

j – k

- java-trustpoint(Deprecated), on page 2
- join-failover-group, on page 4
- jumbo-frame reservation, on page 6
- kcd-server, on page 8
- keepout, on page 10
- kerberos-realm, on page 12
- kernel crash-dump, on page 14
- key (aaa-server host), on page 16
- key (cluster group), on page 18
- key chain, on page 20
- key config-key password-encryption, on page 22
- key-hash, on page 24
- keypair, on page 26
- keysize, on page 28
- keysize server, on page 30
- key-string, on page 32
- kill, on page 34

java-trustpoint(Deprecated)

To configure the WebVPN Java object signing facility to use a PKCS12 certificate and keying material from a specified trustpoint location, use the **java-trustpoint** command in webvpn configuration mode.To remove a trustpoint for Java object signing, use the **no** form of this command.

java-trustpoint*trustpoint*
no java-trustpoint

Syntax Description

trustpoint	Specifies the trustpoint location configured by the **crypto ca import** command.

Command Default

By default, a trustpoint for Java object signing is set to none.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Webvpn configuration	• Yes	—	• Yes	—	—

Command History

Release	Modification
7.1(2)	This command was added.
9.17(1)	This command was deprecated due to support removal for web VPN.

Usage Guidelines

A trustpoint is a representation of a certificate authority (CA) or identity key pair. For the **java-trustpoint** command, the given trustpoint must contain the X.509 certificate of the application signing entity, the RSA private key corresponding to that certificate, and a certificate authority chain extending up to a root CA. This is typically achieved by using the **crypto ca import** command to import a PKCS12 formatted bundle. You can obtain a PKCS12 bundle from a trusted CA authority or you can manually create one from an existing X.509 certificate and an RSA private key using open source tools such as openssl.



Note An uploaded certificate cannot be used to sign Java objects that are embedded with packages (for example, the CSD package).

Examples

The following example first configures a new trustpoint, then configures it for WebVPN Java object signing:

```
ciscoasa(config)# crypto ca import mytrustpoint pkcs12 mypassphrase
```

```
Enter the base 64 encoded PKCS12.
End with the word "quit" on a line by itself.
[ PKCS12 data omitted ]
quit
INFO: Import PKCS12 operation completed successfully.
ciscoasa(config)#
```

The following example configures the new trustpoint for signing WebVPN Java objects:

```
ciscoasa(config)# webvpn
ciscoasa(config)# java-trustpoint mytrustpoint
ciscoasa(config)#
```

Related Commands

Command	Description
crypto ca import	Imports the certificate and key pair for a trustpoint using PKCS12 data.

join-failover-group

To assign a context to a failover group, use the **join-failover-group** command in context configuration mode. To restore the default setting, use the **no** form of this command.

join-failover-group*group_num*
no join-failover-group *group_num*

Syntax Description

group_num	Specifies the failover group number.

Command Default

Failover group 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Context configuration	• Yes	• Yes	—	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added.

Usage Guidelines

The admin context is always assigned to failover group 1. You can use the **show context detail** command to display the failover group and context association.

Before you can assign a context to a failover group, you must create the failover group with the **failover group** command in the system context. Enter this command on the unit where the context is in the active state. By default, unassigned contexts are members of failover group 1, so if the context had not been previously assigned to a failover group, you should enter this command on the unit that has failover group 1 in the active state.

You must remove all contexts from a failover group, using the **no join-failover-group** command, before you can remove a failover group from the system.

Examples

The following example assigns a context named ctx1 to failover group 2:

```
ciscoasa(config)# context ctx1
ciscoasa(config-context)# join-failover-group 2
ciscoasa(config-context)# exit
```

Related Commands

Command	Description
context	Enters context configuration mode for the specified context.

Command	Description
failover group	Defines a failover group for Active/Active failover.
show context detail	Displays context detail information, including name, class, interfaces, failover group association, and configuration file URL.

jumbo-frame reservation

To enable jumbo frames for supported models, use the **jumbo-frame reservation** command in global configuration mode. To disable jumbo frames, use the **no** form of this command.

