounting requests:

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode.

Example:

```
Router> enable
```

Enter your password if prompted.

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **access-session attributes filter-list list** *list-name* command to add access-session protocol data to accounting and authentication records and enter common filter list configuration mode.

Example:

```
Router(config)# access-session attributes filter-list list mylist
```

The **filter-list** keyword configures a sensor protocol filter list to accounting and authentication records.

Step 4 Use the { **cdp** | **dhcp** | **dhcpv6** | **http** | **lldp** | **vlan-id** } command to include the specified protocol for the attribute.

Example:

```
Router(config-com-filter-list)# dhcp
```

Step 5 Use the **exit** command to exit common filter list configuration mode and return to global configuration mode.

Example:

```
Router(config-com-filter-list)# exit
```

Step 6 Use the **access-session** { **accounting** | **authentication** } **attributes filter-spec include list** *list-name* command to configure a sensor protocol filter specification, and bind an attribute filter list with accounting and authentication records.

Example:

```
Router(config)# access-session authentication attributes filter-spec include list mylist
```

Step 7 Use the **end** command to exit global configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# end
```

What to do next

Create a protocol filter.

Create a protocol filter

Perform this task to create a CDP, LLDP, DHCP, mDNS, or DHCPv6 filter containing TLVs that can be included or excluded in the device sensor output.

Before you begin

Follow these steps to create a protocol filter:

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode.

Example:

```
Router> enable
```

Enter your password if prompted.

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **device-sensor** { **filter-list** { **cdp** | **dhcp** | **dhcpv6** | **mdns** | **lldp** } } **list** *tlv-list-name* command to apply a sensor protocol filter list and enter configuration mode, where you can configure individual TLVs.

Example:

```
Router(config)# device-sensor filter-list cdp list cdp-list
```

- **list** *list-name*: Specifies the protocol TLV filter list name.

Step 4 Use the **option** { **name** *option-name* | **number** *option-number* } command to add individual DHCP options to the option list.

Example:

```
Router(config-sensor-cdplist)# option name SV30-0169266 number 10
```

This step applies only to DHCP and DHCPv6 protocols.

You can delete the option list without individually removing options from the list by using the **no device-sensor filter-list dhcp list** *option-list-name* command.

Step 5 Use the **tlv** { **name** *tlv-name* | **number** *tlv-number* } command to add individual Cisco Discovery Protocol TLVs to the TLV list.

Example:

```
Router(config-sensor-cdplist)# tlv number 10
```

This step applies only to CDP, LLDP and MDNS protocols.

You can delete the TLV list without individually removing TLVs from the list by using the **no device-sensor filter-list cdp list** *tlv-list-name* command.

Step 6 Use the **end** command to exit global configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# end
```

What to do next

Apply a protocol filter to the sensor output.

Apply a protocol filter to the sensor output

Perform this task to apply a Cisco Discovery Protocol, LLDP, or DHCP filter to the sensor output. Session notifications are sent to internal sensor clients and accounting requests.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode.

Example:

```
Router> enable
```

Enter your password if prompted.

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **device-sensor filter-spec** { **cdp** | **dhcp** | **lldp** } { **exclude** { **all** | **list** *list-name* } | **include list** *list-name* } command to apply a specific protocol filter containing a list of TLV fields to the device sensor output.

Example:

```
Router(config)# device-sensor filter-spec cdp include list list1
```

- **exclude**: Specifies the TLVs that must be excluded from the device sensor output.

- **include**: Specifies the TLVs that must be included from the device sensor output.
- **all**: Disables all notifications for the associated protocol.
- **list** *list-name*: Specifies the protocol TLV filter list name.

Step 4 Use the **end** command to exit global configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# end
```

Track TLV changes

Perform this task to enable client notifications and accounting events for all TLV changes. By default, for each supported peer protocol, client notifications and accounting events will only be generated where an incoming packet includes a TLV that has not previously been received in the context of a given session.

Before you begin

Follow these steps to track TLV changes:

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode.

Example:

```
Router> enable
```

Enter your password if prompted.

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **device-sensor notify all-changes** command to enable client notifications and accounting events for all TLV changes.

Example:

```
Router(config)# device-sensor notify all-changes
```

Notification and accounting events are enabled for a new TLV is received or a previously received TLV is received with a new value in the context of a given session.

Use the **default device-sensor notify** or the **device-sensor notify new-tlvs** command to return to the default TLV.

Step 4 Use the **end** command to exit global configuration mode and returns to privileged EXEC mode.

Example:

```
Router(config)# end
```

Device sensor configuration verification commands

You can use these show commands to verify the device sensor configuration:

- To display protocol configuration details for all devices, you can use the **show device-sensor details** command.

```
Device# show device-sensor details

        Device-Sensor Details
-------------------------------------

Status = Enabled

Protocols:
-----------
CDP             registered  Proto Tlv Limit = 128
LLDP            registered  Proto Tlv Limit = 128
DHCP            registered  Proto Tlv Limit = 128
MDNS            registered  Proto Tlv Limit = 128
DHCPv6          registered  Proto Tlv Limit = 128

Protocol Filter Configuration:
---------------------------------
CDP              Include List - cdp-list
LLDP             Include List - lldp-list
DHCP             Include List - dhcp-list
MDNS             Include List - mdnsList
DHCPv6           Include List - dhcpv6List
```

- To display sensor cache entries (the list of protocol TLVs or options received from a device) for a specific device, you can use the **show device-sensor cache mac** command.

```
Device# show device-sensor cache mac bcd2.9531.8105

Device: bcd2.9531.8105 on port GigabitEthernet0/1/5
-------------------------------------------------------------------------------
Proto Type:Name                           Len Value                       Text
LLDP     8:management-address              16 00 08 00 0C 05 01 0A 01 01  .........
                                              05 03 00 00 00 26 00        .....&.
LLDP     1:chassis-id                      11 00 01 00 07 04 BC D2 95 31  .....<R^U1
                                              81 00                       ^A.
CDP   4105:<unknown>                       42 10 09 00 26 50 49 44 3A 43  ...&PID:C
                                              39 35 30 30 2D 32 34 59 34  9500-24Y4
                                              43 2C 56 49 44 3A 56 30 33  C,VID:V03
                                              2C 53 4E 3A 46 44 4F 32 35  ,SN:FDO25
                                              31 34 30 57 43 4D           140WCM
CDP     22:mgmt-address-type              17 00 16 00 0D 00 00 00 01 01  .........
                                              01 CC 00 04 0A 01 01 05     .L......
CDP     19:cos-type                        5 00 13 00 01 00              .....
CDP     18:trust-type                      5 00 12 00 01 00              .....
CDP     10:native-vlan-type                6 00 0A 00 02 00 0A           ......
CDP      9:vtp-mgmt-domain-type            9 00 09 00 05 63 69 73 63 6F  ....cisco
CDP      4:capabilities-type               8 00 04 00 04 00 00 00 29     .......)
CDP      3:port-id-type                   23 00 03 00 13 54 77 65 6E 74  ....Twent
                                              79 46 69 76 65 47 69 67 45  yFiveGigE
                                              31 2F 30 2F 35              1/0/5
```

```
CDP      6:platform-type               21 00 06 00 11 63 69 73 63 6F  ....cisco
                                          20 43 39 35 30 30 2D 32 34   C9500-24
                                          59 34 43                     Y4C
CDP      5:version-type               293 00 05 01 21 43 69 73 63 6F  ...!Cisco
                                          20 49 4F 53 20 53 6F 66 74   IOS Soft
                                          77 61 72 65 20 5B 49 4F 53  ware [IOS
                                          58 45 5D 2C 20 43 61 74 61  XE], Cata
                                          6C 79 73 74 20 4C 33 20 53  lyst L3 S
                                          77 69 74 63 68 20 53 6F 66  witch Sof
                                          74 77 61 72 65 20 28 43 41  tware (CA
                                          54 39 4B 5F 49 4F 53 58 45  T9K_IOSXE
                                          29 2C 20 45 78 70 65 72 69  ), Experi
                                          6D 65 6E 74 61 6C 20 56 65  mental Ve
                                          72 73 69 6F 6E 20 32 36 2E  rsion 26.
                                          30 31 2E 32 30 32 35 31 30  01.202510
                                          30 39 3A 30 31 33 31 33 34  09:013134
                                          20 5B 42 4C 44 5F 50 4F 4C   [BLD_POL
                                          41 52 49 53 5F 44 45 56 5F  ARIS_DEV_
                                          4C 41 54 45 53 54 5F 32 30  LATEST_20
                                          32 35 31 30 30 39 5F 30 30  251009_00
                                          33 35 30 39 2D 30 2D 67 37  3509-0-g7
                                          30 37 65 36 65 39 37 66 61  07e6e97fa
                                          34 33 37 3A 2F 6E 6F 62 61  437:/noba
                                          63 6B 75 70 2F 6D 63 70 72  ckup/mcpr
                                          65 2F 73 32 63 2D 62 75 69  e/s2c-bui
                                          6C 64 2D 77 73 20 31 30 31  ld-ws 101
                                          5D 0A 43 6F 70 79 72 69 67  ].Copyrig
                                          68 74 20 28 63 29 20 31 39  ht (c) 19
                                          38 36 2D 32 30 32 35 20 62  86-2025 b
                                          79 20 43 69 73 63 6F 20 53  y Cisco S
                                          79 73 74 65 6D 73 2C 20 49  ystems, I
                                          6E 63 2E 0A 43 6F 6D 70 69  nc..Compi
                                          6C 65 64 20 54 68 75 20 30  led Thu 0
                                          39 2D 4F 63 74 2D 32 35 20  9-Oct-25
                                          30 31 3A 33 31 20 62 79 20  01:31 by
                                          6D 63 70 72 65              mcpre
CDP     11:duplex-type                  5 00 0B 00 01 01              .....
CDP      2:address-type                17 00 02 00 0D 00 00 00 01 01  ..........
                                          01 CC 00 04 0A 01 01 05     .L......
CDP      1:device-name                 13 00 01 00 09 63 61 74 39 35  ....cat95
                                          30 30 5F 31                 00_1
```

- To display sensor cache entries for all devices, you can use the **show device-sensor cache all** command.

```
Device# show device-sensor cache all

Device: a400.4e07.2688 on port GigabitEthernet0/1/0
-------------------------------------------------------------------------------
Proto Type:Name                          Len Value                           Text
LLDP     5:system-name                    19 00 05 00 0F 53 45 50 41 34  ....SEPA4
                                             30 30 34 45 30 37 32 36 38  004E07268
                                             38                          8
LLDP     1:chassis-id                     10 00 01 00 06 05 01 00 00 00  ..........
                                             00                          .
CDP      6:platform-type                  23 00 06 00 13 43 69 73 63 6F  ....Cisco
                                             20 49 50 20 50 68 6F 6E 65   IP Phone
                                             20 37 38 36 31               7861
CDP      5:version-type                   33 00 05 00 1D 73 69 70 37 38  ....sip78
                                             78 78 2E 31 34 2D 31 2D 31  xx.14-1-1
                                             2D 30 32 31 31 2D 31 33 34  -0211-134
                                             2E 6C 6F 61 64 73           .loads
CDP      1:device-name                    19 00 01 00 0F 53 45 50 41 34  ....SEPA4
```

```
                                        30 30 34 45 30 37 32 36 38  004E07268
                                        38                         8
```

- To display sensor cache entries (the list of protocol TLVs or options received from a device) for a specific interface, you can use the **show device-sensor cache interface** command.

```
Device# show device-sensor cache interface gi0/1/4

Device: 6c13.d547.1958 on port GigabitEthernet0/1/4
-------------------------------------------------------------------------------
Proto Type:Name                           Len Value                       Text
DHCP    50:requested-address                8 00 32 00 04 29 01 01 17     .2..)...
DHCP    54:server-identifier                8 00 36 00 04 29 01 01 01     .6..)...
DHCP    12:host-name                       19 00 0C 00 0F 53 45 50 36 43  ....SEP6C
                                              31 33 44 35 34 37 31 39 35  13D547195
                                              38                         8
LLDP     5:system-name                     19 00 05 00 0F 53 45 50 36 43  ....SEP6C
                                              31 33 44 35 34 37 31 39 35  13D547195
                                              38                         8
LLDP     1:chassis-id                      10 00 01 00 06 05 01 29 01 01  ......)..
                                              17                         .
CDP      2:address-type                    17 00 02 00 0D 00 00 00 01 01  ..........
                                              01 CC 00 04 29 01 01 17     .L..)...
CDP      6:platform-type                   23 00 06 00 13 43 69 73 63 6F  ....Cisco
                                              20 49 50 20 50 68 6F 6E 65   IP Phone
                                              20 38 38 35 31               8851
```

- To troubleshoot any issues in the device sensor configuration, you can use the **debug device-sensor** command.

```
router# debug device-sensor ?
  errors     Device Sensor Errors
  events     Device Sensor Events
  ha-events  Device Sensor HA logs
  verbose    Device Sensor verbose logs
```

Configuration examples of device sensor

Device sensor configuration examples

This section provides some sample device sensor configurations.

Cisco Discovery Protocol filter configuration example

This example shows how to create a Cisco Discovery Protocol filter containing a list of TLVs:

```
 Device> enable
Device#  configure terminal
Device(config)# device-sensor filter-list cdp list cdp-list
Device(config-sensor-cdplist)# tlv name address-type
Device(config-sensor-cdplist)# tlv name device-name
Device(config-sensor-cdplist)# tlv number 34
Device(config-sensor-cdplist)# end
```

LLDP filter configuration example

This example shows how to create an LLDP filter containing a list of TLVs:

```
Device> enable
Device# configure terminal
Device(config)# device-sensor filter-list lldp list lldp-list
Device(config-sensor-lldplist)# tlv name chassis-id
Device(config-sensor-lldplist)# tlv name management-address
Device(config-sensor-lldplist)# tlv number 28
Device(config-sensor-lldplist)# end
```

DHCP filter creation example

This example shows how to create a DHCP filter containing a list of options:

```
 Device> enable
Device# configure terminal
Device(config)# device-sensor filter-list dhcp list dhcp-list
Device(config-sensor-lldplist)# option name address-type
Device(config-sensor-lldplist)# option name device-name
Device(config-sensor-lldplist)# option number 34
Device(config-sensor-lldplist)# end
```

Apply a Cisco Discovery Protocol TLV filter example

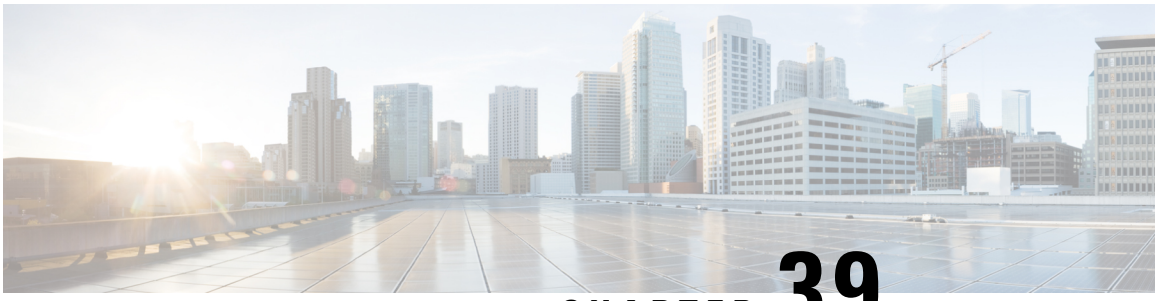
This example shows how to apply a Cisco Discovery Protocol TLV filter list to the device sensor output:

```
Device> enable
Device# configure terminal
Device(config)# device-sensor filter-spec cdp include cdp-list1
Device(config-sensor-lldplist)# end
```

Enable client notifications and accounting events example

This example shows how to enable client notifications and accounting events for all TLV changes:

```
Device>  enable
Device# configure terminal
Device(config)# device-sensor notify all-changes
Device(config)#  end
```



CHAPTER 39

DHCP Option 82

Table 30: Feature History Table

Feature name	Release information	Feature description
DHCP Option 82	Release 26.2.1	Starting with Cisco IOS XE Release 26.2.1, DHCP servers on routers support DHCP Option 82 relay information. The DHCP server can use the Circuit ID and Remote ID inserted by a DHCP relay agent to select an address pool for a client. This support enables predictable, location-based IP address assignment without requiring administrators to track client MAC addresses.

- DHCP Option 82, on page 463
- Prerequisites, on page 464
- Key considerations, on page 465
- Limitations, on page 465
- DHCP Option 82 packet flow, on page 465
- Hexadecimal pattern and suboption decoding, on page 466
- Configure DHCP Option 82 on server, on page 467
- Configure DHCP address pool, on page 469
- Configure DHCP relay agent, on page 470
- Configure relay chaining on relay agents, on page 472
- Verify DHCP Option 82, on page 473
- Debug DHCP Option 82 logs, on page 473

DHCP Option 82

The Dynamic Host Configuration Protocol (DHCP) Option 82 feature allows the server to allocate dynamic IP addresses based on the relay information sent by the relay agent. It enables predictive IP address assignment by the server based on network switch ports.

DHCP Option 82 enables DHCP relay agents to insert additional information such as Circuit ID, Remote ID and other configuration information into DHCP request messages before forwarding them from the client to the server.

Starting with Cisco IOS XE Release 26.2.1, DHCP servers support DHCP Option 82 on routers. This platform-independent feature includes the following suboptions:

- Circuit ID suboption: Identifies the relay agent's source interface or port where the client is connected.
- Remote ID suboption: Identifies the relay agent.

The DHCP server allows for address reservation based on Option 82 values, enabling consistent IP assignments per interface or location.

Benefits of DHCP Option 82

The use of DHCP Option 82 provides several operational and security advantages for network management such as:

- Predictable IP assignment: Enables static or predictable IP assignment for DHCP clients based on switch ports instead of MAC addresses.
- Enhanced security: Improves IP address management and security.
- Operational efficiency: Eliminates manual intervention or the need to track device MAC addresses.
- Granular control: Allows ISPs and network administrators to map specific user accounts to physical locations, enabling consistent, location-based IP assignments.

Prerequisites

Before configuring DHCP Option 82, you must ensure these requirements are met:

- A relay agent must be configured with global and VLAN-based DHCP snooping enabled.
- A DHCP server is available.
- A DHCP client is connected.
- A relay chaining topology is present wherever applicable.
- The client must be connected to the relay agent through an SVI interface for enabling VLAN-based snooping.
- Class based support must be enabled on the DHCP server.



Note You must know the hexadecimal value of each byte location in DHCP Option 82 to configure the relay-information hexadecimal command.For more information, see Circuit ID and Remote ID decode values

Key considerations

Key considerations for using DHCP Option 82 are:

- DHCP Option 82 support must be enabled on both the relay agent and DHCP server.
- You need to consider whether to use the DHCP Option 82 feature or not for predictable usage of IP assignment for DHCP clients.
- You need to consider whether to enable DHCP Option 82 globally for all interfaces or selectively on specific VLANs or interfaces only.
- You need to consider which information you want to match in the DHCP message received by the server. This can be Circuit ID, Remote ID or both.

Limitations

Limitations while using DHCP Option 82 are:

- Support must be enabled on both the relay agent and the DHCP server for it to work properly.
- If the DHCP server ignores DHCP Option 82, port-based deterministic assignment will not occur.
- It is not suitable for environments where MAC based IP address assignment is required.

DHCP Option 82 packet flow

Summary

DHCP Option 82 enables network administrators to track the location information of DHCP clients by inserting additional metadata as DHCP packets traverse relay agents. This process ensures accurate IP assignment and aids in network management.

Workflow

The DHCP packet flow through a network when DHCP Option 82 is enabled is as given:

1. The DHCP client broadcasts a DHCP request on the local network.
2. The first DHCP relay agent intercepts the broadcast and inserts the DHCP Option 82 field, which includes suboptions such as Circuit ID and Remote ID.
3. If a second relay agent resides closer to the DHCP server, the second relay agent encapsulates the DHCP Option 82 field coming from the first relay agent and forwards it to the DHCP server. This process preserves the information from both relay agents.
4. The relay agent sets the gateway IP address (giaddr) field in the DHCP packet to its own IP address. The giaddr field enables the DHCP server to identify the subnet from which the request originated.
5. The relay agent unicasts the DHCP packet to the DHCP server.

6. The DHCP server uses the DHCP Option 82 information to determine the appropriate IP address pool and configuration parameters for the client.
7. The DHCP server sends the reply to the relay agent using the giaddr address.
8. The relay agent removes the DHCP Option 82 information from the reply before forwarding the DHCP reply to the client.

Hexadecimal pattern and suboption decoding

The DHCP Option 82 hexadecimal pattern can be decoded for Circuit ID and Remote ID separately as shown in the examples.

Hexadecimal pattern sample

The Option 82 hexadecimal pattern `010600040005010a020701057065747261` consists of two primary suboptions:

- `01 06 00040005010a`: Circuit ID (Suboption 1).
- `02 07 01057065747261`: Remote ID (Suboption 2).

Field	Meaning
01	Circuit-ID suboption.
06	Circuit-ID length.
00040005010a	Circuit-ID value.
02	Remote-ID suboption.
07	Remote-ID length.
01057065747261	Remote-ID value.

Circuit ID Decode

The Circuit ID `00040005010a` identifies client's point of attachment specifying the port and VLAN of the connected client. By default, it follows the **vlan-mod-port** format.

```
Router# show cdp neighbors
         Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
         device1         Gig 1/10            175           R S I  ESR-6300- Gig 0/1/0

Router# show vlan brief
         VLAN          Name                               Status    Ports
         5    TURBINE_INTERNAL                 active    Gi1/10
```

Note Decoding the Circuit ID `00040005010a`:

- `00`: Default format (vlan-mod-port).
- `04`: Length (4 bytes).
- `0005`: VLAN ID (Hex 0005 = Decimal 5).
- `010a`: Module 1, Port 10 (Hex 0a = Decimal 10).

Remote ID Decode

The Remote ID `01057065747261` identifies the originating device (MAC address or Hostname). It can be configured to use the device hostname.

```
Router(config)# ip dhcp snooping information option format remote-id hostname
```

Note Decoding the Remote ID `01057065747261`:

- `01`: Hostname format.
- `05`: Length (5 bytes).
- `7065747261`: Hexadecimal ASCII for router.

Configure DHCP Option 82 on server

To configure DHCP Option 82 on Cisco IR router server, you need to follow these tasks:

- Enable DHCP Option 82 for DHCP address allocation on the server.
- Define the DHCP class and relay agent information patterns.
- Define the DHCP address pool on the server.
- Enable relay chaining on DHCP relay agents.
- Verify DHCP Option 82
- Debug DHCP Option 82 logs.

Follow these steps to configure DHCP Option 82 on the server.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp class** *class_name* command to enable DHCP address allocation using DHCP class.

Example:

```
Router(config)# ip dhcp class class_name
```

Note
- You can use Step 2 to enable DHCP Option 82 for DHCP address allocation on the server.
- You can disable this functionality, without deleting the DHCP class configuration.

Step 3 Use the **relay agent information** command to enter relay agent information option configuration mode and define the DHCP class and relay agent information patterns.

Example:

```
Router(dhcp-class)# relay agent information
```

Note The `relay agent information` command validates only the presence of Option 82 in the DHCP packet. It does not evaluate the Circuit ID or Remote ID. As a result, any DHCP packet containing Option 82, regardless of its source port or relay agent will match this DHCP class and be assigned an IP address.

Step 4 (Optional) Use the **relay-information hex** *pattern* [*] [**mask** *mask*] command to optionally specify a hexadecimal value for the full relay information option.

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 010600040005010a020701057065747261
```

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 01030a0b0c02050000000123 mask
00000000000000000ffffffffffffffff
```

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 010600040005010a*
```

Note

- The Option 82 hexadecimal pattern includes two sub options: one for Circuit ID and the other for Remote ID.
- Each suboption can be decoded to represent the suboption type, length, VLAN, module port, client interface and MAC address.
- You can match the hexadecimal pattern either fully or partially for specific scenarios of IP assignment.
- You can use the mask option to mask either the Circuit ID or Remote ID partially.
- You can use * as a wildcard pattern at the end of the hexadecimal pattern to match only the beginning of the relay information value.
- If you omit this step, no pattern is configured and it is considered a match to any relay agent information option value, but the relay information option must be present in the DHCP packet.
- Repeat Step 4 for each DHCP class you need to configure.

What to do next

For more information, see Configuring DHCP Address Allocation Using Option 82.

Configure DHCP address pool

Follow these steps to configure and define DHCP address pool for DHCP address allocation.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp pool** *name* command to configure a DHCP address pool on a DHCP server and enter DHCP pool configuration mode.

Example:

```
Router(config)# ip dhcp pool ABC
```

Step 3 Use the **network** *network-number* [*mask* |*prefix-length*] command to configure the subnet number and mask for a DHCP address pool on the DHCP server.

Example:

```
Router(dhcp-pool)# network 10.0.20.0 255.255.0.0
```

Step 4 Use the **class** *name* command to associate a class with a pool and enters DHCP pool class configuration mode.

Example:

```
Router(dhcp-pool)# class CLASS1
```

Note This command will also create a DHCP class if the DHCP class is not yet defined.

Step 5 Use the **address range** *start-ip end-ip* command optionally to set an address range for a DHCP class in a DHCP server address pool.

Example:

```
Router(dhcp-pool-class)# address range 10.0.20.1 10.0.20.100
```

Note
- Repeat Step 5 for each DHCP class you need to associate to the DHCP pool.
- If this command is not configured for a class, the default value is the entire subnet of the pool.

Configure DHCP relay agent

Use this task to configure the client-facing interface as a relay agent port on the router.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp snooping** command to enable the DHCP snooping feature globally

Example:

```
Router(config)# ip dhcp snooping
```

Step 3 Use the **ip dhcp snooping vlan** *vlan-id* command to enable DHCP snooping for VLAN.

Example:

```
Router(config)# ip dhcp snooping vlan 5
```

Step 4 Use these steps to configure SVI:

a) Use the **interface** *vlan-id* command to select the SVI corresponding to the client VLAN.

Example:

```
Router(config)# interface Vlan5
```

b) Use the **ip address** *ip-address subnet-mask* command to assign an IP address and subnet mask to the SVI.

Example:

```
Router(config-if)# ip address 10.166.1.1 255.255.0.0
```

c) Use the **ip helper-address** *ip-address* command to specify the IP address of the DHCP server.

Example:

```
Router(config-if)# ip helper-address 110.1.1.1
```

Step 5 Use the **interface** command to select the specific interface connected to the client.

Example:

```
Router(config)# interface GigabitEthernet1/10
```

Step 6 Use the **description** *client* command to provide a label for the client connection.

Example:

```
Router(config-if)# description CLIENT-1
```

Step 7 Use the **switchport mode access** command to set the interface to access mode.

Example:

```
Router(config-if)# switchport mode access
```

Step 8 Use the **switchport access vlan** command to assign the interface to the client VLAN.

Example:

```
Router(config-if)# switchport access vlan 5
```

Step 9 (Optional) Use these commands to define custom Circuit ID and Remote ID values, overriding the defaults:

a) (Optional) Use the **ip dhcp snooping vlan** *vlan-id* **information option format-type circuit-id string** *circuit-id-string* command to override the default Circuit ID (**vlan mod port**) with a user-defined string.

Example:

```
Router(config-if)# ip dhcp snooping vlan 5 information option format-type circuit-id
string abc
```

b) (Optional) Use the **ip dhcp snooping information option format remote-id string** *remote-id-string* command to override the default Remote ID (MAC address) with a user-defined string.

Example:

```
Router(config)# ip dhcp snooping information option format remote-id string abc
```

c) Use the **exit** command to return to global configuration mode.

Example:

```
Router(config-if)# exit
```

What to do next

For more information, see Configuring the Cisco IOS XE DHCP Relay Agent.

Configure relay chaining on relay agents

Use this task to configure relay chaining when multiple relay agents are used in the network. Enabling encapsulation on the secondary relay agent ensures that the original DHCP Option 82 data is preserved as the request is forwarded to the DHCP server.

Before you begin

Note

- Ensure DHCP snooping is enabled on both relay agents for their respective client VLANs.
- Relay chaining requires that DHCP snooping be enabled on both relay agents for their respective client VLANs. Ensure that interfaces are correctly configured between the relay agents to allow the passage of DHCP traffic.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp relay information option vpn** command to enable the inclusion of VPN information in the relay agent inforamtion.

Example:

```
Router(config)# ip dhcp relay information option vpn
```

Step 3 Use the **ip dhcp relay information policy encapsulate** command to enable the encapsulation policy on the secondary relay agent.

Example:

```
Router(config)# ip dhcp relay information policy encapsulate
```

Note In a double relay agent scenario, this configuration must be applied on the relay agent closest to the server.

Step 4 Use the **exit** command to return to privileged EXEC mode.

Example:

```
Router(config)# exit
```

An example of an encapsulated hexadecimal pattern format received by server in relay chaining is as given:

```
091f000000091a0a18bea80a0152120106000401f40105020800066 8d972ad3540020c020a00003c3c3c010e000000
```

Verify DHCP Option 82

Use these steps to verify DHCP Option 82 configuration.

Procedure

Step 1 Use these commands on the DHCP server side:

a) Use the **more system:running-config** command to display the running configuration.

Example:

```
Router# show running-config
```

Note This command will display the configured DHCP classes and DHCP pools.

b) Use the **show ip dhcp binding** command to display the client's MAC IP address.

Example:

```
Router# show ip dhcp binding
```

Step 2 Use these commands on the DHCP relay agent side:

a) Use the **show ip dhcp snooping binding** command to verify the relay agent's snooping bindings.

Example:

```
Router# show ip dhcp snooping binding
```

b) Use the **show ip dhcp snooping** command to validate the Circuit ID and Remote ID.

Example:

```
Router# show ip dhcp snooping
```

Debug DHCP Option 82 logs

Use this task to troubleshoot DHCP Option 82 configurations and derive the required hexadecimal patterns for class matching by enabling debug logs on both the DHCP server and the relay agent.

Procedure

Step 1 Use the **debug ip dhcp server** command on the DHCP server to monitor server-side events, classes, and packets.

Example:

```
Router# debug ip dhcp server event
         Router# debug ip dhcp server class
         Router# debug ip dhcp server packet
```

Step 2 Use the **debug ip dhcp snooping** command on the relay agent to monitor snooping events, agent processing, and packets.

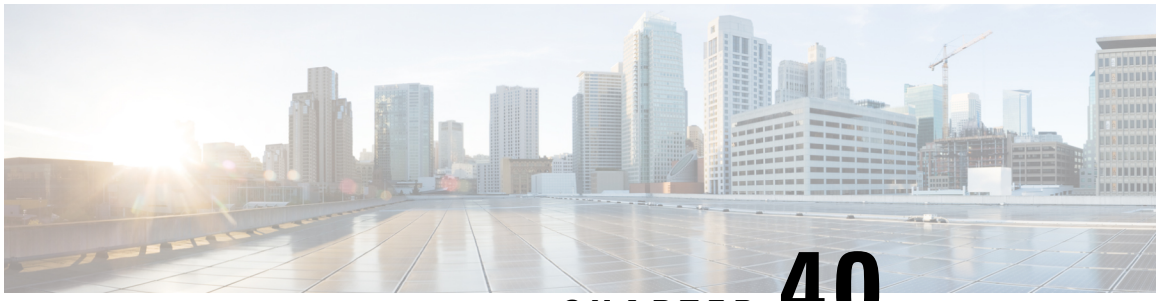
Example:

```
Router# debug ip dhcp snooping agent
         Router# debug ip dhcp snooping event
         Router# debug ip dhcp snooping packet
```

Note You can analyze the generated logs to identify the `relay-information` hexadecimal pattern being received by the server.

Example:

```
*May 13 2026 08:25:19.629: DHCPD: Searching for a match to 'relay-information
010600040005010a020701057065747261' in class class_turbine
          *May 13 2026 08:25:19.629: DHCPD: input pattern 'relay-information
010600040005010a020701057065747261' matches class CLASS1
```



CHAPTER 40

Segment Routing over MPLS

Table 31: Feature History Table

Feature name	Release information	Feature description
Segment Routing over MPLS	Release 26.2.1	Starting from Cisco IOS XE Release 26.2.1, you can configure Segment Routing over Multiprotocol Label Switching (SR-MPLS) on supported WAN ports. SR-MPLS encodes an ordered list of segments in the packet and integrates with MPLS services such as Layer 3 VPN and Ethernet VPN.

- Segment Routing over MPLS, on page 475
- Prerequisites, on page 476
- Configure Segment Routing over MPLS, on page 476
- Configure Segment Routing MPLS on OSPF network, on page 477
- Configure Prefix-SID for OSPF, on page 478
- Verify SR-MPLS configuration, on page 479

Segment Routing over MPLS

Segment Routing over Multi-protocol Label Switching (SR-MPLS) on WAN ports is a routing architecture that provides a flexible and scalable approach to source routing. It allows the source to define a path and encode it in the packet header as an ordered list of segments. Each segment is identified by a Segment ID (SID). When operating using the MPLS dataplane, segment routing integrates with the multi-service capabilities of MPLS, including Layer 3 VPN (L3VPN) and Ethernet VPN (EVPN).

Note SR-MPLS is applied specifically to the WAN ports of the supported routers. Ensure the WAN interfaces are active and configured with the appropriate routing protocols (OSPF or IS-IS) before proceeding.

Benefits of using SR-MPLS

SR-MPLS helps in:

- Scalability: The network does not need to maintain per-application or per-flow states.
- Traffic Engineering: Offers lower latency and more effective bandwidth utilization than traditional MPLS networks.
- Automatic Protection: Provides automatic traffic protection against link and node failures without requiring additional signaling.

Prerequisites

Before configuring SR-MPLS on your router, you must ensure that:

- Cisco Catalyst IR1101 or IR8340 routers with active WAN ports are used.
- The software image supports SR-MPLS configuration for these routers.
- Intermediate System to Intermediate System (IS-IS) and Open Shortest Path First (OSPF) routing protocols are enabled and configured for SR-MPLS operation.

Configure Segment Routing over MPLS

Use this task to enable SR-MPLS mode.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **segment-routing mpls** command to enable segment routing MPLS.

Example:

```
Router(config)# segment-routing mpls
```

Step 3 Use the **global-block** *start-range end-range* command to enable Segment Routing Global Block (SRGB) to define the label range for Prefix-SIDs.

Example:

```
Router(config)# global-block 16000 23999
```

For more information, see Segment Routing Global Block

Step 4 Use the **connected-prefix-sid-map** command to enter the sub-mode for configuring address-family specific mappings.

Example:

```
Router(config-srmpls)# connected-prefix-sid-map
```

Step 5 Use the **address-family** *ipv4* command to specify the IPv4 address family.

Example:

```
Router(config-srmpls-conn)# address-family ipv4
```

Step 6 Use the *ip-address/mask* **index** *sid-index* **range** *range-value* command to associate a Segment Identifier (SID) with the IP address.

Example:

```
Router(config-srmpls-conn-af)# 4.4.4.4/32 index 4 range 1
```

Step 7 Use the **exit-address-family** command to exit the address family configuration mode and return to the **connected-prefix-sid-map** sub-mode.

Example:

```
Router(config-srmpls-conn-af)# exit-address-family
```

Configure Segment Routing MPLS on OSPF network

Use this task to define the network topology, select OSPF as the control protocol, and configure the necessary SID mappings.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **router ospf** *process-id* command to enable the OSPF routing process.

Example:

```
Router(config)# router ospf 10
```

Step 3 Use the **router-id** *id* command to manually configure the OSPF router ID.

Example:

```
Router(config-router)# router-id 10.0.0.1
```

Step 4 Use the **segment-routing mpls** command to enable segment routing under the OSPF process.

Example:

```
Router(config-router)# segment-routing mpls
```

Step 5 Use the **segment-routing area** *area-id* **mpls** command to configure segment routing MPLS mode for a specific OSPF area.

Example:

```
Router(config-router)# segment-routing area 0 mpls
```

Configure Prefix-SID for OSPF

This task allows you to map a specific Prefix-SID to a loopback or interface address. Mapping the Prefix-SID to a loopback address creates a Node-SID, which serves as the node's unique identity. This enables other routers to forward traffic to this specific prefix using the associated SID label.

Before you begin

Ensure that segment routing is enabled on the corresponding address family.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **segment-routing mpls** command to enable segment routing MPLS mode.

Example:

```
Router(config)# segment-routing mpls
```

Step 3 Use the **connected-prefix-sid-map** command to enter the sub-mode for configuring address-family specific mappings.

Example:

```
Router(config-srmpls)# connected-prefix-sid-map
```

Step 4 Use the **address-family** *ipv4* command to specify the IPv4 address family.

Example:

```
Router(config-srmpls-conn)# address-family ipv4
```

Step 5 Use the *ip-address/mask* **index** *sid-index* **range** *range-value* command to associate a SID with the specified IPv4 prefix.

Example:

```
Router(config-srmpls-conn-af)# 10.1.1.1/32 index 100 range 1
```

Step 6 Use the **exit** command to exit to global configuration mode.

Example:

```
Router(config-srmpls-conn-af)# exit
```

Verify SR-MPLS configuration

Use this task to validate connectivity, path establishment, and resilience of your SR-MPLS deployment.

Before you begin

Verify the node SID reachability to ensure each node in the SR-MPLS domain can reach its designated node SID.

Procedure

Step 1 Use the **show segment-routing mpls connected-prefix-sid-map local ipv4** command to verify the SR-MPLS configuration on the router.

Example:

```
Router# show segment-routing mpls connected-prefix-sid-map local ipv4
```

Step 2 Use the **show mpls forwarding-table** command to check the MPLS label table.

Example:

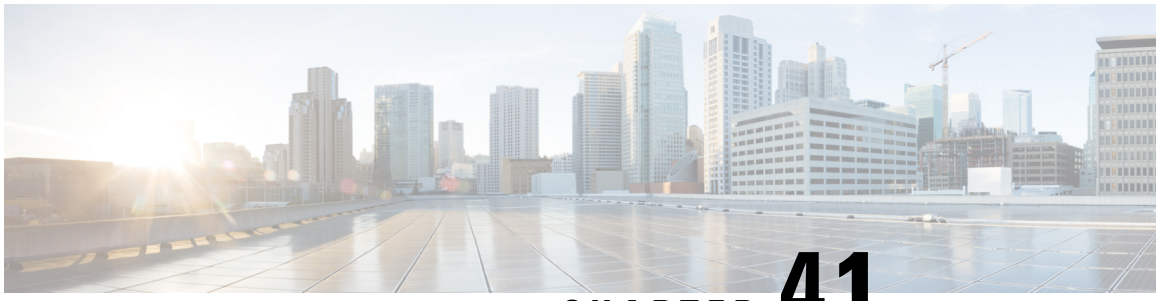
```
Router# show mpls forwarding-table
```

Step 3 Use the **ping mpls** command to validate the end-to-end connectivity between critical devices.



Note

- You can simulate link or node failures to ensure traffic is automatically rerouted along available alternate paths.
- You can also test adjacency SID reachability between adjacent nodes.



CHAPTER 41

SGT Inline Tagging on LAN Ports

Table 32: Feature History Table

Feature name	Release information	Feature description
SGT Inline Tagging on LAN Ports	Release 26.2.1	You can configure Security Group Tag (SGT) Inline Tagging on supported IR8340 LAN ports. The router can impose and propagate SGT metadata in Ethernet frames, enabling Cisco TrustSec microsegmentation without requiring Security Group Tag Exchange Protocol peering.