Note Changes in this setting require you to reboot the ASA.

jumbo-frame reservation
no jumbo-frame reservation

Syntax Description

This command has no arguments or keywords.

Command Default

Jumbo frame reservation is disabled by default on ASA hardware, ASA Virtual, and ISA 3000.

Jumbo frames are supported by default on other models.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	• Yes	• Yes	• Yes	—	• Yes

Command History

Release	Modification
8.1(1)	This command was added for the ASA 5580.
8.2(5)/8.4(1)	Support for the ASA 5585-X was added.
8.6(1)	Support for the ASA 5512-X through ASA 5555-X was added.
9.3(2)	Support for the ASA 5506-X was added.
9.3(3)	Support for the ASA 5508-X and 5516-X was added.

Usage Guidelines

This procedure only applies to ASA hardware models, the ISA 3000 and the ASA Virtual. Other models support jumbo frames by default.

Jumbo frames are not supported on the ASAv5 and ASAv10 with less than 8GB RAM.

A jumbo frame is an Ethernet packet larger than the standard maximum of 1518 bytes (including Layer 2 header and VLAN tagging, 18 bytes), up to 9216 bytes. Note that the **mtu** command specifies the *payload* value only, so for a 9216 byte jumbo frame, set the MTU to be 9198 (9216-18 bytes for the header)

Jumbo frame support requires extra memory, which might limit the maximum use of other features, such as access lists.

Jumbo frames are not supported on the Management *n* /*n* interface.

Be sure to set the MTU for each interface that needs to transmit jumbo frames to a higher value than the default 1500; for example, set the value to 9198 using the **mtu** command. For the ASASM, you do not need to set the **jumbo-frame** reservation command; it supports jumbo frames by default. Just set the MTU to the desired value.

Also, be sure to configure the MSS (maximum segment size) value for TCP when using jumbo frames. The MSS should be 120 bytes less than the MTU. For example, if you configure the MTU to be 9000, then the MSS should be configured to 8880. You can configure the MSS with the **sysopt connection tcpmss** command.

Both the primary and the secondary units require a reboot so that the failover pair supports jumbo frames. To avoid downtime, do the following:

- Issue the command on the active unit.
- Save the running configuration on the active unit.
- Reboot the primary and secondary units, one at a time.

Examples

The following example enables jumbo frame reservation, saves the configuration, and reloads the ASA:

```
ciscoasa(config)# jumbo-frame reservation
WARNING: this command will take effect after the running-config is saved
and the system has been rebooted. Command accepted.
ciscoasa(config)# write memory
Building configuration...
Cryptochecksum: 718e3706 4edb11ea 69af58d0 0a6b7cb5
70291 bytes copied in 3.710 secs (23430 bytes/sec)
[OK]
ciscoasa(config)# reload
Proceed with reload? [confirm] Y
```

Related Commands

Command	Description
mtu	Specifies the maximum transmission unit for an interface.
show jumbo-frame reservation	Shows the current configuration of the **jumbo-frame reservation** command.

kcd-server

To configure Kerberos Constrained Delegation (KCD) for clientless SSL remote access VPN, use the **kcd-server** command in webvpn configuration mode. To disable KCD, use the **no** form of this command.

kcd-server *aaa-server-group_name* **username** *user_id* **password** *password* [**validate-server-certificate**]
no kcd-server

Syntax Description

username	Specifies the Active Directory user with administrator or service level privileges to add devices to the domain.
password	Specifies the password for the user.
validate-server-certificate	(Optional.) Instructs the ASA to validate the server certificate and thus the identity of the server when joining the domain. If you omit this option, the system assumes the domain controller is valid.

Command Default

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Webvpn configuration	• Yes	—	• Yes	—	—

Command History

Release	Modification
8.4(1)	This command was added.
9.15(1)	The **validate-server-certificate** keyword was added.