- SGT Inline Tagging support, on page 481
- Prerequisites, on page 482
- Limitations, on page 482
- Licensing, on page 482
- Configure SGT Inline Tagging, on page 482
- Scenarios for Troubleshooting, on page 483
- Verify CTS, on page 483
- Debug commands for CTS, on page 484

SGT Inline Tagging support

Starting with Cisco IOS XE Release 26.2.1, the Cisco Catalyst IR8340 Rugged Series Router supports Security Group Tag (SGT) Inline Tagging on LAN ports. This feature enables the router to perform Layer 2 (L2) SGT Imposition, which facilitates Cisco TrustSec micro-segmentation by embedding an 8-byte SGT metadata header into Ethernet frames.

L2-SGT Imposition allows you to enable specific Ethernet interfaces (GigabitEthernet0/1/1 to GigabitEthernet0/1/11) to insert an SGT into packets, which is then carried to the next-hop Ethernet neighbor.

SGT Inline Tagging works by using:

- SGT Assignment: An SGT is assigned to traffic at the ingress point or propagated from a trusted peer device to identify the source security group.

- Egress Determination: At the egress point, the Destination Group Tag (DGT) is determined.
- Access Control: Access decisions are enforced using Security Group Access Control Lists (SGACLs), which define permitted interactions between source and destination groups.

A network device at the ingress of a Cisco TrustSec cloud must determine the SGT of an incoming packet to tag it correctly before forwarding. The SGT can be determined using the following methods:

- Cisco TrustSec Header: If the packet originates from a trusted peer device, the router assumes the Cisco TrustSec header already contains the correct SGT. This method is used when the device is not the first hop within the Cisco TrustSec cloud by using the trusted keyword on the interface.
- If a packet does not have an SGT tag and the keyword **trusted** is configured on the interface, then the SGT assigned to the interface will be added to the traffic. SGT Inline Tagging feature is not supported without the **trusted** keyword.

Prerequisites

To use SGT Inline Tagging ensure that:

- LAN interface GigabitEthernet0/1/1 to GigabitEthernet0/1/11 is available.
- Network Advantage license is enabled on the router.

Limitations

The limitations for using the SGT Inline Tagging feature on the Cisco Catalyst IR8340 Rugged Series Router are:

- CTS inline tagging functions on a LAN port only when the egress interface is also a LAN port.
- Only SGT imposition (tagging) and tag propagation are supported.
- CTS enforcement is not supported on LAN ports.

Licensing

This feature requires a Network Advantage license.

Configure SGT Inline Tagging

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **interface gigabitethernet** *port* command to select the interface on which Cisco TrustSec SGT authorization and forwarding has to be enabled.

Example:

```
Router(config)# interface gigabitethernet 0/1/11
```

Step 3 Use the **cts manual** command to enable Cisco TrustSec SGT authorization and forwarding on the interface and enter Cisco TrustSec manual interface configuration mode.

Example:

```
Router(config-if)# cts manual
```

Step 4 Use the **propagate sgt** command to enable Cisco TrustSec SGT propagation on the interface.

Example:

```
Router(config-if-cts-manual)# propagate sgt
```

Step 5 Use the **policy static sgt** *tag* [**trusted**] command to configure a static SGT ingress policy and define the trustworthiness of an SGT received on the interface.

Example:

```
Router(config-if-cts-manual)# policy static sgt 77 trusted
```

Step 6 Use the **end** command to exit to privileged EXEC mode.

Example:

```
Router(config-if-cts-manual)# end
```

Scenarios for Troubleshooting

If SGT Inline Tagging fails or is not working as expected, consider these troubleshooting steps:

- Verify SGT tagging and propagation using packet capture or show commands. See *Verify CTS*.
- Check for correct tag imposition and propagation.
- Use debug commands to verify the tagging and propagation. See *Debug commands for CTS*.

Verify CTS

Use this task to check the Cisco TrustSec (CTS) status and related information on a specific GigabitEthernet interface.

Before you begin

Ensure you have privileged EXEC access to the device.

Procedure

Step 1 Use the **enable** command to enter privileged EXEC mode.

Example:

```
Router> enable
```

Enter your password if prompted.

Step 2 Use the **show cts interface gigabitEthernet** *interface-id* command to verify if CTS is enabled on the interface or not.

Example:

```
Router#sh cts interface gigabitEthernet 0/1/1
Global Dot1x feature is Disabled
Interface GigabitEthernet0/1/1:
    CTS is enabled, mode:    MANUAL
    IFC state:               OPEN
    Interface Active for 00:00:07.111
    Authentication Status:   NOT APPLICABLE
        Peer identity:       "unknown"
        Peer's advertised capabilities: ""
    Authorization Status:    NOT APPLICABLE
    SAP Status:              NOT APPLICABLE
    Propagate SGT:           Enabled
    Cache Info:
        Expiration            : N/A
        Cache applied to link : NONE

    Statistics:
        authc success:              0
        authc reject:               0
        authc failure:              0
        authc no response:          0
        authc logoff:               0
        sap success:                0
        sap fail:                   0
        authz success:              0
        authz fail:                 0
        port auth fail:             0

    L3 IPM:   disabled.

    CTS sgt-caching Ingress : Disabled

    CTS sgt-caching Egress  : Disabled
```

Debug commands for CTS

Use this task to enable various debug options for CTS to assist in troubleshooting and monitoring CTS operations.

Procedure

Step 1 Use the **debug platform software cts** command with one of the following options to enable CTS debug logs.

Example:

```
Router# debug platform software cts all
Router# debug platform software cts debug
Router# debug platform software cts error
Router# debug platform software cts info
```

all: Enables all CTS platform debug messages.

debug: Enables CTS platform debug logs.

error: Enables CTS platform error logs.

info: Enables CTS platform informational logs.

Step 2 Use the **debug cts all** command to enable debugging of all CTS messages.

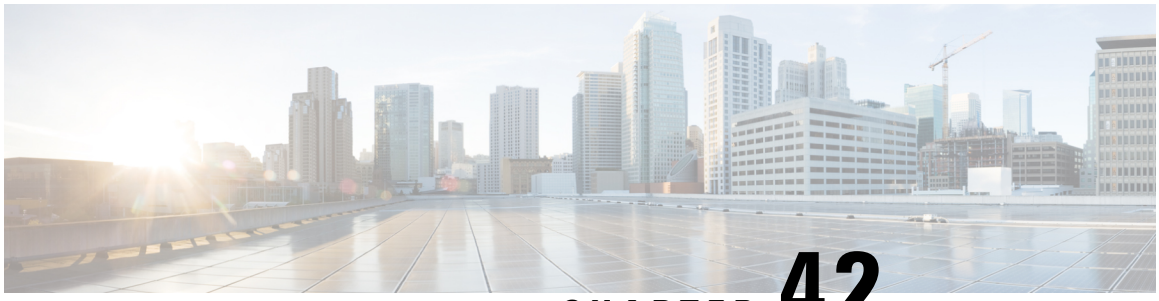
Example:

```
Router# debug cts all
```

Step 3 Use the **debug cts** command followed by one of the specific CTS debug options to enable targeted debugging.

Example:

```
Router# debug cts aaa
Router# debug cts authentication
Router# debug cts authorization
Router# debug cts cache
Router# debug cts coa
Router# debug cts critical-authentication
Router# debug cts dp
Router# debug cts environment-data
Router# debug cts error
Router# debug cts ha
Router# debug cts ifc
Router# debug cts layer3-trustsec
Router# debug cts odm
Router# debug cts policy-server
Router# debug cts provisioning
Router# debug cts rcl-server
Router# debug cts relay
Router# debug cts sap
Router# debug cts server-list
Router# debug cts sgacl-db
Router# debug cts states
Router# debug cts sxp
```



CHAPTER 42

IKEv2 and IPsec PFS Post Quantum Cryptography

Table 33: Feature History Table

Feature name	Release information	Feature description
IKEv2 and IPsec PFS Post Quantum Cryptography	Release 26.2.1	Cisco IoT routers can combine a traditional Diffie-Hellman key exchange with the NIST-standardized Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) to establish keys for IKEv2/IPsec sessions. This hybrid approach helps protect recorded VPN traffic against future harvest-now, decrypt-later attacks.

- IKEv2 and IPsec PFS Post Quantum Cryptography, on page 487
- Benefits of PQC, on page 488
- Prerequisites, on page 488
- Limitations, on page 488
- ML-KEM algorithm and parameter sets, on page 489
- PFS negotiation behavior, on page 490
- Configure an IKEv2 proposal for PQC, on page 490
- Configure Perfect Forward Secrecy with PQC, on page 492
- Verify the PQC hybrid key exchange, on page 493
- Troubleshooting PQC hybrid key exchange, on page 494

IKEv2 and IPsec PFS Post Quantum Cryptography

Quantum-Safe Encryption refers to cryptographic techniques designed to protect network communications against attacks by computers and future cryptographically relevant quantum computers.

Post-Quantum Cryptography (PQC) is part of this broader security approach. PQC includes quantum-resistant algorithms for functions such as key establishment and digital signatures. These algorithms help mitigate harvest-now, decrypt-later attacks, in which an attacker records encrypted traffic and attempts to decrypt it later using a sufficiently capable quantum computer.

PQC does not replace existing public-key algorithms. Instead, it works alongside them to create a quantum-safe hybrid key exchange.

The Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) is a National Institute of Standards and Technology (NIST)-standardized post-quantum key-encapsulation mechanism. ML-KEM enables two peers to establish a shared secret that can be used to derive cryptographic session keys.

On supported Cisco routers, ML-KEM is combined with a traditional key-exchange algorithm to provide a hybrid key establishment for IKEv2/IPsec Perfect Forward Secrecy (PFS). The hybrid approach protects the negotiated session as long as at least one of the constituent key-establishment mechanisms remains secure.

For more information on PQC on Cisco routers with IKEv2 sessions, see Security and VPN Configuration Guide.

For more information on Quantum-Safe Encryption and the related protocols, see: https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/sec-vpn/b-security-vpn/m-sec-cfg-quantum-encryption-ppk.html.



Note Postquantum preshared keys (PPKs), including dynamic PPKs obtained through the Secure Key Integration Protocol (SKIP), can be used with ML-KEM-based hybrid key establishment.

Benefits of PQC

You can use the PQC feature to strengthen the security of your IPsec and IKEv2 connections with quantum-safe, future-proof protection for traffic between Cisco IoT routers.

The PQC hybrid key exchange provides these benefits:

- Future-proof security: Protects traffic against future quantum-computer-based attacks.
- Defense in depth: Combines a classical DH algorithm with ML-KEM, so the session stays secure even if one algorithm is broken.
- Smooth migration: Supports phased migration in existing networks through the optional fallback keyword.

Prerequisites

Before you configure the PQC hybrid key exchange, ensure that the following requirements are met:

- Both peers support PQC, or the optional keyword is used to allow fallback during migration.
- IKEv2/IPsec is configured and operational between the peers.

Limitations

The limitations with respect to using PQC are:

- IKEv2 always uses hybrid key exchange: a combination of Diffie-Hellman (DH) and ML-KEM algorithms. Pure-PQC key exchange is not supported for IKEv2.

- The encapsulation keys and ciphertexts used by ML-KEM-768 and ML-KEM-1024 may cause IKEv2 messages to exceed typical path maximum transmission units (MTUs), resulting in IP fragmentation and potential packet drops. To avoid IP fragmentation, enable IKEv2 fragmentation and set the IKEv2 fragmentation MTU to 1400 bytes.
- This feature provides post-quantum key establishment only. It does not provide post-quantum peer authentication. IKEv2 peers continue to use the configured authentication method, such as a preshared key or certificate-based authentication.
- ML-KEM establishes shared keying material; it does not encrypt user traffic. The encryption algorithm configured in the IPsec transform set, such as AES-CBC or AES-GCM, continues to encrypt the user traffic.

Note License, platform, and software support: This feature does not require an additional feature license. ML-KEM-based quantum-safe key establishment is supported on all Cisco IoT routers with Cisco IOS XE Release 26.2.1.

Supported routers

The supported routers are:

- Cisco Catalyst IR1101 Rugged Series Router,
- Cisco Catalyst IR1800 Rugged Series Router,
- Cisco Catalyst IR8340 Rugged Series Router,
- Cisco Catalyst IR8140 Heavy Duty Router, and
- Cisco ESR6300 Embedded Series Router.

Phased migration in Hub-and-Spoke deployments

If an older responder (hub) receives a PQC proposal from upgraded spokes (initiators), the negotiation fails.

To avoid this, it is recommended that you always upgrade the hub first and configure **optional** in the hub's IKEv2 proposal.

When any issue occurs, the responder logs the following errors:

IKEv2-INTERNAL: Transform type 6 invalid.

IKEv2-ERROR: Failed to parse the packet: The peer's proposal is invalid.

ML-KEM algorithm and parameter sets

Cisco IoT routers support the three National Institute of Standards and Technology (NIST)-standardized ML-KEM parameter sets: ML-KEM-512, ML-KEM-768, and ML-KEM-1024. During the key exchange, the peers exchange an encapsulation key (public key) and a ciphertext.

Table 34: ML-KEM parameter sets (sizes in bytes)

Parameter	ML-KEM-512	ML-KEM-768	ML-KEM-1024
Encapsulation key size (public key)	800	1184	1568
Decapsulation key size (private key)	1632	2400	3168
Shared secret key size	32	32	32
Ciphertext size	768	1088	1568

PFS negotiation behavior

Table 35: PFS negotiation behavior for Cisco IOS XE to Cisco IOS XE devices

Initiator/ Responder	no set pfs	set pfs	set pfs <group>	set pfs <group> pqc <mlkem>
set pfs	UP (ML-KEM inherit)	UP (ML-KEM inherit)	UP (DH+ML-KEM inherit)	UP (DH+ML-KEM inherit)
set pfs <*group*>	UP (DH)	DOWN	UP (DH)	DOWN
set pfs <*group*> **pqc** <*mlkem*>	UP (DH+ML-KEM)	UP (DH+ML-KEM inherit)	UP (DH+ML-KEM)	UP (DH+ML-KEM)

Note During initial session establishment, `CHILD_SA` output might display `PFS: N`. After the `CHILD_SA` rekeys, verify the negotiated ML-KEM and DH PFS settings.

Configure an IKEv2 proposal for PQC

Follow these steps to enable ML-KEM algorithm in an IKEv2 proposal.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **crypto ikev2 fragmentation mtu** *mtu-size* command to enable IKEv2 fragmentation and set the IKEv2 fragmentation MTU to 1400 bytes.

Example:

```
Router(config)# crypto ikev2 fragmentation mtu 1400
```

Step 3 Use the **crypto ikev2 proposal** *name* command to create or edit an IKEv2 proposal.

Example:

```
Router(config)# crypto ikev2 proposal prop1
```

Step 4 Use the **encryption** *aes-cbc-key* command to configure AES key in CBC mode.

Example:

```
Router(config-ikev2-proposal)# encryption aes-cbc-256
```

a) Use the **integrity** *integrity-algorithm* command to configure the IKEv2 integrity algorithm.

Example:

```
Router(config-ikev2-proposal)# integrity sha512
```

b) Use the **group** *group-number* command to configure Diffie-Hellman group for classical key exchange used with ML-KEM.

Example:

```
Router(config-ikev2-proposal)# group 19
```

Step 5 Use the **pqc mlkem1024 mlkem768 mlkem512 optional** command to enable one or more ML-KEM algorithm parameter sets.

Example:

```
Router(config-ikev2-proposal)# pqc mlkem1024 mlkem768 mlkem512 optional
```

Note The proposal enables **pqc mlkem768 optional** by default. The **optional** keyword allows the SA to fall back to a classical (non-PQC) negotiation if the peer does not support PQC, which supports coexistence during phased migration.

Step 6 Use these commands to create the IKEv2 policy and attach the proposal:

a) Use the **crypto ikev2 policy** *policy* command to create the IKEv2 policy.

Example:

```
Router(config)# crypto ikev2 policy pol1
```

b) Use the **match fvrf any** command to apply the IKEv2 policy to negotiations through any front-door VRF (FVRF).

Example:

```
Router(config-ikev2-policy)# match fvrf any
```

c) Use the **proposal** *proposal-name* command to add the PQC-enabled proposal to the IKEv2 policy.

Example:

```
Router(config-ikev2-policy)# proposal prop1
```

The **proposal** *proposal* command makes the PQC-enabled proposal available during IKEv2 negotiation for sessions that match this policy.

Step 7 Use these commands to associate the existing IKEv2 profile with the IPsec profile:

a) Use the **crypto ipsec profile** *profile-name* command to create an IPsec profile.

Example:

```
Router(config)# crypto ipsec profile gen
```

b) Use the **set ikev2-profile** *ikev2-profile-name* command to associate an existing IKEv2 profile with the IPsec profile.

Example:

```
Router(ipsec-profile)# set ikev2-profile prof1
```

Note The profile and IPsec transform set can already exist, if not you can create it using the create IPsec profile option given in step 7.

Step 8 Use these commands to apply the IPsec profile to the tunnel:

a) Use the **interface tunnel** *tunnel-interface-number* command to create a tunnel interface.

Example:

```
Router(config)# interface tunnel 1
```

b) Use the **tunnel protection ipsec profile** *ipsec-profile-name* command to protect the tunnel with IPsec profile.

Example:

```
Router(config-if)# tunnel protection ipsec profile gen
```

Step 9 Use the **end** command to return to privileged EXEC mode.

Example:

```
Router(config-if)# end
```

Configure Perfect Forward Secrecy with PQC

Perfect Forward Secrecy (PFS) adds extra randomness to derive session keys during each key exchange. With PQC, the key exchange can use DH, ML-KEM, or DH + ML-KEM. PFS is optional and can be disabled.

Follow these steps to configure PFS with PQC.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **crypto ipsec profile** *name* command to enter the IPsec profile.

Example:

```
Router(config)# crypto ipsec profile P1
```

Step 3 Use the **set pfs** *group* **pqc** *mlkem-algorithm* command to enable PFS, specify a DH group, and an ML-KEM algorithm, or both.

Example:

```
Router(ipsec-profile)# set pfs group19 pqc mlkem1024
```

Note When **set pfs** is used:

- Without PQC: Starting from Cisco IOS XE Release 26.2.1 onwards, the DH group is inherited from the IKEv2 SA.
- With PQC: The DH group is inherited from the IKEv2 SA. Exchanges are optimized for Cisco IOS XE to Cisco IOS XE deployments using an ML-KEM-only IPsec Child Security Association exchange.

Verify the PQC hybrid key exchange

Follow these steps to verify that the PQC hybrid key exchange is active.

Procedure

Step 1 Use the **show crypto ikev2 sa detail** command to verify the IKEv2 SA and the PQC key exchange algorithm.

Example:

```
Router# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local              Remote             fvrf/ivrf   Status
1         10.0.149.203/500   10.0.149.217/500   none/none   READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19, Auth sign: PSK, Auth
verify: PSK
PQC Key Exchange: ML-KEM-1024
...
Quantum-safe Encryption using PQC: ML-KEM-1024
PEER TYPE: IOS-XE
```

Step 2 Use the **show crypto ipsec sa** command to verify the IPsec SA. The PQC key exchange appears after the first IPsec SA rekey.

Example:

```
Router# show crypto ipsec sa
...
PFS (Y/N): Y, DH group: group19, PQC Key Exchange: ML-KEM-1024
```

Note The PQC key exchange value in the IPsec SA output appears only after an IPsec SA rekey.

When the show commands display the configured ML-KEM algorithm and the tunnel status is READY, the quantum-safe hybrid key exchange is active.

Troubleshooting PQC hybrid key exchange

Use these debug commands to troubleshoot PQC hybrid key exchange issues on a Cisco IoT router.