Usage Guidelines

Use the **kcd-server** command in webvpn configuration mode to allow the ASA to join an Active Directory domain. The domain controller name and realm are specified in the **aaa-server-groupname** command. The AAA server group has to be a Kerberos server type. The **username** and **password** options do not correspond to a user with Administrator privileges, but they should correspond to a user with service-level privileges on the domain controller. To view the existing configuration, use the **show webvpn kcd** command.

Kerberos Constrained Delegation, or KCD, in the ASA environment provides clientless SSL remote access VPN users Single Sign-on (SSO) access to all web services that are protected by Kerberos. The ASA maintains a credential on behalf of the user (a service ticket) and uses this ticket to authenticate the user to the services.

In order for the **kcd-server** command to function, the ASA must establish a trust relationship between the *source* domain (the domain where the ASA resides) and the *target* or *resource* domain (the domain where

the web services reside). The ASA crosses the certification path from the source to the destination domain and acquires the necessary tickets on behalf of the remote access user to access the services.

This path is called cross-realm authentication. During each phase of cross-realm authentication, the ASA relies on the credentials at a particular domain and the trust relationship with the subsequent domain.

The KCD configuration also requires that you configure the domain controller as a DNS server (for example, in the DefaultDNS group), and enable DNS lookup on the interface through which the domain controller can be reached.

Examples

The following is a configuration example of KCD, where the Domain Controller is 10.1.1.10 (reachable via inside interface) and the domain name is PRIVATE.NET. Additionally, the Service Account username and password on the domain controller is dcuser and dcuser123! .

```
----Enable a DNS lookup by configuring the DNS server and Domain name --------------
ciscoasa
(config)#
dns domain-lookup inside
ciscoasa
(config)#
dns server-group DefaultDNS
ciscoasa
(config-dns-server-group)#
name-server 10.1.1.10
ciscoasa
(config-dns-server-group)#
domain-name
private.net
----Configure the AAA server group with Server and Realm-------------------------------
ciscoasa
(config)#
aaa-server KerberosGroup protocol Kerberos
ciscoasa
(config-asa-server-group)#
aaa-server KerberosGroup (inside) host 10.1.1.10
ciscoasa
(config-asa-server-group)#
kerberos-realm PRIVATE.NET
----Enable KCD-----------------------------
ciscoasa
(config)#
webvpn
ciscoasa
(config-webvpn)#
kcd-server KerberosGroup username dcuser password dcuser123!
validate-server-certificate
```

Related Commands

Command	Description
aaa-server	Enters aaa-server configuration mode, so you can configure AAA server parameters.
aaa-server host	Enters aaa-server host configuration mode, so you can configure AAA server parameters that are host-specific.
show aaa kerberos	Displays Kerberos tickets.
show webvpn kcd	Displays the KCD configuration.

keepout

To present an administrator-defined message rather than a login page for new user sessions (when the ASA undergoes a maintenance or troubleshooting period), use the **keepout** command in webvpn configuration mode. To remove a previously set keepout page, use the **no** version of the command.

keepout
no keepout *string*

Syntax Description

string	An alphanumeric string in double quotation marks.

Command Default

No keepout page.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Webvpn configuration (global)	Yes	—	Yes	—	—

Command History

Release	Modification
8.0(2)	This command was added.

Usage Guidelines

When this command is enabled, the clientless WebVPN portal page becomes unavailable. You receive an administrator-defined message stating the unavailability of the portal rather than a login page for the portal. Use the **keepout** command to disable clientless access, but still allow AnyConnect access. You can also use this command to indicate portal unavailability when maintenance is occurring.

Note

- If HostScan is installed, the keepout feature does not stop the ASA from opening pages like Cisco Secure Desktop portal. To avoid the Cisco Secure Desktop port, HostScan needs to be uninstalled.
- **keepout** command is supported only in the global webvpn configuration context and is not supported within group-policy attributes or username attributes configuration modes. Do not include this command in a group-policy FlexConfig object.