Follow these steps to debug the PQC hybrid key exchange.

Procedure

Step 1 Use the **debug crypto ikev2** commands to debug IKEv2 negotiation, errors, packets, and internal events.

Example:

```
Router# debug crypto ikev2
Router# debug crypto ikev2 error
Router# debug crypto ikev2 packet
Router# debug crypto ikev2 internal
```

Step 2 Use the **debug crypto kmi** command to debug Key Management Infrastructure (KMI) events.

Example:

```
Router# debug crypto kmi
```

Step 3 Use the **debug crypto socket** command to debug crypto socket events.

Example:

```
Router# debug crypto socket
```

Step 4 Use the **debug crypto ipsec** commands to debug IPsec negotiation, errors, messages, and state changes.

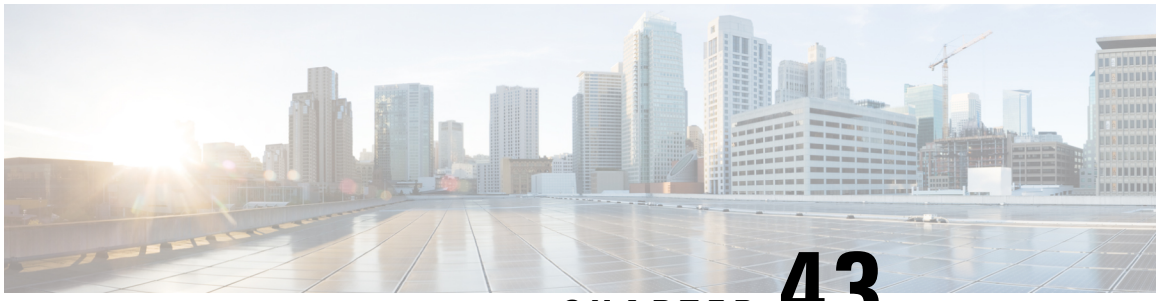
Example:

```
Router# debug crypto ipsec
Router# debug crypto ipsec error
Router# debug crypto ipsec message
Router# debug crypto ipsec state
```

Step 5 Use the **debug tunnel protection** command to debug tunnel protection events.

Example:

```
Router# debug tunnel protection
```



CHAPTER 43

Post Quantum Cryptography for SSHv2 sessions

Table 36: Feature History Table

Feature name	Release information	Feature description
Post Quantum Cryptography for SSHv2 sessions	Release 26.2.1	Cisco IoT routers can combine the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) with classical elliptic-curve key-exchange algorithms to establish keys for SSHv2 sessions.

- Implementation of Post Quantum Cryptography, on page 495
- Overview, on page 496
- Factors to consider for deploying PQC, on page 496
- Protocol requirements, on page 496
- Restrictions for PQC, on page 496
- PQC negotiation scenarios , on page 497
- Configure PQC for SSHv2 sessions, on page 497
- Verify PQC for SSHv2 sessions, on page 498
- Key elements of PQC, on page 498
- Supported platforms, on page 500

Implementation of Post Quantum Cryptography

Cisco utilizes a hybrid approach to implement Post Quantum Cryptography (PQC). The hybrid approach to PQC combines a quantum-resistant algorithm, such as the module-lattice based ML-KEM (e.g., mlkem768), with a classical elliptic curve key exchange algorithm like NIST P-256 or X25519, along with a hash function such as SHA-256.

This combination provides enhanced security by protecting against both classical and quantum attacks while maintaining compatibility and performance with existing cryptographic protocols.

Cisco implements this hybrid approach in protocols like SSH allowing configurations that enforce the use of ML-KEM algorithms for quantum resistance without falling back to classical-only key exchanges.

Overview

PQC refers to cryptographic algorithms designed to be secure against attacks from quantum computers which can break many classical encryption methods. Current widely used encryption methods—such as Diffie-Hellman (DH), and Elliptic Curve Diffie-Hellman (ECDH) — are vulnerable to being broken as quantum computing technology advances.

Implementing quantum-safe encryption now is critical for the long-term confidentiality and integrity of sensitive communications, particularly for government, military, and financial sectors.

Factors to consider for deploying PQC

Review these factors before deploying PQC.

Backward compatibility with classical algorithms
If a device is only configured for PQC and the SSH client does not support PQC, the SSH session will fail and no connection will be established.
Impact on computational time
The key size in PQC implementations may have an impact on the computational time due to the mathematical complexity and processing requirements of quantum-resistant algorithms. Larger key sizes generally increase the computational workload. PQC implementations must consider hardware acceleration and software optimizations to mitigate the increased computational time due to larger key sizes.

Protocol requirements

Protocol requirements

SSHv2 :

To establish secure, encrypted, and remote command-line access to servers and network devices over unsecured network use SSH version 2.0. PQC cannot be implemented on SSHv1 due to severe security vulnerabilities and it does not support the modern, complex key exchange mechanisms required to resist quantum computer attacks.

Key Pairs :

Ensure that RSA key pair or EC key pair for the SSH server is generated using crypto key generate command.

Restrictions for PQC

Implementing PQC is not supported on devices that operate in Federal Information Processing Standards (FIPS) mode.

PQC negotiation scenarios

During the SSH protocol connection setup, the client and server negotiate which cryptographic algorithms to use for securing the session. This negotiation process is a basic example of how algorithm negotiation works in SSH and is applicable to any similar negotiation process.

The algorithms are considered in the order they are configured on the device, meaning the priority or preference of algorithms is determined by the sequence in which they are listed in the configuration. The negotiation proceeds by trying the algorithms in that configured order until a mutually supported algorithm is found and agreed upon for use in the session. The server must support at least one algorithm from the client's list for successful negotiation.

Table 37: sfds

Client configuration	Server configuration	Negotiated algorithm
Classical, Hybrid	Classical, Hybrid	Classical
Hybrid, Classical	Hybrid, Classical	Hybrid
Classical, Hybrid	Hybrid	Hybrid
Hybrid, Classical	Classical	Classical
Classical	Classical, Hybrid	Classical
Hybrid	Classical, Hybrid	Hybrid
Classical	Hybrid	None
Hybrid	Classical	None
Mixed (No overlap)	Mixed (No overlap)	None

Configure PQC for SSHv2 sessions

Use these commands to set up devices as an SSH server and an SSH client. This allows the device to accept connection requests from remote SSH clients, enabling secure remote management of the device through SSH connections.

Note The PQC algorithms are not enabled by default and must be explicitly enabled.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip ssh server algorithm kex** *algorithm* command to configure the PQC based key-exchange algorithms that the SSH server supports.

Example:

```
Router(config)# ip ssh server algorithm kex mlkem768nistp256-sha256
```

Step 3 Use the **ip ssh client algorithm kex** *algorithm* command to configure the PQC based key-exchange algorithms that SSH client supports.

Example:

```
Router(config)# ip ssh client algorithm kex mlkem768x25519-sha256
```

This command is used to define the order and selection of key exchange (KEX) algorithms that the SSH client will use during the SSH handshake process.

Verify PQC for SSHv2 sessions

Use this section to understand how to verify PQC implementation for SSHv2.

Command	Purpose	What to look for in output
show ip ssh	Displays the status of the SSH server, its configuration details.	In the output of this command, look for these details that indicate PQC implementation is successful: KEX algorithms: **mlkem1024nistp384-sha384,** **mlkem768nistp256-sha256,** **mlkem768x25519-sha256**

This example shows the response when a PQC algorithm command is entered on a platform that does not support PQC.

```
Router(config)# ip ssh server algorithm kex mlkem1024nistp384-sha384 ecdh-sha2-nistp256
Router(config)#
000205: Jun 18 16:41:43.086: %SSH-3-KEX_PQC_NOT_SUPP: PQ/T hybrid key exchange algorithms
are not supported on this platform.
Router(config)#
```

Key elements of PQC

PQC is implemented by using the algorithms available as part of Module-lattice based Cryptography. Module-lattice based Cryptography is a type of quantum-resistant public-key cryptography that uses structured, high-dimensional grid points (lattices) to secure data. Cisco implements PQC into protocols like SSH using ML-KEM.

These algorithms are resistant to quantum attacks and are used for secure key establishment, encryption, and digital authentication or signatures.

This table lists the areas in which each algorithm can be used:

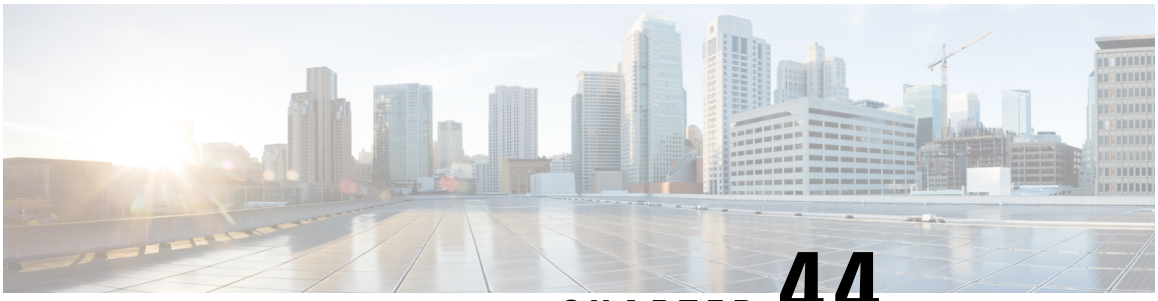
ML-KEM algorithm	What this algorithm means	Security Level
mlkem768x25519-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. X25519 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm. SHA-256 is the hash function used for cryptographic hashing in the key exchange process. In summary, mlkem768x25519-sha256 is a hybrid PQC key exchange algorithm combining the ML-KEM 768-bit lattice-based scheme with classical X25519 and SHA-256, used to secure key exchanges against quantum and classical threats in Cisco's PQC-enabled network protocols.	The security level of this implementation is balanced and is best suited for deployments where performance metrics are critical.
mlkem768nistp256-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. NIST P-256 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm standardized by NIST. SHA-256 is the hash function used for cryptographic hashing in the key exchange process. In summary, mlkem768x25519-sha256 is a hybrid PQC key exchange algorithm combining the ML-KEM 768-bit lattice-based scheme with classical X25519 and SHA-256, used to secure key exchanges against quantum and classical threats in Cisco's PQC-enabled network protocols.	The security level of this implementation is good and is best suited for most deployments.

ML-KEM algorithm	What this algorithm means	Security Level
mlkem1024nistp384-sha384	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. 1024 is the highest security parameter set, equivalent to the security level of AES-256. NIST P-384 is the traditional (classical) component. It uses Elliptic Curve Diffie-Hellman (ECDH) with the NIST P-384 curve to maintain security against current non-quantum threats. In summary, mlkem1024nistp384-sha384 represents a hybrid cryptographic proposal combining a quantum-resistant ML-KEM algorithm with classical elliptic curve cryptography (NIST P-384) and SHA-384 hashing. This hybrid approach ensures security against both classical and quantum attacks by negotiating keys using both classical and post-quantum algorithms.	The security level of this implementation is the highest and is best suited for deployments where maximum security is needed.

Supported platforms

PQC can be configured only on these platforms in autonomous mode:

- Cisco Catalyst IR1101 Rugged Series Routers,
- Cisco IR1000 Rugged Series Secure Routers
- Cisco Catalyst IR1800 Rugged Series Routers,
- Cisco Catalyst IR8140 Heavy Duty Series Routers,
- Cisco Catalyst IR8340 Rugged Series Routers,
- Cisco Embedded Service 6300 Series Routers.



CHAPTER 44

Cisco Network Plug and Play Agent

This chapter contains the following sections:

- Prerequisites for Cisco Network Plug and Play Agent, on page 501
- Restrictions for Cisco Network Plug and Play Agent, on page 502
- Information About Cisco Network Plug and Play Agent, on page 502
- Security Methods for the PnP Discovery Process, on page 517
- Security Methods for Post-PnP Discovery Process, on page 523
- How to Configure Cisco Network Plug and Play Agent, on page 526
- Troubleshooting and Debugging, on page 537
- Glossary, on page 538
- Additional References for Open Plug-n-Play Agent, on page 538

Prerequisites for Cisco Network Plug and Play Agent

- Cisco Network Plug and Play (PnP) deployment method depends on the type of discovery process as required by the customer.
- Deploy the discovery mechanism, either a DHCP server discovery process or a Domain Name Server (DNS) discovery process, before launching the PnP.
- Configure the DHCP server or the DNS server before deploying the PnP.
- Ensure that the PnP server talks to the PnP agent.
- Ensue that the Cisco Network PnP Agent has connectivity with the PnP Server. The Cisco Network PnP Agent should be able to PING server.
- The PnP agent enforces the PnP server to send user credentials for every request. Cisco recommends the usage of HTTP secure (HTTPS) protocol.

Note

- The terms Cisco Network Plug and Play, PnP are interchangeably used in this guide and all mean the same.
- The terms PnP agent, agent, and deployment agent are interchangeably used in this guide and all mean the same.
- The terms PnP server, server, and deployment server are interchangeably used in this guide and all mean the same.

Restrictions for Cisco Network Plug and Play Agent

- Cisco Network Plug and Play (PnP) agent facilitates HTTP and HTTP secure (HTTPS) transport based communication with the server.
- HTTPS cannot be used on platforms where crypto-enabled images are not supported (also, do not use Secure Sockets Layer [SSL] or Transport Layer Security [TLS] protocols if crypto-enabled images are used).
- Non-VLAN 1 configuration-Cisco Network Plug and Play supports devices using VLAN 1 by default. To use a VLAN other than 1, adjacent upstream devices must use supported releases and configure the following global CLI command on the upstream device to push this CLI to the upcoming Plug and Play device: **pnp startup-vlan x**. When you execute this command on an adjacent upstream device, the VLAN membership change does not happen on that device. However, all the active interfaces on the upcoming Plug and Play device are changed to the specified VLAN. This guideline applies to both routers and switches.
- PnP is not supported on LAN ports, which are GigabitEthernet ports 0/1/0 to 0/1/11. PnP is supported only on GigabitEthernet ports 0/0/0 and 0/0/1, and Cellular PIMs.

Note When performing a firmware upgrade during the PNP process, it is best to remove any old images on the router to prevent an incorrect image from loading.

Further details are available by viewing CSCwd68868.

Information About Cisco Network Plug and Play Agent

Cisco Network Plug and Play Deployment Solution

The Cisco Network PnP Agent is a part of Cisco Network Plug and Play solution. The Cisco initiated Network Plug and Play (PnP) deployment solution supports the concept of redirection and includes a PnP agent, a PnP server, and other components. Simplified deployment process of any Cisco device automates the following deployment related operational tasks:

- Establishing initial network connectivity for the device
- Delivering device configuration
- Delivering software and firmware images
- Delivering licenses
- Delivering deployment script files
- Provisioning local credentials
- Notifying other management systems about deployment related events

Simplified deployment reduces the cost and complexity and increases the speed and security of deployments.

Cisco Network Plug and Play (PnP) agent is a software application that is running on a Cisco IOS or IOS-XE device. The PnP agent together with the PnP deployment server provides effortless deployment services. When a device is powered on for the first time, the PnP agent process wakes up in the absence of the startup config, user input on the device's console, and attempts to discover the address of the PnP server. The PnP agent uses methods like DHCP, Domain Name System (DNS), and others to acquire the desired IP address of the PnP server. When the PnP agent successfully acquires the IP address, it initiates a long lived, bidirectional layer 3 connection with the server and waits for a message from the server. The PnP server application sends messages to the agent requesting for information and services to be performed on the device.

The PnP agent converges existing solutions into a unified agent and adds functionality to enhance the current solutions. The main objectives of PnP agent are:

- Provide consistent day 1 deployment solution for all the deployment scenarios.
- Add new features to improve existing solutions.
- Provide day 2 management framework mainly in the context of configuration and image upgrades.

Cisco Network Plug and Play Features

Some of the features that the Cisco Network Plug and Play agent provides:

- Day 0 boot strapping—Configuration, image, licenses, and other files
- Day 2 management—Configuration and image upgrades and on-going monitoring of Simple Network Management Protocol (SNMP) and syslog messages.
- Open communication protocol—Enables customers and partners to write applications
- XML based payload over HTTP between the server and the agent.
- Security—Authentication and encrypted communication channel between the management app and the agent
- Deployment and management of devices behind firewall and Network Address Translation (NAT).
- Support for one-to-one and one-to-many communication
- Support for policy based deployment (product ID or location of the device)
- Deployment based on unique ID (Unique Device Identifier [UDI] or MAC)
- Unified solution across Cisco platforms (including IOS classic)

- Support for various deployment scenarios and use cases
- Zero-touch when possible, low-touch when needed

Cisco Network Plug and Play Agent Services and Capabilities

The services and capabilities of the Cisco Network Plug and Play agent are as follows:

1. Backoff
2. CLI execution
3. Configuration upgrade
4. Device information
5. File transfer
6. Image install
7. License install
8. PnP tagging
9. Script execution
10. Topology information



Note The PnP server provides an optional checksum tag to be used in the image installation and config upgrade service requests by the PnP agent. When the checksum is provided in the request, the image install process compares the checksum against the current running image checksum.

If the checksums are same, the image being installed or upgraded is the same as the current image running on the device. The image install process will not perform any other operation in this scenario.

If the checksums are not same, then the new image will be copied to the local file system, and checksum will again be calculated and compared against the checksum provided in the request. If same, the process will continue to install the new image or upgrade the device to new image. If now, the checksums are not same, the process will exit with error.

Backoff

A Cisco IOS device that supports PnP protocol (that uses HTTP transport), requires the PnP agent to send the work request to the PnP server continuously. In case the PnP server does not have any scheduled or outstanding PnP service for the PnP agent to execute, the continuous no operation work requests exhausts both network bandwidth and device resource. This PnP backoff service allows the PnP server to inform the PnP agent to rest for the specified time and call back later.

CLI Execution

Cisco IOS supports two modes of command execution—EXEC mode and global configuration mode. Most of the EXEC commands are one-time commands, such as **show** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. The EXEC commands are not

saved when a device reboots. Configuration modes allow user to make changes to the running configuration. If you save the configuration, these commands are saved when a device reboots.

Note For **show** command request and response details and for all PnP configuration commands, see *Cisco Network Plug and Play Agent Command Reference*.

Configuration Upgrade

There are two types of configuration upgrades that can happen in a Cisco device—copying a new configuration files to startup configuration and copying new configuration files to running configuration.

Copying a new configuration files to startup configuration— The new configuration file is copied from the file server to the device through **copy** command and file check is performed to check the validity of the file. If the file is valid, then the file is copied to startup configuration. Backing up the previous configuration file will be done if there is enough disk space available. The new configuration is seen when the device reloads again.

Copying new configuration files to running configuration— The new configuration file is copied from the file server to the device through **copy** command or **configure replace** command. Configuration file replace and rollback may leave the system in an unstable state if rollback is performed efficiently. So configuration upgrade by copying the files is preferred.

Device Information

The PnP agent provides capability to extract device inventory and other important information to the PnP server on request. The following five types of device-profile requests are supported:

1. all—returns complete inventory information, which includes unique device identifier (UDI), image, hardware and file system inventory data.
2. filesystem— returns file system inventory information, which includes file system name and type, local size in bytes, free size in bytes, read flag, and write flag.
3. hardware— returns hardware inventory information, which includes hostname, vendor string, platform name, processor type, hardware revision, main memory size, I/O memory size, board ID, board rework ID, processor revision, midplane revision, and location.
4. image—returns image inventory information, which includes version string, image name, boot variable, return to rommon reason, bootloader variable, configuration register, configuration register on next boot, and configuration variable.
5. UDI— returns device UDI.

File Transfer

The PnP file server hosts files that can be copied over by the deploying devices in the network. The file server can be a dedicated server hosting files or a part of the device hosting the PnP server. The PnP agent uses standard file transfer protocols to copy files from the remote file server to the device. If the device is running a crypto image then secured file transfer protocols such as SFTP, SCP, HTTPS are supported. For devices running non-crypto images, the PnP agent supports unsecured copy protocols such as FTP, TFTP, HTTP.

Image Install

Image installation service enables a PnP-enabled device to perform image upgrade on receiving a request from the PnP server.

Standalone Devices

When the PnP agent on a standalone device receives a request from the PnP server, the agent parses the XML payload and identifies the request as an Image Upgrade request. The agent then creates an ImageInstall process, which identifies the request as a standalone image install request. The PnP agent populates the data structure defined by the ImageInstall service and passes it to the ImageInstall service.

The Image Install service then performs the following operations to successfully load the device with the new image:

1. Copies the image from the file server to a local disk (the file server information is provided by the PnP server in the request).
2. Configures the device to load the new image on next reload by executing the **boot system** command.
3. Reloads the device and sends a message to the PnP server.

PnP Tagging

Cisco IOS provides capability to assign tags to the devices for better grouping and tracking of all Cisco devices. The PnP agent provides XML service for configuring the tag information on the device and for propagating the tag information within the network using Cisco Discovery Protocol (CDP). The purpose of this service is for the PnP agents to get to know their tag information and to pass on this information to the PnP server upon request.

Topology Information

By default, every Cisco device on the network runs Cisco Discovery Protocol (CDP). Through CDP, devices in the network discover their immediate neighbors and populate their databases with the attributes learnt or derived through the protocol. This neighbor information is stored in the database and is available on demand by the device to the PNP server. Typical neighbor information comprises neighboring device ID, software version, hardware platform, interface ip, and the port on which CDP messages are sent or received.

Software Maintenance Upgrade

The software maintenance upgrade (SMU) is a package that contains fixes for a specific defect or security resolution to a released image. SMUs are created to respond to immediate issues and do not include new features. SMUs do not have a large impact on router operations. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.

To install and activate a software maintenance upgrade package:

Procedure

Step 1 Use the **install add** <*filename*> command to unpack the package software file and copy it to the boot device (usually disk0). If the file is on a remote source, use the tftp/ftp option to copy the file to the device.

After the file is copied to the device, information within the package is used to verify compatibility with the target cards and with the other active software. Actual activation is performed only after the package compatibility and application program interface (API) compatibility checks are passed.

Step 2 To activate a package, use the **install activate** <*filename*> command. The activate operation will run the compatibility checks and install the software maintenance upgrade package. If it is a reload software maintenance upgrade, it will automatically initiate a reload.

Step 3 Use the **install commit** command to commit the changes

Step 4 To deactivate the package, use the **install deactivate** <*filename*> command.

Step 5 If you find that you prefer a previous package set over the currently active package set, you can use the **install rollback to committed** command to make a previously active package set active again

Step 6 To remove the installed version, use the **install remove** <*filename*> command.

This example shows how to install and remove the software maintenance upgrade package on a device.

```
install add <filename>
install activate <filename>
install commit
install rollback to committed
install remove <filename>
```

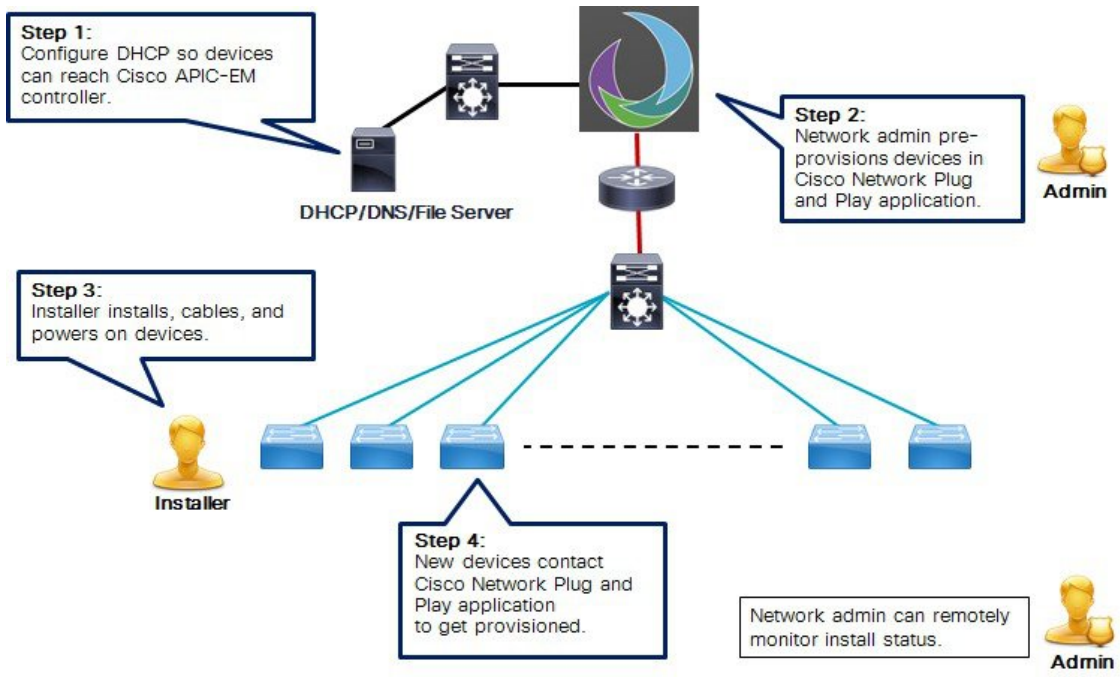
Cisco Network Plug and Play Agent

The Cisco Network Plug and Play agent is an embedded software component that is present in all Cisco network devices that support simplified deployment architecture. The PnP agent understands and interacts only with a PnP server. The PnP agent first tries to discover a PnP server, with which it can communicate. Once a server is found and connection established, the agent performs deployment related activities like configuration, image, license, and file updates by communicating with the server. It also notifies the server of all interesting deployment related events like out-of-band configuration changes and a new device connection on an interface.

Cisco Network Plug and Play Server

The Cisco Network Plug and Play server is a central server that encodes the logic of managing and distributing deployment information (images, configurations, files, and licenses) for the devices being deployed. The server communicates with the agent on the device that supports the simplified deployment process using a specific deployment protocol.

Figure 12: Simplified Deployment Server



The PnP server also communicates with proxy servers like deployment applications on smart phones and PCs, or other PnP agents acting as Neighbor Assisted Provisioning Protocol (NAPP) servers, and other types of proxy deployment servers like VPN gateways.

The PnP server can redirect the agent to another deployment server. A common example of redirection is a PnP server redirecting a device to communicate with it directly after sending the bootstrap configuration through a NAPP server. A PnP server can be hosted by an enterprise. This solution allows for a cloud based deployment service provided by Cisco. In this case, a device discovers and communicates with Cisco's cloud based deployment service for initial deployment. After that, it can be redirected to the customer's deployment server.

In addition to communicating with the devices, the server interfaces with a variety of external systems like Authentication, Authorizing, and Accounting (AAA) systems, provisioning systems, and other management applications.

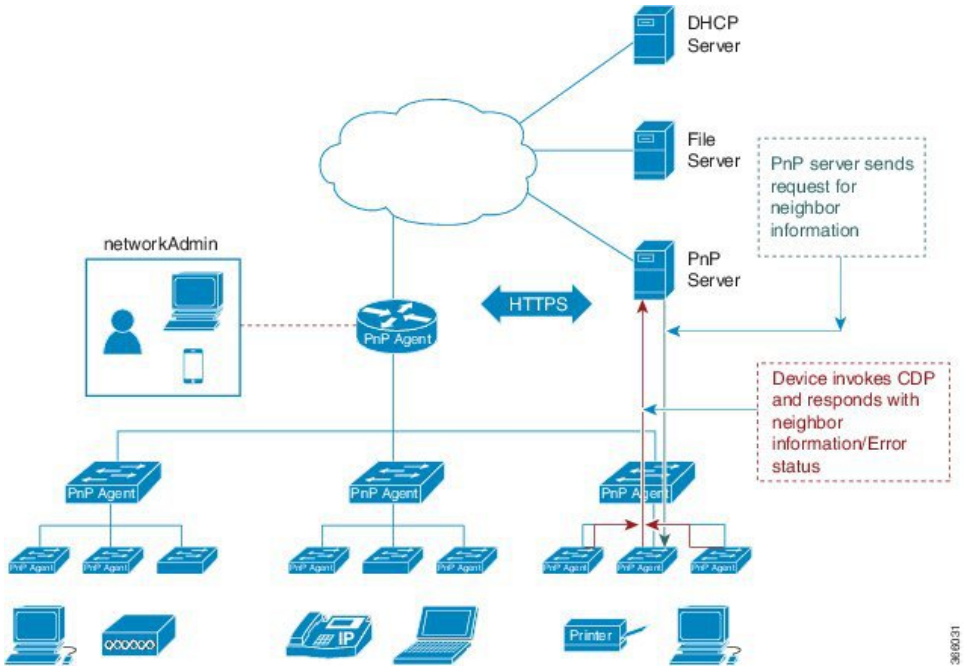
Cisco Network Plug and Play Agent Deployment

The following steps indicate the Cisco Network Plug and Play agent deployment on Cisco devices:

1. The Cisco device, having PnP agent contacts the PnP server requesting for a task, that is, the PnP Agent sends its unique device identifier (UDI) along with a request for work.
2. The PnP server if it has any task for the device, sends a work request. For example, image install, config upgrade, and so on.
3. When the PnP agent receives the work request, executes the task and sends back a reply to the PnP server about the task status, whether it is a success or error, and the corresponding information requested.

Cisco Network Plug and Play Agent Network Topology

Figure 13: Network Topology of Cisco Network Plug and Play Agent Deployment



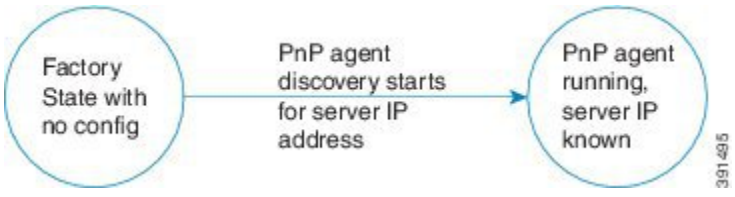
Cisco Network Plug and Play Agent Initialization

The Cisco Network Plug and Play agent software is currently available on all Cisco IOS XE platforms, and is enabled by default. The PnP agent can be initiated on a device by the following ways:

Absence of Startup Configuration

New Cisco devices are shipped to customers with no startup configuration file in the NVRAM of the devices. When a new device is connected to a network and powered on, the absence of a startup configuration and the user input file on the device, it will automatically trigger the Cisco Network Plug and Play agent to discover the PnP server IP address.

Figure 14: State Diagram for PnP Trigger with no Startup Configuration



CLI Configuration for Open Plug-n-Play Agent

Network administrators may use CLI configuration mode to initiate the Open Plug-n-Play (PnP) agent process at any time. By configuring a PnP profile through CLI, a network administrator can start and stop the PnP agent on a device. When the PnP profile is configured with CLI, the device starts the PnP agent process which, in turn, initiates a connection with the PnP server using the IP address in the PnP profile.

Figure 15: State Diagram for PnP Trigger with CLI Configured PnP Profile



Cisco Network Plug and Play Agent Deployment Solutions

This section discusses the functionality of the Cisco Network Plug and Play agent, exposed to the PnP server, for device deployment and management. The PnP agent deployment solution comprises the discovery process initiated by the agent, communication between the device, agent, and the server, and the PnP agent services. The PnP solution is described in detail in the following sections:

Cisco Network Plug and Play Agent Discovery Process

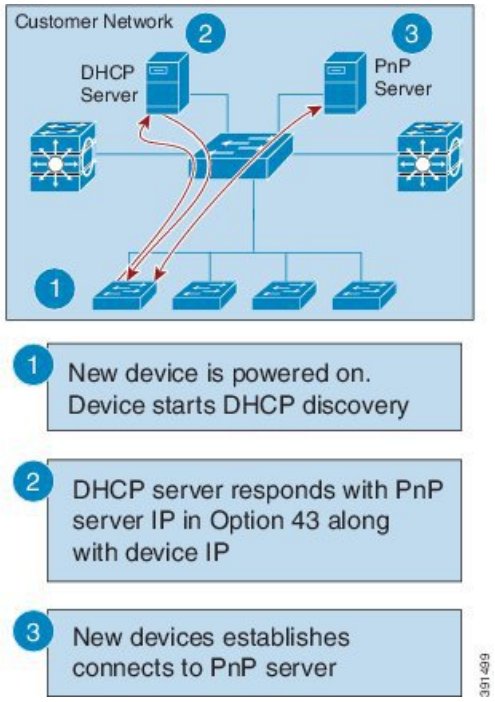
When the device boots up, the absence of any startup config on the NVRAM triggers the PnP discovery agent to acquire the IP address of the PnP server. In order to acquire the IP address of the PnP server, the PnP agent goes through one of the following discovery mechanisms:

1. PnP discovery through DHCP server
2. PnP discovery through DHCP snooping
3. PnP discovery through DNS lookup
4. PnP proxy for layer 2 and layer 3 devices
5. PnP deployment application

Cisco Network Plug and Play Discovery through DHCP Server

Device with no startup configuration in the NVRAM triggers the Cisco Network Plug and Play agent to initiate a DHCP discovery process which acquires the IPv4 configuration from the DHCP server required for the device. The DHCP server can be configured to insert additional information using vendor specific option 43 upon receiving option 60 from the device with the string ‘cisco pnp’, to pass on the IPv4 address or hostname of the PnP server to the requesting device. When the DHCP response is received by the device, the PnP agent extracts the option 43 from the response to get the IP address or the hostname of the PnP server. PnP agent then uses this IPv4 address or hostname to communicate with the PnP server.

Figure 16: DHCP Discovery Process for PnP server



Assumptions:

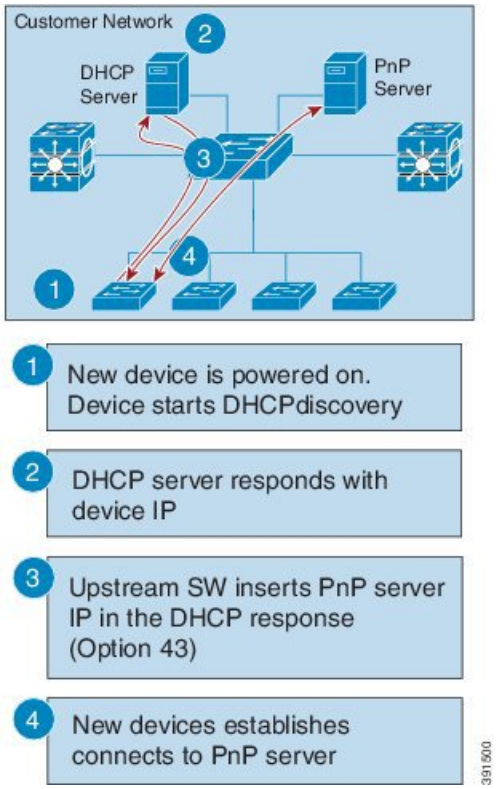
- New devices can reach DHCP server
- Customer is willing to configure DHCP server for network devices

Plug-n-Play Discovery through DHCP Snooping

If a third party DHCP server cannot be configured to insert any vendor specific options, an existing Cisco Open Plug-n-Play (PnP) enabled device can be configured to snoop in to the DHCP response and insert PnP specific option 43 with the PnP server IP address.

Before inserting the option 43, the snooping agent verifies if the DHCP message is from a Cisco device in the network. The remaining DHCP discovery process is same as described in the previous section.

Figure 17: DHCP Snooping by PnP Server



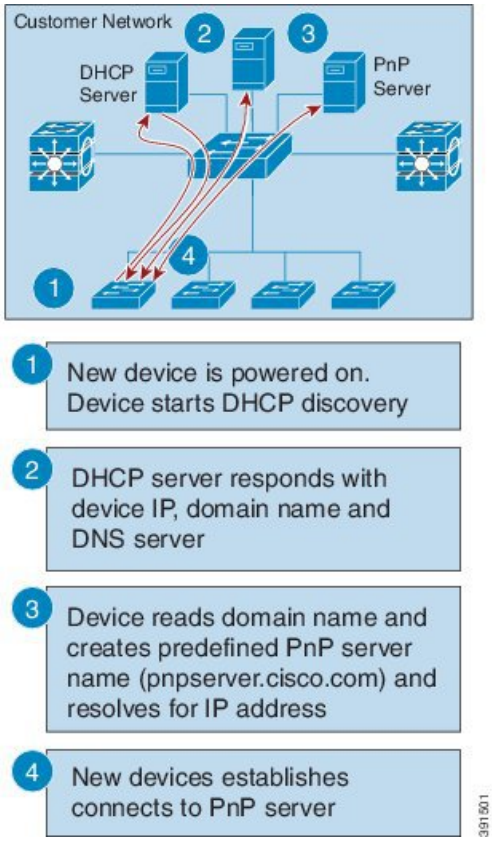
Assumptions:

- New devices can reach DHCP server
- New devices can reach DNS server
- Customer is not willing to configure DHCP server for network devices
- Upstream switch (SW) is configured to snoop DHCP and insert PnP server IP

Cisco Network Plug and Play Discovery through DNS Lookup

When the DHCP discovery fails to get the IP address of the Cisco Network Plug and Play server, the agent falls back on Domain Name System (DNS) lookup method. PnP agent then uses a preset deployment server name. The agent obtains the domain name of the customer network from the DHCP response and constructs the fully qualified domain name (FQDN). The following FQDN is constructed by the PnP agent using a preset deployment server name and the domain name information for the DHCP response, *deployment.customer.com*. The agent then looks up the local name server and tries to resolve the IP address for the above FQDN.

Figure 18: DNS Lookup for deployment.customer.com



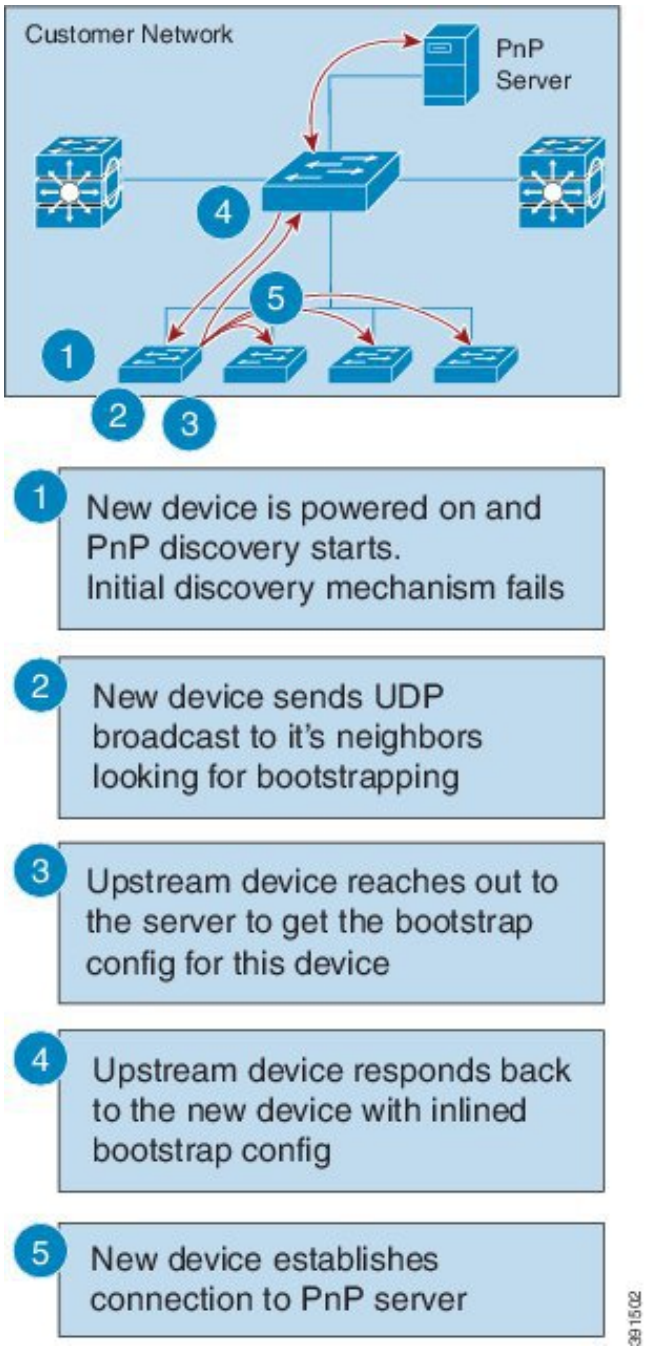
Assumptions:

- New devices can reach DHCP server
- Customer deployed PnP server in the network with the name “pnpserver”

Cisco Network Plug and Play Proxy Server for Layer 3 and Layer 2 Devices

This device listens to a specific port for any incoming PnP messages. The Cisco device which is trying to come up as a PnP device sends a UDP broadcast message to its network every 30 min for ten times. Hence, if the device does not receive a response, the broadcasts stop after 300 min.