Examples

The following example shows how to configure a keepout page:

```
ciscoasa
```

```
(config)#
webvpn
ciscoasa
(config-webvpn)#
 keepout "The system is unavailable until 7:00 a.m. EST."
ciscoasa(config-webvpn)#
```

Related Commands

Command	Description
webvpn	Enters webvpn configuration mode, which lets you configure attributes for clientless SSL VPN connections.

kerberos-realm

To specify the realm name for this Kerberos server, use the **kerberos-realm** command in aaa-server host configuration mode. To remove the realm name, use the **no** form of this command:

kerberos-realm*string*
no kerberos-realm

Syntax Description

string A case-sensitive, alphanumeric string, up to 64 characters long. Spaces are not permitted in the string.

Note
Kerberos realm names use numbers and upper case letters only. Although the ASA accepts lower case letters in the *string* argument, it does not translate lower case letters to upper case letters. Be sure to use upper case letters only.

Command Default

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Aaa-server host configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added.

Usage Guidelines

This command is valid only for Kerberos servers.

The value of the *string* argument should match the output of the Microsoft Windows **set USERDNSDOMAIN** command when it is run on the Windows 2000 Active Directory server for the Kerberos realm. In the following example, EXAMPLE.COM is the Kerberos realm name:

```
C:\>set USERDNSDOMAIN
USERDNSDOMAIN=EXAMPLE.COM
```

The *string* argument must use numbers and upper case letters only. The **kerberos-realm** command is case sensitive, and the ASA does not translate lower case letters to upper case letters.

Examples

The following sequence shows the **kerberos-realm** command to set the kerberos realm to "EXAMPLE.COM" in the context of configuring a AAA server host:

```
ciscoasa
(config)# aaa-server svrgrp1 protocol kerberos
```

```
ciscoasa
(config-aaa-server-group)# aaa-server svrgrp1 host 1.2.3.4
ciscoasa
(config-aaa-server-host)# timeout 9
ciscoasa
(config-aaa-server-host)# retry 7
ciscoasa
(config-aaa-server-host)# kerberos-realm EXAMPLE.COM
ciscoasa
(config-aaa-server-host)#
exit
ciscoasa
(config)#
```

Related Commands

Command	Description
aaa-server host	Enter AAA server host configuration submode so you can configure AAA server parameters that are host-specific.
clear configure aaa-server	Remove all AAA command statements from the configuration.
show running-config aaa-server	Displays AAA server statistics for all AAA servers, for a particular server group, for a particular server within a particular group, or for a particular protocol.

kernel crash-dump

To enable the secondary crash dump kernel, enter the **kernel crash-dump** command in global configuration mode. To disable the crash dump kernel configuration, enter the **no kernel crash-dump** command.

kernel crash-dump
no kernel crash-dump

Command Default

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	Yes	Yes	Yes	—	Yes

Command History

Release	Modification
9.24(1)	This command was added.

Usage Guidelines

A Secure Firewall product that experiences unexpected Linux kernel crashes or silent reboots does not provide information for debugging the root cause. To initiate a crash dump and thereby effectively capture, store, and facilitate the analysis of Linux kernel crash incidents on the system, use the **kernel crash-dump** command to enable the administrator.

Examples

The following is a sample output of the **kernel crash-dump** command:

```
ciscoasa(config)# kernel crash-dump
Configuring the secondary crash dump kernel requires rebooting the system to apply any
operational state changes.
[...]
-
-
-
```

Examples

The following is a sample output of the **no kernel crash-dump** command:

```
ciscoasa(config)# no kernel crash-dump

Configuring the secondary crash dump kernel requires rebooting the system to apply any
operational state changes.
[...]
-
-
```

–
–

Related Commands

Command	Description
show kernel crash-dump	Displays the status of the secondary crash dump kernel.
crashinfo force kernel-dump	Initiates primary kernel crash dump.

key (aaa-server host)

To specify the server secret value used to authenticate the NAS to the AAA server, use the **key** command in aaa-server host configuration mode. The aaa-server host configuration mode is accessible from aaa-server protocol configuration mode. To remove the key, use the **no** form of this command.

key [**0** | **8**] *key*
no key

Syntax Description

key An alphanumeric keyword, which can be up to 127 characters long. You can optionally precede the key with a number to indicate encryption:

- 0 means the key is not encrypted. This is the default.
- 8 means the key is an AES encrypted base64 hash.