Figure 19: DNS Lookup for Layer 3 and Layer 2 Devices



When the device hosting the proxy server process receives the incoming broadcasts, it verifies the version field in the request and forwards the request to the PnP server if version validation is successful. The proxy server process also caches the unique device identifier (UDI) of the requesting client coming in via incoming datagram before forwarding the request to PnP server.

Upon receiving the configlet datagram from PnP server, the proxy server validates UDI in the incoming datagram with the entries in the UDI cache. If validation is successful, proxy server process broadcasts the datagram to a specific port number reserved for the proxy client processes to receive datagrams.

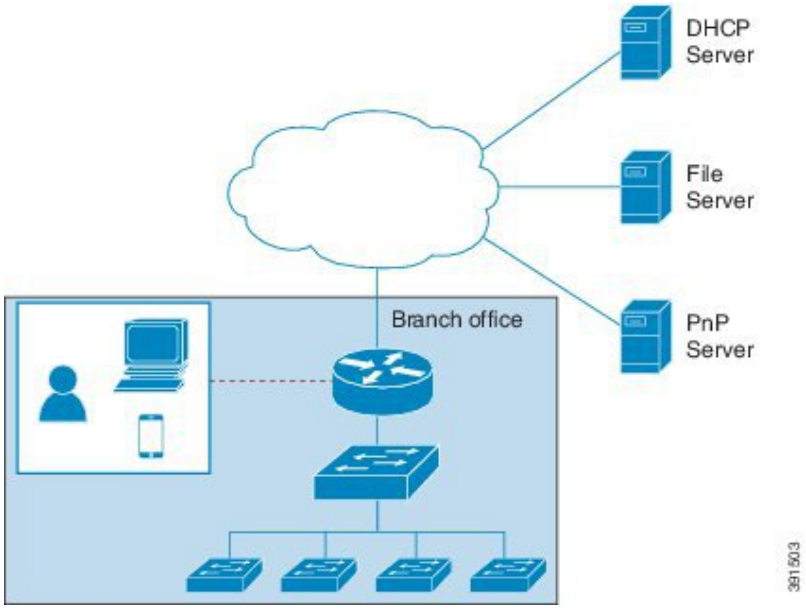
Upon receiving the datagrams, devices running proxy client processes, parse the incoming datagram for the target UDI. If the target UDI in the datagram matches the UDI of the device, proxy client process proceeds with framing, error control and configuring the configlet.

If the target UDI in the datagram fails to match UDI of the device, the packet is dropped.

Plug-n-Play Agent Deployment Application

A Cisco device can alternatively be manually configured by the network administrator using a deployment application running on their PC or on a smart phone. The PC or the smart phone can be connected to the device using an USB or an Ethernet cable.

Figure 20: Manually Configured PnP Agent



Plug-n-Play Agent Deployment Protocol

Deployment can be run over different transports. These transports include Ethernet and IP with Transport Layer Security (TLS). Layer 2 transport is typically used between a deployment agent and a proxy deployment server like a deployment application or as a deployment agent acting as a proxy. Transport between an agent and a server is over an IP connection with TLS for security. Transport between a proxy deployment server and a deployment server is also over IP with TLS.

Plug-n-Play Agent Application Protocol

The Cisco Open Plug-n-Play (PnP) agent application protocol is an XML-based protocol that defines a mechanism that allows network devices to be monitored and controlled by a remote application. The PnP agent is a software module running on a Cisco device. The PnP server is an application running as the network manager that remotely manages the network devices. The main features of the PnP protocol are as follows:

1. Supports HTTP protocols

2. Supports Transport Level Security (TLS) based encryption for HTTP
3. Uses HTTP secure (HTTPS) certificate for TLS handshake

Plug-n-Play Transport over Ethernet

Cisco Open Plug-n-Play (PnP) agent uses the Ethernet based transport in the following two scenarios:

- **Deployment agent communicating with a deployment application on a PC**: In this case, the PC is connected to the device being deployed using an Ethernet cable. The deployment application advertises itself as a deployment server supporting Ethernet transport.
- **Deployment agent communicating with an already deployed device acting as a proxy deployment server**: In this case, the new device being deployed has an Ethernet connection to an already deployed device. The deployment agent on the deployed device responds to the discovery requests and acts as a proxy deployment server for the new device.

Once discovery is complete, the deployment agent starts an unsecured XML stream with the deployment server over Ethernet. This protocol reserves an Ethertype (0xXX TBD) for this purpose. The deployment agent and the server then negotiate to use Extensible Authentication Protocol–Transport Layer Security (EAP-TLS) to protect the communication and complete the EAP-TLS session establishment. The deployment server then authenticates the device with the HTTP secure (HTTPS) certificate or some other supported mechanism.

Plug-n-Play Transport over IP

In Cisco Network Plug-n-Play (PnP) agent, the deployment agent opens TCP connection to the deployment server and starts an XML stream of messages. The server can request the use of Transport Layer Security (TLS) at this time. The agent closes the existing XML stream, initiates a TLS connection to the server, and then restarts the XML stream. The server can request agent authentication over the TLS connection.

Plug-n-Play Agent Security

Security to all Cisco Open Plug-n-Play (PnP) devices is provided at both transport level as well as the application level. The following sections describe the security mechanisms in detail:

Plug-n-Play Transport Layer 3 Security

For non-crypto or non-crypto-enabled images, TLS security choice is not possible. One alternative minimum security is to have the PnP agent initiate the connection to the specified trusted PnP server on port 5222.

Authentication and Authorization between Plug-n-Play Agent and Server

Once the Cisco Open Plug-n-Play (PnP) deployment agent discovers the PnP server, the agent engages the server in a Transport Layer Security (TLS) handshake. In order to authenticate itself to the server, the agent presents its HTTP secure (HTTPS) certificate. The administrator for the PnP server sets device authentication mechanisms which are acceptable for a particular deployment.

The deployment server presents its certificate to the deployment agent so that the agent can authenticate the server. Irrespective of whether the agent is able to verify the server certificate, the agent engages the deployment server in a post-TLS authorization exchange. In this exchange the agent requests the server to present its server authorization token. In response to this request the server presents the authorization token it had obtained from Cisco. The agent verifies the signature on the authorization token. If the authorization token is specific to a Unique Device Identifier (UDI), the agent also ensures its UDI is listed in the authorization-token. At the

end of this step, a secure communication channel is established between the deployment agent and the server. This secure communication channel is leveraged by the server to send deployment information to the agent.

Security Methods for the PnP Discovery Process

This section describes the methods that are used to secure the PnP agent-server communication in various scenarios. The security options are used by the PnP agent during the zero-touch PnP server discovery.

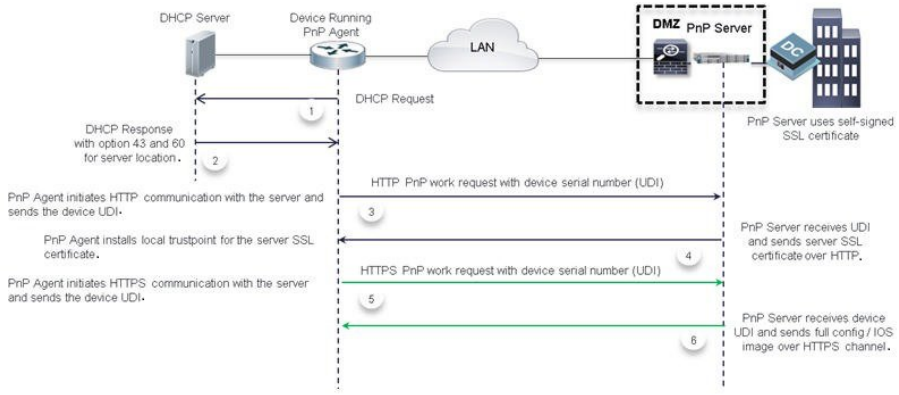
Self-Signed Certificate Based Authentication

The PnP server has an option to use a self-signed SSL certificate for server side authentication. When the PnP server uses a self-signed certificate, the PnP discovery cannot be used for automatically initiating secured communication from the agent to the server. The device goes through usual PnP discovery mechanisms and when it finds the server, the agent sends a work-request over HTTP. The server should use the PnP certificate-install service to instruct the agent to install the server self-signed certificate, and then automatically reconnect back to the server over HTTPs.

To keep the solution secured, it is recommended that you use the unsecured port 80 of server to deliver the one-time certificate installation to the devices. All other services should be sent over the secured port.

The following figure shows the end-to-end secured PnP workflow using a self-signed server SSL certificate.

Figure 21: PnP Deployment with Self-Signed Certificate

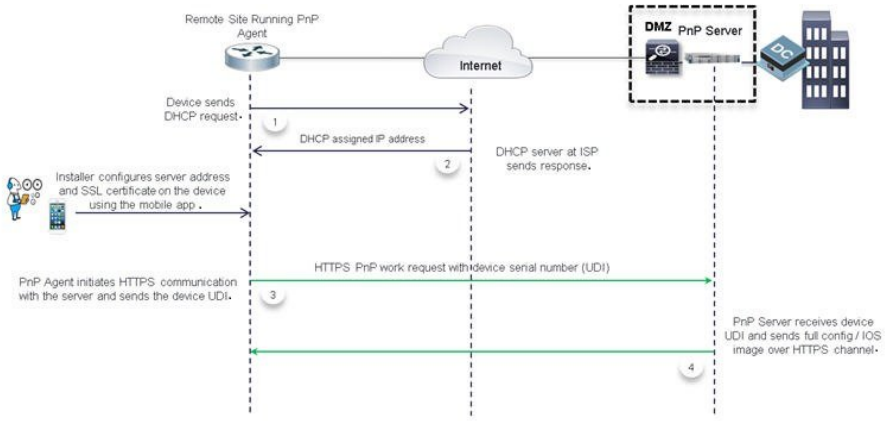


Mobile Device Based Secured Installation

As part of this solution, an application for mobile devices is available to configure the bootstrap on the devices. The mobile application can be used to install the server certificate directly on each device along with other bootstrap configuration and then allow the PnP agent initiate secured communication with the server. In this method, the server does not open up any unsecured port for certificate-install.

The following figure shows the end-to-end secured PnP workflow using the application on the mobile devices.

Figure 22: Secured PnP Deployment with the Mobile Application



CA-Signed Certificate based Authentication

Cisco distributes certificates signed by a signing authorities in a tar file format and signs the bundle with Cisco Certificate Authority (CA) signature. This certificate bundle is provided by Cisco infoSec for public downloads on cisco.com.

The certificates from this bundle can be installed on the Cisco IOS device for server side validation during SSL handshake. It is assumed that the server uses a certificate, which is signed by one of the CA that is available in the bundle.

The PnP agent uses the built-in PKI capability to validate the certificate bundle. As the bundle is signed by Cisco CA, the agent is capable of identifying the bundle that is tampered before installing the certificates on the device. After the integrity of the bundle is ensured by the agent, the agent installs the certificates on the device. After the certificates are installed on the device, the PnP agent initiates an HTTPs connection to the server without any additional steps from the server. The following mechanisms helps the PnP agent to initiate a zero-touch secured communication.

DHCP Option based Discovery over an IPv4 Network

The DHCP option 43 and option 60 is a vendor specific identifier which is used by the PnP agent to locate and connect to the PnP server. To support multiple vendors, the PnP agent in Cisco device sends out a case-sensitive “ciscopnp” as the option 60 string during the DHCP discovery. The DHCP server can be configured with multiple classes matching with a different option 60 string that comes from each network device. After the option 60 string matches, the DHCP server sends out the corresponding option 43 string back to the device. The following is the format for defining the option 43 for PnP deployments:

*option 43 ascii "5A;K5;B2;I10.30.30.10;J443;**Ttftp://10.30.30.10/ ios.p7b**;Z10.30.30.1*

The field ‘T’ in the PnP string provides an option for the network administrator to specify the location of the certificate bundle, which can be hosted on a local or remote file server.

If the certificate bundle is available at the specified location, then the agent:

1. Downloads the bundle from the file server to the device.
2. Checks the signature of the downloaded bundle to ensure it has a genuine Cisco signature.
3. Installs the certificates on the device.

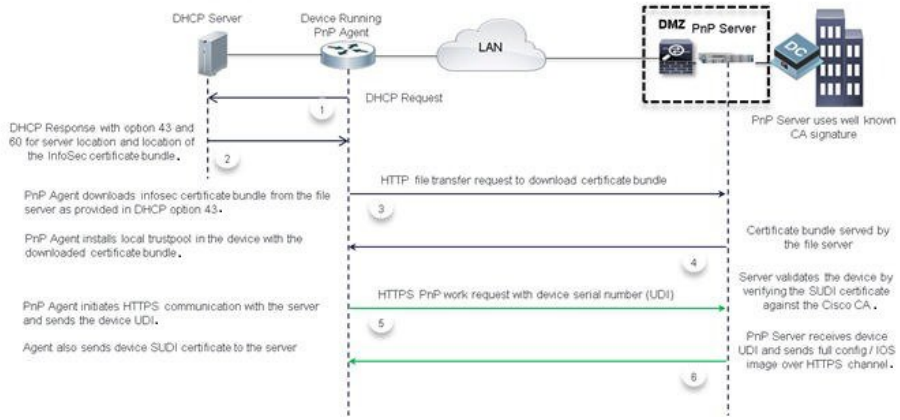
If the ‘T’ option is not specified and the transport mechanism is specified in the option 43 string as HTTPs, the PnP agent looks for the Cisco signed certificate bundle in the default folder of the same server *http://10.30.30.10:443/certificates/default/cert.p7b* .

If the certificates are available at the default location then the agent performs the steps mentioned above to install the certificates.

After the certificates are installed and the server discovery is complete, the agent initiates the HTTPs connection with the server without any additional configuration. During the HTTPs handshake, the device uses the certificates installed from the bundle to validate the server certificate.

The following figure shows the end-to-end secured PnP workflow using the CA bundle-based certificate.

Figure 23: Secured PnP Deployment with Trustpool



This flow works only if the server is using a certificate signed by one of the known signing authorities that is available in the bundle. If the server uses a certificate that is not a part of the bundle then the HTTPs handshake will fail. When you specify the option 43 string with HTTPs as a transport option and if the bundle download fails, the agent will not fall back to any of the unsecured communication protocol even if the server is reachable. If the transport option is specified as HTTP with a parameter 'T' pointing to a valid certificate bundle location, the agent overrides the transport option HTTP and changes it to HTTPs for secured communication. Generally, the agent will choose the most secured communication from the available options.

The path specified in the DHCP option 43 to locate the certificate bundle file can be an absolute URL or a relative URL. If you specify a relative URL, the agent forms a full URL with the server IP address or hostname as specified in the option 43 string and uses HTTP as the file transfer protocol.

Also, to install the certificates, the agent expects the device to have an updated system clock. Because, you configure the DHCP server first, you cannot specify the current time in the DHCP server. In such a scenario, an IP address or a URL can be specified as an alternative parameter in the option 43 with the prefix 'Z', which can point the device to a NTP server. The agent synchronizes the clock on the device with the NTP server and then installs the certificates.

DHCP Option based Discovery over an IPv6 Network

Cisco Network PnP uses the DHCP Option 16 and Option 17 for an IPv6 DHCP discovery process.The Option 16 and Option 17 are vendor specific identifiers. These are used by the Cisco Network PnP agent to locate and connect to the Cisco Network PnP server. The DHCP server can be configured to insert an additional information using the vendor specific Option 17. When the DHCP server receives an Option 16 from the device with the string *cisco pnp* , and if it matches the Option 17 string, the server passes the IP address or the hostname of the Cisco PnP server to the requesting device. When the device receives the DHCPv6 response,

the Cisco Network PnP agent extracts the option from the response and identifies the IPv6 address of the Cisco PnP server. Cisco PnP agent uses this IPv6 address to communicate with the Cisco PnP server. To obtain and install the certificate, use the same process explained in the DHCP Option based Discovery over an IPv4 Network section.

The following example shows how to configure a pool (DHCPv6-pool) with the vendor-specific options:

```
ipv6 dhcp pool dhcpv6-pool
 address prefix 2003::/64 lifetime infinite infinite
 vendor-specific 9
  suboption 16 ascii "ciscopnp"
  suboption 17 ascii "5A1D;K4;B3;IFE80::2E0:81FF:FE2D:3799;J6088"
```

DNS-based Discovery

In DNS-based discovery, a DHCP server receives the domain name of the customer network. The domain name is used to create a PnP-specific, fully qualified domain name (FQDN) such as *pnpserver.<domain_name>*. In this method, the customer network resolves this URL to a valid PnP server IP address. Because, there is no mechanism to specify the certificate location, the agent locates the server certificate to initiate the HTTPs connection without manual intervention.

During the system boot up, the device acquires IP network information from a DHCP server along with the domain name. With the customer specific domain name, the Cisco PnP agent creates the following URL *pnpserver.<domain_name>* and looks for the Cisco signed certificate bundle in a default folder of the server *<domain_name>/ca/trustpool/cabundle.p7b*.

If the certificate bundle is available at the specified location, then the agent:

1. Downloads the bundle from the file server to the device.
2. Checks the signature of the downloaded bundle to ensure it has genuine Cisco signature.
3. Installs the certificates on the device.

If the certificate bundle is not available at the specified location, the PnP agent use a predefined URL,*pnpcertserver.<domain_name>* and looks for the Cisco signed certificate bundle in the default folder of the server, *<domain_name>/ca/trustpool/cabundle.p7b*.

If the certificates are available at the specified location, then the agent performs the steps specified above to install the certificates.

After the certificates are installed and the server discovery is complete, the agent initiates the HTTPs connection with the server at the URL, *pnpserver.<domain_name >* without any additional configuration. During the HTTPs handshake, the device uses the certificates that are installed from the bundle to validate the server certificate.

Also, to install the certificates, the agent expects the device to have an updated system clock. Because, you configure the DHCP server first, you cannot specify the current time in the DHCP server. In such a scenario, the agent uses a predefined URL, *pnpntpserver.<domain_name>* which needs to be mapped to a NTP sever to synchronize the clock on the device, and then installs the certificates.

However, if the certificate is not present at either URL, the Cisco PnP agent will fall back and establish the HTTP connection to the server using the created FQDN pnpserver.<domain_name>. With this workflow, the agent expects the server to use the certificate-install service to install the self-signed certificates first and then start the provisioning steps.