Command Default

No default behaviors or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Aaa-server host configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added.

Usage Guidelines

The *key* value is a case-sensitive, alphanumeric keyword of up to 127 characters, which is the same value as the key on the TACACS+ server. Any characters over 127 are ignored. The key is used between the client and the server for encrypting data between them. The key must be the same on both the client and server systems.The key cannot contain spaces, but other special characters are allowed. The key (server secret) value authenticates the ASA to the AAA server.

This command is valid only for RADIUS and TACACS+ servers.

Examples

The following example configures a TACACS+ AAA server named “srvgrp1” on host “1.2.3.4,” sets a timeout of 9 seconds, sets a retry-interval of 7 seconds, and configures the key as “myexclusivemumblekey.”

```
ciscoasa
(config)# aaa-server svrgrp1 protocol tacacs+
ciscoasa
```

```
(config-aaa-server-group)# aaa-server svrgrp1 host 1.2.3.4
ciscoasa
(config-aaa-server-host)# timeout 9
ciscoasa
(config-aaa-server-host)# retry-interval 7
ciscoasa
(config-aaa-server-host)# key myexclusivemumblekey
```

Related Commands

Command	Description
aaa-server host	Enters aaa-server host configuration mode, so that you can configure host-specific AAA server parameters.
clear configure aaa-server	Removes all AAA command statements from the configuration.
show running-config aaa-server	Displays the AAA server configuration.

key (cluster group)

To set an authentication key for control traffic on the cluster control link, use the **key** command in cluster group configuration mode. To remove the key, use the **no** form of this command.

key *shared_secret*
no key [*shared_secret*]

Syntax Description

shared_secret	Sets the shared secret to an ASCII string from 1 to 63 characters. The shared secret is used to generate the key.

Command Default

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	• Yes	• Yes	• Yes	—	• Yes

Command History

Release	Modification
9.0(1)	This command was added.

Usage Guidelines

This command does not affect datapath traffic, including connection state update and forwarded packets, which are always sent in the clear.

Examples

The following example sets a shared secret:

```
ciscoasa(config)# cluster group cluster1
ciscoasa(cfg-cluster)# key chuntheunavoidable
```

Related Commands

Command	Description
clacp system-mac	When using spanned EtherChannels, the ASA uses cLACP to negotiate the EtherChannel with the neighbor switch.
cluster group	Names the cluster and enters cluster configuration mode.
cluster-interface	Specifies the cluster control link interface.
cluster interface-mode	Sets the cluster interface mode.

Command	Description
conn-rebalance	Enables connection rebalancing.
console-replicate	Enables console replication from slave units to the master unit.
enable (cluster group)	Enables clustering.
health-check	Enables the cluster health check feature, which includes unit health monitoring and interface health monitoring.
key	Sets an authentication key for control traffic on the cluster control link.
local-unit	Names the cluster member.
mtu cluster-interface	Specifies the maximum transmission unit for the cluster control link interface.
priority (cluster group)	Sets the priority of this unit for master unit elections.

key chain

To configure rotating keys for authenticating IGP peers, use the **key chain** command in the global configuration mode. To remove the configuration, use the **no** form of the command.

key chain *key-chain-name* **key** *key-id* **key-string** { **0** | **8** } *key-string-text* **cryptographic-algorithm md5** [**accept-lifetime** [*local* | *start-time*] [**duration** { *duration value* | *infinite* | *end-time* }]

no key chain *key-chain-name* **key** *key-id* **key-string** { **0** | **8** } *key-string-text* **cryptographic-algorithm md5** [**accept-lifetime** [*local* | *start-time*] [**duration** { *duration value* | *infinite* | *end-time* }]

Syntax Description

key-chain-name	The name for the key chain to be configured for OSPFv2 authentication.
key-id	The unique identifier in the key chain; the valid range being 1 to 255.
0	Specifies an unencrypted password will follow.
8	Specifies an encrypted password will follow.
key-string-text	The password for the key id. The string can be a plain text or an encrypted value.
md5	The supported cryptographic algorithm. Only md5 is supported.
accept-lifetime	(Optional) The time interval within which the device accepts the key during key exchange with another device.
send-lifetime	(Optional) The time interval within which the device sends the key during key exchange with another device.