DNS-based Discovery over an IPv6 Network

To enable DNS-based discovery over an IPv6 network:

Procedure

Step 1 Configure the DNS server with an IPv6 option. To enable the Cisco Network PnP DNS discovery, configure the DNS server as shown in this example:

```
ip host pnpntpserver.domain.com 2001::1
ip host pnptrustpool.domain.com 2001::2
ip host pnpserver.domain.com   2001::3
```

Step 2 DHCPv6 server is discovered through DHCP bootstrap process. The following example shows how to configure the DHCP server:

```
ipv6 unicast routing
ipv6 cef

ipv6 dhcp pool test
dns-server 2001::4
domain-name example.com
```

The device sends the DHCPv6 packets to the server over an IPv6 network. After receiving the DHCPv6 packets, the DNS server information and the domain-name are returned to the device as Option 23 and Option 24 respectively.

Step 3 Configure the NTP server. The following example shows how to configure the NTP server:

```
ntp master 1
```

Note Similarly, the device NTP configuration should use the NTPv4 option.

Step 4 Host the trustpool server on an IPv6 network. Trustpool is supported only on DHCP Options T and Z. If the Option T is configured, specify the URL of the trustpool CA bundle. If the Option Z is configured, specify the NTP server IP address.

Note When the Cisco Network PnP agent attempts to download the trustpool bundle over HTTP by using an IPv6 option, the trustpool server should support HTTP over an IPv6 network. Also, the clock must be syncronized before configuring the trustpool.

Step 5 Host the Cisco Network PnP server on an IPv6 network.

Cisco Cloud Redirection over an IPv4 and IPv6 Network

Cisco Cloud Redirection service supports Cisco Network PnP zero-touch discovery. It is supported on both IPv4 and IPv6 based Cisco Cloud discovery.

Note Some of the Cisco PnP devices may have root certificate embedded in the devices. These devices will communicate with the CCO server using HTTPS from the beginning. If the device does not have the embedded certificate then the legacy behavior is initiated.

When the device boots up without any start-up configuration or authentication certificates, and if the DHCP and DNS discovery fails, the device tries to contact the Cisco Cloud server at *devicehelper.cisco.com*.

If the *devicehelper.cisco.com* is reachable, the Cisco Network PnP agent downloads the trustpool bundle and establishes a secure HTTP connection with the Cisco Cloud Redirection service. When the device tries the Cisco cloud discovery for the first time, Cisco Network PnP agent downloads the trustpool from this location *devicehelper.cisco.com/ca/trustpool* and saves it to the local flash memory. This location is shared with a Public Key Infrastructure for a trustpool installation. If the Cisco cloud discovery fails, trustpool bundle is retained in the flash memory and Cisco Network PnP checks for a copy of the trustpool bundle in the local device flash memory. If the copy is not available in the local flash memory, it retries to download the trustpool bundle from this location *devicehelper.cisco.com/ca/trustpool download*.

Cisco Network PnP agent sends a HTTPS hello message to the Cisco cloud. The Cisco Network PnP redirection service running at Cisco cloud server replies to the HTTP request. A Cisco cloud server PnP profile is created on the device as shown in this example.

```
pnp profile pnp_cco_profile
transport https host devicehelper.cisco.com port 443
```

After the Cisco cloud profile is created, the device sends a work-information message with its unique device identifier information to the Cisco cloud server. Cisco Cloud Redirection service sends a redirection non-backoff PnP request with the Cisco Network PnP server information. It can be an IPv4 address, IPv6 address, or a hostname. When the redirection is successful, the following redirection profile is configured on the device.

```
pnp profile pnp_redirection_profile
transport https ipv4 172.19.153.133 port 443
```

If the non-backoff PnP request is not received within default wait time, Cisco Network PnP discovery process continues with the next discovery mechanism.

Cisco Network PnP Discovery Over 4G Interface

Cisco Network PnP over 4G interface is available on platforms that have 4G NIMs and running Cisco IOS XE.When a device with an activated SIM card boots up, the 4G interface is activated and used for the Cisco Network PnP cloud discovery process. When a device without an activated SIM card boots up, the non-4G interfaces are preferred for the discovery process. Cisco Network PnP cloud discovery over 4G interfaces is attempted when the non-4G interfaces are not available or if the Cisco Network PnP discovery does not succeed on the non-4G interfaces. When the device has multiple 4G interfaces with active SIM cards, the Cisco Network PnP tries the cloud discovery on all the 4G interfaces one after the other until one of them succeeds.

Note To use the 4G interface for the Cisco Network PnP discovery, the 4G NIMs should have an activated SIM card on it.

Cisco Network PnP Cloud discovery over 4G interfaces works when all the 4G interfaces are activated during the device bootup by default. In the absence of a startup configuration, the device attempts to bring up the 4G

inutrafec by default and attempts Cicso PnP over cloud. After the device is redirected, the device connects to the Cisco Network PnP server and downloads the appropriate image and configuration to the device.

Note The DNS server is available as part of the 4G network and the cloud portal should be programmed to redirect the calling device to an appropriate Cisco Network PnP server for provisioning the device. Currently, Cisco Network PnP support over 4G interface uses only the IPv4 network.

Ensure that the configuration pushed through the Cisco Network PnP server contains a route to Cisco Network PnP server over the 4G interface. This can be a default route and should retain the Cisco Network PnP agent and server communication to continue to work over the 4G interface, after the provisioning is completed.

Cisco Network PnP Discovery over a Management Interface

Cisco Network PnP Agent supports discovery and four-way handshake over a management interface with a default VPN Routing/Forwarding (VRF). To send and receive the DHCP traffic over an VRF interface, you have to configure the IOS DHCP server . This feature helps the new devices to access the Cisco Network PnP features when only the management interface is active.

When the device boots up, the management interface under the default VRF is assigned an IP address though DHCP. This interface establishes a connection to a Cisco Network PnP server and the Cisco Network PnP agent on the device records this information (VRF name and source interface). This information is used for future PnP communication with the Cisco Network PnP server. In this case, the Cisco PnP profile that is created on the device will have an extra keyword **VRF** attached to it.

Cisco PnP over an EtherChannel

When you deploy an access switch by using the Cisco Network Plug and Play, the existence of LACP EtherChannels on the provisioned switch (which acts as trunk) does not allow you to configure the device. When the access device tries to connect through the provisioned switch over an L2 EtherChannel using LACP, it breaks the connectivity. Since the configuration does not exists on the access device, the access device cannot bring up the EtherChannel with the switch. This results in keeping the EtherChannel ports in suspended state and breaks the L2 connectivity. Cisco Network PnP Agent detects the presence of EtherChannels and auto-configures the EtherChannel on the device to bring up the Layer 2 connectivity automatically for the day-zero configuration.

Security Methods for Post-PnP Discovery Process

This section explains the methods provided by the Cisco PnP agent which can be, used by the Cisco PnP server to secure the client-server communication after completing the discovery process. This section includes the following topic:

• Certificate Install Service, on page 523

Certificate Install Service

The Cisco PnP agent provides a mechanism to manage SSL certificates on the device by providing the certificate-install service to the Cisco PnP server. The certificate-install service provides a simple XML to install the server's self-signed certificates or certificates signed by standard CA certificates on the device,

before initiating an HTTPs connection. The certificate-install service also provides an option to install the client SSL certificate and instruct the device to use the same SSL certificate during the next device authentication process.

SUDI-based PnP Application Level Authentication

The SSL communication ensures encryption of the data packets exchanged between the server and the device, but does not provide a solution to authenticate the device.

To ensure that the server is talking to a genuine Cisco device, the agent uses the built-in Secure Unique Device Identifier (SUDI) certificate support on the device. SUDI is a X.509 compliant device certificate burnt into the device's secured chip (ACT2) during the manufacture time. The SUDI certificate contains the device's serial number, private-public keys, and the Cisco CA signature. The agent provides the following mechanisms that can be used by the server to authenticate the device as a genuine Cisco device:

- SUDI-based Client Certificate Validation, on page 524
- SUDI-based Serial Number, on page 524

SUDI-based Client Certificate Validation

Before the agent initiates an HTTPs connection with the server, the agent checks whether the device has a built-in SUDI certificate. If the device has a certificate, then the agent sends the SUDI certificate to the client during the SSL handshake for validating. Optionally, the HTTPs server may choose to validate the device using the SUDI certificate during the SSL handshake. After validating, the HTTPs server allows the device to connect to the server. To validate the device's SUDI certificate, the server should use Cisco CA to complete the validation.

SUDI-based Serial Number

If the device is loaded with SUDI certificate, the PnP agent reads the serial number from the SUDI certificate and presents the same information as an additional tag in the work-request body for all communication with the server. To achieve this, the following optional tag is added in the work-info message, which goes out from the device in every work-request. This field is optional and does not show up for devices that does not have SUDI certificate.

There is no change in the existing UDI mechanism that is read from the chassis inventory. The agent continues to be backwards compatible by sending the chassis UDI as the primary identifier. The server can use the additionally provided SUDI-based serial number to authenticate the device and then continue to use the primary UDI. For the devices without a SUDI certificate, the agent does not send this additional SUDI-based serial number. Therefore, the server should continue with the primary UDI for authentication and further communication.

There is no mechanism available to read the SUDI-based serial number from member hardware and there is no change in how UDI is read from other members on a stack or HA unit. The agent will continue to read the UDI from all the hardware units as it does presently.

SUDI-Based Device Authentication

In SUDI-based device authentication, the agent checks whether the device has a built-in SUDI certificate at the boot-up time. If the device is loaded with the SUDI certificate, the agent provides a new PnP service, which allows the server to help the device to identify itself. The availability of this new service depends on the presence of the SUDI certificate and is listed in the agent's capability service.

Along with the above change in the capability-service, the agent adds an additional field under the hardware-info section of the device-info response, to specify and check whether the SUDI certificate is built into the device.

After, the agent initiates an HTTPs connection with the server and sends a work-request, the server should be able to use the device authentication service for a challenge request-response. The device authentication service requires a minimum of one field to generated a string by the server. Optionally, the server can send a list of encryptions and hashing methods that it can support. The agent checks whether it has the capability to use any of the listed encryption methods specified by the server, uses the encryption method and sends a notification to the server. If the agent does not have the capability to use any of the methods specified by the server, then the agent responds with an error message.

When the server sends a device authentication service request to the agent, the agent does the following:

1. Uses one of the specified encryption and hashing methods.
2. If the agent does not have capability to use one of the specified encryption and hashing methods, the agent responds with an error message.
3. Encrypts the challenge string provided by the server using the private key using the PKI APIs.
4. Sends a response back with the following:
    a. Cipher text
    b. Methods used to cipher
    c. Certificate (SUDI or client installed certificate)

After, the server receives the above response from the device, the server does the following:

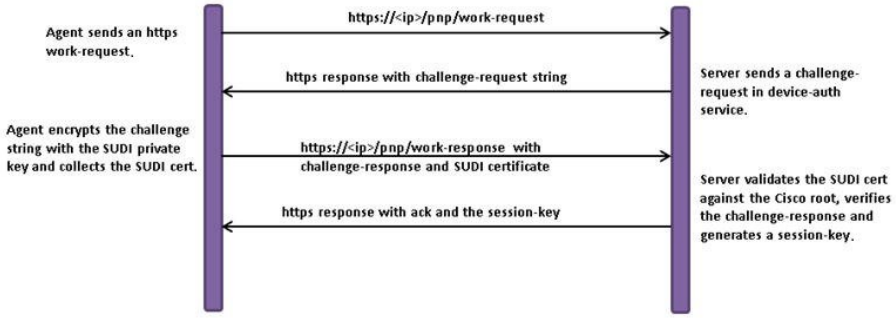
1. Verifies the SUDI or the client certificate against the Cisco or customer CA.
2. Decrypts the cipher-string using the public key that is available in the SUDI or client certificate.
3. Verifies whether the deciphered string matches the original version.
4. Generates a session key (string) and sends it back to the device as an acknowledgment.

After the agent receives the final acknowledgment from the server with the session-key, it associates the corresponding profile with the provided session-key and sends it to the server as an attribute in the root PnP section of all the subsequent messages that the agent sends.

The server validates the session-key before sending any message from the device. Optionally, the server maintains a timer for the session-keys and moves to invalid status when the timer expires. If the agent sends a message with an expired session-key, the server repeats the device authentication process and generate a new session-key before sending to the same device again. If the device sends a request without any session-key, then the server performs the device authentication process and generates a new session-key before sending to the same device.

The following figure displays the message sequence between the agent and the server to accomplish the device authentication using the SUDI certificate.

Figure 24: Message Sequence



How to Configure Cisco Network Plug and Play Agent

Configuring Cisco Network Plug and Play Agent Profile

Perform the following task to create a Cisco Network Plug and Play agent profile:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Network Plug and Play Agent Device

Perform the following task to create a Cisco Network Plug and Play agent device:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**device** {**username** *username* } {**password {0 | 7}** *password*} **Example:** `Device(config-pnp-init)# device username sjohn password 0 Tan123`	Configures the PnP agent on the device. • Establishes a username and password based authentication system. • *username*—User ID • *password*—Password that a user enters • **0**—Specifies that an unencrypted password or secret (depending on the configuration) follows. • **7**—Specifies that an encrypted (hidden) password follows.
Step 5	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Cisco Network Plug and Play Reconnect Factor

Perform the following task to configure the time to wait before attempting to reconnect a session in either fixed-interval-backoff, exponential-backoff, or random-exponential-backoff mode:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**reconnect** [*pause-time* [*exponential-backoff-factor* [**random**]]] **Example:** `Device(config-pnp-init)# reconnect 100 2 random`	Specifies the time for the PnP agent initiator profile to wait before attempting to reconnect a session. • The pause-time value is the time to wait, in seconds, before attempting to reconnect after a connection is lost. The range is from 1 to 2000000. The default is 60. • Exponential backoff factor value is the value that triggers the reconnect attempt exponentially. The range is from 2 to 9.
Step 5	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Cisco Network Plug and Play HTTP Transport Profile

Perform the following task to create a HTTP transport profile of the Cisoc Plug and Play agent manually on a device.

Both IPv4 and IPv6 addresses can be used for PnP server IP configuration. Alternately, a hostname can also be used in the configuration to connect to the PnP server.

Every profile can have one primary server and a backup server configuration. The Cisco PnP agent attempts to initiate a connection with the primary server first and if it fails, it will try the backup server. If the backup

server fails, the Cisco PnP agent will attempt to connect to the primary server again. This will continue until a connection is established with one of the servers.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**transport http host** *host-name* [**port** *port-number*] [**source** *interface-type*] **Example:** `Device(config-pnp-init)# transport http host hostname-1 port 1 source gigabitEthernet 0/0/0`	Creates a HTTP transport configuration for the PnP agent profile based on the hostname of the server on which the PnP agent is deployed. • The value of the host specifies the host name, port, and source of the server. • The value of the port-number specifies the port that is used. • The value of the interface-type specifies the interface on which the agent is connected to the server.
Step 5	**transport http ipv4** *ipv4-address* [**port** *port-number*] [**source** *interface-type*] **Example:** `Device(config-pnp-init)# transport http ipv4 10.0.1.0 port 221 source gigabitEthernet 0/0/0`	Creates a HTTP transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed.
Step 6	**transport http ipv6** *ipv6-address* [**port** *port-number*] [**source** *interface-type interface-number*] **Example:**	Creates a HTTP transport configuration for the PnP agent profile based on the IPv6 address of the server on which the PnP agent is deployed.

	Command or Action	Purpose
	`Device(config-pnp-init)# transport http ipv6 2001:DB8:1::1 port 331 source gigabitEthernet 0/0/1`	
Step 7	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Cisco Network Plug and Play HTTPS Transport Profile

Perform the following task to create a HTTP Secure (HTTPS) transport profile of the Cisco Network Plug and Play agent manually on a device.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**transport https host** *host-name* [**port** *port-number*][**source** *interface-type*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# transport https host example.com port 231 source gigabitEthernet 0/0/0 localcert abc remotecert xyz`	Creates a HTTPS transport configuration for the PnP agent profile based on the hostname of the server on which the PnP agent is deployed. • The value of *localcert* specifies the trustpoint used for client-side authentication during the transport layer security (TLS) handshake. • The value of *remotecert* specifies the trustpoint used for server certificate validation.

	Command or Action	Purpose
		Note Configure the trustpoint-nam
Step 5	**transport https ipv4** *ipv4-address* [**port** *port-number*][**source** *interface-type*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# transport https ipv4 10.0.1.0 port 221 source gigabitEthernet 0/0/0 localcert abc remotecert xyz`	Creates a HTTPS transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed.
Step 6	**transport https ipv6** *ipv6-address* [**port** *port-number*][**source** *interface-type interface-number*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# transport https ipv6 2001:DB8:1::1 port 331 source gigabitEthernet 0/0/1 localcert abc remotecert xyz`	Creates a HTTPS transport configuration for the PnP agent profile based on the IPv6 address of the server on which the PnP agent is deployed.
Step 7	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Backup Cisco Network Plug and Play Device

Perform the following task to create a backup profile and to enable or disable Cisco Network Plug and Play agent manually on a device:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**backup device** {**username** *username* } {**password {0 | 7}** *password*} **Example:** `Device(config-pnp-init)# backup device username sjohn password 0 Tan123`	Configures the PnP agent backup profile on the device. • Establishes a username and password based authentication system. • *username*-User ID • *password*-Password that a user enters • 0—Specifies that an unencrypted password or secret (depending on the configuration) follows. • 7—Specifies that a hidden password follows.
Step 5	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Backup Cisco Network Plug and Play Reconnect Factor

Perform the following task to configure backup reconnection of the Cisco Network Plug and Play (PnP) agent to the server in either fixed-interval-backoff, exponential-backoff, or random-exponential-backoff manner :

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**backup reconnect** [*pause-time* [*exponential-backoff-factor* [**random**]]] **Example:** `Device(config-pnp-init)# backup reconnect 100 2 random`	Specifies the time for the PnP agent initiator profile to wait before attempting to reconnect a session. • The pause-time value is the time to wait, in seconds, before attempting to reconnect after a connection is lost. The range is from 1 to 2000000. The default is 60. • Exponential backoff factor value is the value that triggers the reconnect attempt exponentially. The range is from 2 to 9.
Step 5	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Backup Cisco Network Plug and Play HTTP Transport Profile

Perform the following task to create a backup HTTP transport profile of the Cisco Network Plug and Play agent manually on a device.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp profile** *profile-name* **Example:**	Creates a PnP agent profile and enters the PnP profile initialization mode.

	Command or Action	Purpose
	`Device(config)# pnp profile test-profile-1`	• String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**backup transport http host** *host-name* [**port** *port-number*] [**source** *interface-type*] **Example:** `Device(config-pnp-init)# backup transport http host hostname-1 port 1 source gigabitEthernet 0/0/0`	Creates a backup HTTP transport configuration for the PnP agent profile based on the hostname of the server on which the PnP agent is deployed. • The value of the host specifies the host name, port, and source of the server. • The value of the port-number specifies the port that is used. • The value of the interface-type specifies the interface on which the agent is connected to the server.
Step 5	**backup transport http ipv4** *ipv4-address* [**port** *port-number*] [**source** *interface-type*] **Example:** `Device(config-pnp-init)# backup transport http ipv4 10.0.1.0 port 221 source gigabitEthernet 0/0/0`	Creates a backup HTTP transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed.
Step 6	**backup transport http ipv6** *ipv6-address* [**port** *port-number*] [**source** *interface-type interface-number*] **Example:** `Device(config-pnp-init)# backup transport http ipv6 2001:DB8:1::1 port 331 source gigabitEthernet 0/0/1`	Creates a backup HTTP transport configuration for the PnP agent profile based on the IPv6 address of the server on which the PnP agent is deployed.
Step 7	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Backup Cisco Network Plug and Play HTTPS Transport Profile

Perform the following task to create a backup HTTPS transport profile of the Cisco Network Plug and Play agent manually on a device.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp profile** *profile-name* **Example:** `Device(config)# pnp profile test-profile-1`	Creates a PnP agent profile and enters the PnP profile initialization mode. • String of alphanumeric characters that specify a name for the PnP agent profile. Profile names cannot be duplicated.
Step 4	**backup transport https host** *host-name* [**port** *port-number*][**source** *interface-type*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# backup transport https host example.com port 231 source gigabitEthernet 0/0/0 localcert abc remotecert xyz`	Creates a HTTPS backup transport configuration for the PnP agent profile based on the hostname of the server on which the PnP agent is deployed. • The value of *localcert* specifies the trustpoint used for client-side authentication during the transport layer security (TLS) handshake. • The value of *remotecert* specifies the trustpoint used for server certificate validation.
Step 5	**backup transport https ipv4** *ipv4-address* [**port** *port-number*][**source** *interface-type*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:** `Device(config-pnp-init)# backup transport https ipv4 10.0.1.0 port 221 source gigabitEthernet 0/0/0 localcert abc remotecert xyz`	Creates a HTTPS backup transport configuration for the PnP agent profile based on the IPv4 address of the server on which the PnP agent is deployed.
Step 6	**backup transport https ipv6** *ipv6-address* [**port** *port-number*][**source** *interface-type interface-number*][**localcert** *trustpoint-name*][**remotecert** *trustpoint-name*] **Example:**	Creates a HTTPS backup transport configuration for the PnP agent profile based on the IPv6 address of the server on which the PnP agent is deployed.