Command Default

The accept or send lifetimes, if not specified, is always active by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	• Yes	• Yes	• Yes	• Yes	• No

Command History

Release	Modification
9.12(1)	This command was added.

Usage Guidelines

Use **key chain** command to configure the key chain to be used in OSPFv2 authentication for an interface. You must enter the **key id**, **key string,** and the **cryptographic-algorithm** command. Enter **accept and send lifetimes** to schedule the rotation of keys. The lifetime variables helps to handle secured key rollover. The

device uses the lifetimes of keys to determine which keys in a key chain are active at any given point in time. When the lifetimes are not specified, the key chain authentication functions similar to that of MD5 authentication without time lines. Use the **no key chain** to remove the configuration of the key chain.

Examples

The following example shows the key chain configuration commands:

```
ciscoasa(config)# key chain CHAIN1
ciscoasa(config-keychain)# key 1
ciscoasa(config-keychain-key)# key-string 0 CHAIN1KEY1STRING
ciscoasa(config-keychain-key)# cryptographic-algorithm md5
ciscoasa(config-keychain-key)# accept-lifetime 11:22:33 1 SEP 2018 infinite

ciscoasa(config-keychain-key)#
```

Examples

The following example provides the output of the running key chain configuration:

```
ciscoasa# show running key chain
key chain CHAIN2
 key 1
  key-string KEY1CHAIN2
  cryptographic-algorithm md5
 key 2
  accept-lifetime 11:00:12 Sep 1 2018 11:12:12 Sep 1 2018
  cryptographic-algorithm md5
key chain CHAIN1
 key 1
  key-string CHAIN1KEY1STRING
  accept-lifetime 11:22:33 Sep 1 2018 duration -1
  cryptographic-algorithm md5
ciscoasa# show runing key chain CHAIN1
key chain CHAIN1
 key 1
  key-string CHAIN1KEY1STRING
  accept-lifetime 11:22:33 Sep 1 2018 duration -1
  cryptographic-algorithm md5
ciscoasa#
```

Related Commands

Command	Description
show key chain	Displays the configured key chains
show running key chain	Displays the key chain details that is currently active
clear configure key chain	Removes the key chains configured

key config-key password-encryption

To set the master passphrase used for generating the encryption key to securely store plain text passwords in encrypted format, use the **key config-key password-encryption** command in global configuration mode. To decrypt passwords encrypted with the passphrase, use the no form of this command.

key config-key password-encryption *passphrase* [*old_passphrase*]
no key config-key password-encryption *passphrase*

Syntax Description

passphrase	The passphrase must be between 8 and 128 character long. All characters except the backspace and double quote will be accepted for the passphrase. If you do not enter the passphrase in the command, you are prompted for it. Use the interactive prompts to enter passwords to avoid having the passwords logged in the command history buffer.
old_passphrase	If you are changing the passphrase, enter the old passphrase.

Command Default

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	• Yes	• Yes	• Yes	—	• Yes

Command History

Release	Modification
8.3(1)	This command was added.

Usage Guidelines

Features that use the master passphrase include the following:

- OSPF
- EIGRP
- VPN load balancing
- VPN (remote access and site-to-site)
- Failover
- AAA servers
- Logging
- Shared licenses

You must enter both the **key config-key password-encrypt** command and the **password encryption aes** command in any order to trigger password encryption. Enter **write memory** to save the encrypted passwords to the startup configuration. Otherwise, passwords in the startup configuration may still be visible. In multiple context mode, use **write memory all** in the system execution space to save all context configurations.

This command will only be accepted in a secure session, for example by console, SSH, or ASDM via HTTPS.

Use the **no key config-key password-encrypt** command with caution, because it changes the encrypted passwords into plain text passwords. You might use the **no** form of this command when downgrading to a software version that does not support password encryption.

If failover is enabled but no failover shared key is set, an error message appears if you change the master passphrase, informing you that you must enter a failover shared key to protect the master passphrase changes from being sent as plain text.

Enabling or changing password encryption in Active/Standby failover causes a **write standby**, which replicates the active configuration to the standby unit. Without this replication, the encrypted passwords on the standby unit will differ even though they use the same passphrase; configuration replication ensures that the configurations are the same. For Active/Active failover, you must manually enter **write standby**. A **write standby** can cause traffic interruption in Active/Active mode, because the configuration is cleared on the secondary unit before the new configuration is synced. You should make all contexts active on the primary ASA using the **failover active group 1** and **failover active group 2** commands, enter **write standby**, and then restore the group 2 contexts to the secondary unit using the **no failover active group 2** command.

The write erase command when followed by the reload command will remove the master passphrase and all configuration if it is lost.

Examples

The following example sets the passphrase used for generating the encryption key, and enables password encryption:

```
ciscoasa
(config)#
 key config-key password-encryption
Old key: bumblebee
New key: haverford
Confirm key: haverford
ciscoasa(config)# password encryption aes
ciscoasa(config)# write memory
```

Related Commands

Command	Description
password encryption aes	Enables password encryption.
write erase	Removes the master passphrase if it is lost when followed by the reload command.

key-hash

To manually add a hashed SSH host key for a server for the on-board Secure Copy (SCP) client, use the **key-hash** command in server configuration mode. You can access the server configuration mode by first entering the **ssh pubkey-chain** command. To remove the key, use the **no** form of this command.

key-hash { **md5** | **sha256** } *fingerprint*
no key-hash { **md5** | **sha256** } *fingerprint*

Syntax Description

fingerprint	Enters the hashed key.
{md5 | sha256}	Sets the type of hash used, either MD5 or SHA-256. The ASA always uses SHA-256 in its configuration.

Command Default

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Server configuration	• Yes	• Yes	• Yes	—	• Yes

Command History

Release	Modification
9.1(5)	This command was added.

Usage Guidelines

You can copy files to and from the ASA using the on-board SCP client. The ASA stores the SSH host key for each SCP server to which it connects. You can manually add or delete servers and their keys from the ASA database if desired.

For each server, you can specify the **key-string** (public key) or **key-hash** (hashed value) of the SSH host. The **key-hash** enters the already hashed key (using an MD5 or SHA-256 key); for example, a key that you copied from **show** command output.

Examples

The following example adds an already hashed host key for the server at 10.86.94.170:

```
ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.86.94.170
ciscoasa(config-ssh-pubkey-server)# key-hash sha256
65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19
```

Related Commands

Command	Description
copy	Copies a file to or from the ASA.
key-hash	Enters a hashed SSH host key.
key-string	Enters a public SSH host key.
ssh pubkey-chain	Manually adds or deletes servers and their keys from the ASA database.
ssh stricthostkeycheck	Enables SSH host key checking for the on-board Secure Copy (SCP) client.

keypair

To specify the key pair whose public key is to be certified, use the **keypair** command in crypto ca trustpoint configuration mode. To restore the default setting, use the **no** form of the command.

[no] keypair *name* | [**rsa modulus** | **2048** | **4096**] | [**ecdsa elliptic-curve 256** | **384** | **521**] | [**eddsa edwards-curve Ed25519**]

Syntax Description

name	Specifies the name of the key pair for non-CMP enrollments.
rsa	Generate RSA keys for any CMP manual and automatic enrollments.
ecdsa	Generate ECDSA keys for any CMP manual and automatic enrollments.
eddsa	Generate EdDSA keys for any CMP manual and automatic enrollments.