	Command or Action	Purpose
	`Device(config-pnp-init)# backup transport https ipv6 2001:DB8:1::1 port 331 source gigabitEthernet 0/0/1 localcert abc remotecert xyz`	
Step 7	**end** **Example:** `Device(config-pnp-init)# end`	Exits the PnP profile initialization mode and returns to privileged EXEC mode.

Configuring Cisco Network Plug and Play Agent Tag

Perform the following task to create a Cisco Network Plug and Play agent tag information:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pnp tag** *tag-name* **Example:** `Device(config)# pnp tag xyz`	Use the **pnp tag** command to configure the tag for the device. The neighboring Cisco devices will receive this tag information through Cisco Discovery Protocol (CDP). **Note** If there is an existing tag for the the PnP server to change the tag • String of alphanumeric characters that specify a name for the PnP agent tag.
Step 4	**end** **Example:** `Device(config)# end`	Exits the global configuration mode and returns to privileged EXEC mode.

Troubleshooting and Debugging

To run the debugging on the Cisco Network Plug and Play (server, start the server, configure the PnP profile and PnP transport. For example, start the service interaction between PnP agent and PnP server.

You can capture the debugs by executing the **debug pnp service** command. When you report a problem, collect all the pnp* files in the PnP agent flash" to the guide.

Note To collect Cisco Plug and Play server log, see the Cisco Application Policy Infrastructure Controller Enterprise Module Deployment Guide.

To troubleshoot the device, server and Cisco PnP Agent, use the following commands:

Table 38: Troubleshooting the Device, Server, and Cisco PnP Agent

Command	Description
dir nvram	Use this command to ensure tht the device does not have left over certificates.
ping vrf interface-name *<controller_ip>*	Use this command to ensure the the device can ping the controller.
show auto install trace	Use this command to view auto install trace log.
show boot	Use the show boot command to display the current value for the BOOTLDR variable.
show cdp neighbor	Use this command to display all CDP neighbors.
Show crypto pki trustpoint	Use this command to view the PKI trustpoint.
Show crypto pki trustful	Use this command to view the PKI trustful.
show ip interface brief	Use this command to view a summary of the router interfaces.
show ipv6 interface brief	Use this command to display the IPv6 interfaces.
show run | inc pnp	Uset this command to ensure that only one PnP profile installed
show pnp trace	Use this command to ensure that the device does not have start-up configuration.
show pnp tech	Use this command to view active connections for the Cisco Plug and Play IOS Agent.
show vlan	Use this command to view the VLAN information.
show ntp status	Use this command to view the NTP status.

Command	Description
show version	Use this command to ensure that the device is running the latest CCO image

Glossary

PnP Agent: An embedded agent on the device to automate deployment process

PnP Helper Applications: Applications on smart phones and personal computers that facilitate deployment. PnP helper applications are not specific to a customer or device and can be used in any deployment scenario. May be needed in limited scenarios

PnP Protocol: Protocol between the PnP agent and PnP server. This is an open protocol allowing third-party development of PnP servers

PnP Server: A central server that manages and distributes deployment information (images, configurations, files, and licenses) for the devices being deployed. Cisco Network Plug and Play server provides a north bound interface for management applications and communicates with the PnP agents on the devices using the PnP protocol.

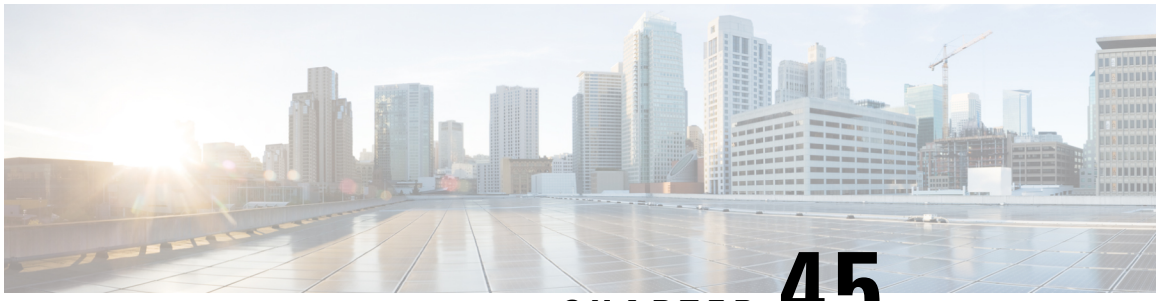
Additional References for Open Plug-n-Play Agent

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
PnP commands: Complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS PnP Command Reference
Cisco Network Plug and Play solution	Solution Guide for Cisco Network Plug and Play.
How to use the Cisco Network Plug and Play in the APIC-EM to configure Cisco network devices.	Configuration Guide for Cisco Network Plug and Play on Cisco APIC-EM.
How to deploy the APIC-EM.	Cisco Application Policy Infrastructure Controller Enterprise Module Deployment Guide.
Getting started with the APIC-EM.	Cisco APIC-EM Quick Start Guide.

MIBs

MIB	MIBs Link
• CISCO-BULK-FILE-MIB • CISCO-DATA-COLLECTION-MIB • CISCO-PROCESS-MIB • Expression-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs



CHAPTER 45

Troubleshooting

- Troubleshooting, on page 541
- Understanding Diagnostic Mode, on page 541
- Before Contacting Cisco or Your Reseller, on page 542
- show interfaces Troubleshooting Command, on page 542
- Software Upgrade Methods, on page 542
- Change the Configuration Register, on page 543
- Recovering a Lost Password, on page 545

Troubleshooting

To troubleshoot a software problem, connect a PC to the router via the console port to view status messages and enter commands. You can also access the interface remotely using Telnet if the interface is operational.

This section describes the troubleshooting scenarios.

Before troubleshooting a software problem, you must connect a PC to the router via the console port. With a connected PC, you can view status messages from the router and enter commands to troubleshoot a problem.

You can also remotely access the interface by using Telnet. The Telnet option assumes that the interface is up and running.

Understanding Diagnostic Mode

The router boots up or accesses diagnostic mode in the following scenarios:

- The IOS process or processes fail, in some scenarios. In other scenarios, the system resets when the IOS process or processes fail.
- A user-configured access policy was configured using the **transport-map** command that directs the user into the diagnostic mode.
- A send break signal (**Ctrl-C** or **Ctrl-Shift-6**) was entered while accessing the router, and the router was configured to enter diagnostic mode when a break signal was sent.

In the diagnostic mode, a subset of the commands that are available in user EXEC mode are made available to the users. Among other things, these commands can be used to:

- Inspect various states on the router, including the IOS state.
- Replace or roll back the configuration.
- Provide methods of restarting the IOS or other processes.
- Reboot hardware, such as the entire router, a module, or possibly other hardware components.
- Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

The diagnostic mode provides a more comprehensive user interface for troubleshooting than previous routers, which relied on limited access methods during failures, such as ROMMON, to diagnose and troubleshoot Cisco IOS problems. The diagnostic mode commands can work when the Cisco IOS process is not working properly. These commands are also available in privileged EXEC mode on the router when the router is working normally.

Before Contacting Cisco or Your Reseller

If you cannot locate the source of a problem, contact your local reseller for advice. Before you call, you should have the following information ready:

- Chassis type and serial number
- Maintenance agreement or warranty information
- Type of software and version number
- Date you received the hardware
- Brief description of the problem
- Brief description of the steps you have taken to isolate the problem

show interfaces Troubleshooting Command

Use the **show interfaces** command to display the status of all physical ports and logical interfaces on the router. describes messages in the command output.

The IR8340 supports the following interfaces:

- GigabitEthernet 0/0/0 and 0/0/1
- Cellular 0/4/0, Cellular 0/4/1, Cellular 0/5/0, and Cellular 0/5/1
- msata

Software Upgrade Methods

Several methods are available for upgrading software on the Cisco IR1840H Routers, including:

- Copy the new software image to flash memory over the WAN interface when the existing Cisco IOS software image is in use.

- Copy the new software image over the console port while in ROM monitor mode.
- From ROM monitor mode, boot the router from a software image that is loaded on a TFTP server. To boot the image from the TFTP server, the TFTP server must be on the same network as the router.

Change the Configuration Register

To change a configuration register, connect a PC to the console port, display the current register value, record the setting, and use the **config-register** <*value*> command to modify the break setting as needed.

Procedure to Change the Configuration Register

- Connect a PC to the CONSOLE port on the router.
- At the privileged EXEC prompt (*router_name* #), enter the **show version** command to display the existing configuration register value (shown in bold at the bottom of this output example):

```
Router#
show version
Cisco IOS XE Software, Version BLD_V177_THROTTLE_LATEST_20210827_030512_V17_7_0_91
Cisco IOS Software [Bengaluru], ir8340 Software (X86_64_LINUX_IOSD-UNIVERSALK9_IOT-M),
 Experimental Version 17.7.20210827:033430
[S2C-build-v177_throttle-289-/nobackup/mcpre/BLD-BLD_V177_THROTTLE_LATEST_20210827_030512
 154]
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Fri 27-Aug-21 15:58 by mcpre
Cisco IOS-XE software, Copyright (c) 2005-2021 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0. For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
ROM: v0.33
Router uptime is 1 week, 1 day, 23 hours, 9 minutes
Uptime for this control processor is 1 week, 1 day, 23 hours, 10 minutes
System returned to ROM by Reload Command
System image file is
"flash:ir8340-universalk9.BLD_V177_THROTTLE_LATEST_20210827_030512_V17_7_0_91.SSA.bin"
Last reload reason: Reload Command
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
Technology Package License Information:
------------------------------------------------------------------
Technology Type Technology-package Technology-package
Current Next Reboot
------------------------------------------------------------------
```

```
Smart License Perpetual network-advantage network-advantage
Smart License Subscription None None
The current crypto throughput level is 250000 kbps
Smart Licensing Status: Registration Not Applicable/Not Applicable
cisco IR8340-K9 (2RU) processor with 3717851K/6147K bytes of memory.
Processor board ID FDO2502JF0L
Router operating mode: Autonomous
1 Virtual Ethernet interface
14 Gigabit Ethernet interfaces
8 Serial interfaces
8 terminal lines
2 Cellular interfaces
32768K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
7574982K bytes of flash memory at bootflash:.
33554432K bytes of mSATA at msata:.
929715K bytes of sdcard flash at sdcard:.
30031856K bytes of USB flash at usb0:.
Configuration register is 0x0
Router#
```

- Record the setting of the configuration register.
- To enable the break setting (indicated by the value of bit 8 in the configuration register), enter the **config-register** <*value*> command from privileged EXEC mode.
  - Break enabled—Bit 8 is set to 0.
  - Break disabled (default setting)—Bit 8 is set to 1.

Configuring the Configuration Register for Autoboot

Note Altering the configuration register is only for advanced troubleshooting and should only be done with guidance from Cisco support.

The configuration register can be used to change router behavior. This includes controlling how the router boots. Set the configuration register to 0x0 to boot into ROM, by using one of the following commands:

- In Cisco IOS configuration mode, use the **config-reg** 0x0 command.
- From the ROMMON prompt, use the **confreg** 0x0 command.

Note Setting the configuration register to 0x2102 will set the router to autoboot the Cisco IOS XE software.

Reset the Router

To reset the router, follow these steps:

Procedure

Step 1 If the break is disabled, turn off the router, wait for 5 seconds and turn the router back on. Within 60 seconds push the Reset button.

The terminal displays the Rommon prompt.

Example:

```
rommon 1>
```

Step 2 Enter **confreg 0x2142** to ignore the running config.

Example:

```
rommon 2> confreg 0x142
```

Step 3 Sync the configuration changes with the **sync** command.

Example:

```
rommon 3>sync
```

Step 4 Reset the router to apply confreg. The router will reload with the reset.

Example:

```
rommon 4>reset

resetting...
```

Step 5 Verify that the correct confreg 0x2142 was applied, and enter**n** when asked if you want to change the configuration.

Example:

```
rommon 1> confreg
Configuration Summary
(Virtual Configuration Register: 0x2142)
enabled are:
[ 0 ] console baud: 9600
boot:...... image specified by the boot system commands
do you wish to change the configuration? y/n [n]: n
```

Step 6 Boot the image with the confreg 0x2142.

Example:

```
rommon 2> boot
flash:ir8340-universalk9.BLD_V177_THROTTLE_LATEST_20210827_030512_V17_7_0_91.SSA.bin
```

Recovering a Lost Password

To recover a lost password, follow the documented steps and refer to Reset the Router, on page 544 for additional details.

Password Recovery Procedure

- Reset the router.
- Change the confreg to 0x2142.
- Boot the router with confreg 0x2142 from Rommon.
- If you used the reset button, add the license:

```
Router#config term
      Router#license smart reservation
```

Note Recovering a lost password is only possible when you are connected to the router through the console port. These procedures cannot be performed through a Telnet session.

Reset the Configuration Register Value

To reset the configuration register value after you have recovered or reconfigured a password, follow these steps:

Procedure

Step 1 Enter the **configure terminal** command to enter global configuration mode:

Example:

```
Router# configure terminal
```

Step 2 Enter the **configure register** command and the original configuration register value that you recorded.

Example:

```
Router(config)# config-reg
value
```

Step 3 Enter **exit** to exit configuration mode:

Example:

```
Router(config)# exit
```

Note To return to the configuration being used before you recovered the lost enable password, do not save the configuration changes before rebooting the router.

Step 4 Reboot the router, and enter the recovered password.

Configuring a console port transport map

This task describes how to configure a transport map for a console port interface on the router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**transport-map type console** *transport-map-name* **Example:** `Router(config)# transport-map type console consolehandler`	Creates and names a transport map for handling console connections, and enters transport map configuration mode.
Step 4	**connection wait** [**allow** [**interruptible**] | **none** [**disconnect**]] **Example:** `Router(config-tmap)# connection wait none`	Specifies how a console connection will be handled using this transport map. • **allow interruptible**—The console connection waits for a Cisco IOS VTY line to become available, and also allows users to enter diagnostic mode by interrupting a console connection that is waiting for a Cisco IOS VTY line to become available. This is the default setting. **Note** Users can interrupt a waitin • **none**—The console connection immediately enters diagnostic mode.

Step 5	(Optional) **banner** [**diagnostic** | **wait**] *banner-message*
	Example:
	`Router(config-tmap)# banner diagnostic X`
	`Enter TEXT message. End with the`
		(Optional) Creates a banner message that will be seen by users entering diagnostic mode or waiting for the Cisco IOS VTY line because of the console transport map configuration.
		• **diagnostic**—Creates a banner message seen by users directed to diagnostic mode

	Command or Action	Purpose

	`character 'X'.`
	`--Welcome to Diagnostic Mode--`
	`X`
	`Router(config-tmap)#`
		because of the console transport map configuration.
		
		Note Users can interrupt a waiting c
		
		• **wait**—Creates a banner message seen by users waiting for Cisco IOS VTY to become available.
		• *banner-message*—Banner message, which begins and ends with the same delimiting character.

Step 6	**exit** **Example:** `Router(config-tmap)# exit`	Exits transport map configuration mode to re-enter global configuration mode.
Step 7	**transport type console** *console-line-number* **input** *transport-map-name* **Example:** `Router(config)# transport type console 0 input consolehandler`	Applies the settings defined in the transport map to the console interface. The *transport-map-name* for this command must match the *transport-map-name* defined in the **transport-map type console** command.

Examples

The following example shows how to create a transport map to set console port access policies and attach to console port 0:

```
Router(config)# transport-map type console consolehandler
Router(config-tmap)# connection wait allow interruptible
Router(config-tmap)# banner diagnostic X
Enter TEXT message. End with the character 'X'.
--Welcome to diagnostic mode--
X
Router(config-tmap)# banner wait X
Enter TEXT message. End with the character 'X'.
Waiting for IOS vty line
X
Router(config-tmap)# exit
Router(config)# transport type console 0 input consolehandler
```

Viewing Console Port, SSH, and Telnet Handling Configurations

Use the following commands to view console port, SSH, and Telnet handling configurations:

- **show transport-map**
- **show platform software configuration access policy**

Use the **show transport-map** command to view transport map configurations.

show transport-map [**all** | **name** *transport-map-name* | **type** [**console**]]

This command can be used either in user EXEC mode or privileged EXEC mode.

Example

The following example shows transport maps that are configured on the router: console port (`consolehandler`):

```
Router# show transport-map all
Transport Map:
Name: consolehandler Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type console
Transport Map:
Name: consolehandler


REVIEW DRAFT - CISCO CONFIDENTIAL

Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type persistent ssh
Transport Map:
Name: consolehandler Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode
```

Use the **show platform software configuration access policy** command to view the current configurations for handling the incoming console port, SSH, and Telnet connections. The output of this command provides the current wait policy for each type of connection (Telnet, SSH, and console), as well as information on the currently configured banners.

Unlike the **show transport-map** command, the **show platform software configuration access policy** command is available in diagnostic mode so that it can be entered in scenarios where you need transport map configuration information, but cannot access the Cisco IOS CLI.

Example

The following example shows the **show platform software configuration access policy** command.

```
Router# show platform software configuration access policy
The current access-policies

Method : telnet
Rule : wait with interrupt Shell banner:
Welcome to Diagnostic Mode

Wait banner :
Waiting for IOS Process

Method : ssh Rule : wait Shell banner: Wait banner :

Method : console
Rule : wait with interrupt Shell banner:
Wait banner :
```

Using the factory reset Commands

The **factory reset** commands are used to remove all the customer specific data on a router/switch that has been added. The data can be configuration, log files, boot variables, core files, and so on.

The **factory-reset all** command erases the bootflash, nvram, rommon variables, licenses, and logs.

Caution Use of the factory reset command should not be done lightly. All customer configurations will be ddeleted and the platform will boot up as if new from the factory.

Note factory-reset all does not work if IOS-XE is running in controller mode. Please refer to SDWAN configuration information.

```
Router#factory-reset all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
*Enter*

*May 12 09:55:45.831: %SYS-5-RELOAD: Reload requested by Exec. Reload Reason: Factory Reset.

***Return to ROMMON Prompt
```

Boot Sequence after Factory Reset

Booting the image:

- The bootloader attempts to boot “golden.bin” from the bootflash: partition
- If no “golden.bin” is present, then boot the first image.

Loading the configuration:

- IOS looks for “golden.cfg” file on nvram: partition and applies it upon booting.
- If no “golden.cfg” is present on nvram: then IOS looks for “golden.cfg” file on bootflash: partition and applies it upon booting.
- If no “golden.cfg” is present on bootflash: then configurations are erased and Software Configuration dialog is used.