Command Default

The default setting is not to include the key pair.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Crypto ca trustpoint configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added.
9.7(1)	New EDCSA and RSA keypairs were added.
9.16(1)	• Support to certificates with RSA key sizes smaller than 2048 bits was removed. Hence the rsa modulus options were modified to display 2048 bits and bigger values. • New EdDSA keypair was added.

Examples

The following example enters crypto ca trustpoint configuration mode for the trustpoint central, and specifies a key pair to be certified for the trustpoint central:

```
ciscoasa(config)# crypto ca trustpoint central
ciscoasa(ca-trustpoint)# keypair exchange
```

Related Commands

Command	Description
crypto ca trustpoint	Enters crypto ca trustpoint configuration mode.
crypto key generate dsa	Generates DSA keys.
crypto key generate rsa	Generates RSA keys.
default enrollment	Returns enrollment parameters to their defaults.

keysize

To specify the size of the public and private keys generated by the local Certificate Authority (CA) server at user certificate enrollment, use the **keysize** command in ca-server configuration mode. To reset the keysize to the default length of 1024 bits, use the **no** form of this command.

keysize*size*
no keysize

Syntax Description

size The size of the key, in bits. The size can be one of the following:

- 512
- 768
- 1024
- 2048
- 4096

Command Default

By default, each key in the key pair is 1024 bits long.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
ca-server configuration	• Yes	—	• Yes	—	—

Command History

Release	Modification
8.0(2)	This command was added.
9.13(1)	This command was removed.

Examples

The following example specifies a key size of 2048 bits for all public and private key pairs generated for users by the local CA server:

```
ciscoasa(config)# crypto ca server
ciscoasa
(config-ca-server)
)# keysize 2048
ciscoasa
```

```
(config-ca-server)
#
```

The following example resets the key size to the default length of 1024 bits for all public and private key pairs generated for users by the local CA server:

```
ciscoasa(config)# crypto ca server
ciscoasa
(config-ca-server)
# no keysize
ciscoasa
(config-ca-server)
#
```

Related Commands

Command	Description
crypto ca server	Provides access to the ca-server configuration mode command set, which allows you to configure and manage the local CA.
issuer-name	Specifies the subject name DN of the certificate authority certificate.
subject-name-default	Specifies a generic subject name DN to be used along with the username in all user certificates issued by a CA server.

keysize server

To specify the size of the public and private keys generated by the local Certificate Authority (CA) server for configuring the size of the CA keypair, use the **keysize server** command in ca-server configuration mode. To reset the keysize to the default length of 1024 bits, use the **no** form of this command.

keysize server *size*
no keysize server

Syntax Description

size The size of the key, in bits. The size can be one of the following:

- 512
- 768
- 1024
- 2048
- 4096

Command Default

By default, each key in the key pair is 1024 bits long.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Ca-server configuration	• Yes	—	• Yes	—	—

Command History

Release	Modification
8.0(2)	This command was added.
9.13(1)	This command was removed.

Examples

The following example specifies a key size of 2048 bits for the CA certificate:

```
ciscoasa(config)# crypto ca server
ciscoasa
(config-ca-server)
)# keysize server 2048
ciscoasa
(config-ca-server)
#
```

The following example resets the key size to the default length of 1024 bits for the CA certificate:

```
ciscoasa(config)# crypto ca server
ciscoasa
(config-ca-server)
# no keysize server
ciscoasa
(config-ca-server)
#
```

Related Commands

Command	Description
crypto ca server	Provides access to the ca-server configuration mode command set, which allows you to configure and manage the local CA.
issuer-name	Specifies the subject name DN of the certificate authority certificate.
keysize	Specifies the key pair size for the user certificate.
subject-name-default	Specifies a generic subject name DN to be used along with the username in all user certificates issued by a CA server.

key-string

To manually add a public SSH host key for a server for the on-board Secure Copy (SCP) client, use the **key-string** command in server configuration mode. You can access the server configuration mode by first entering the **ssh pubkey-chain** command. This command prompts you to enter a key string. When the string is saved to the configuration, it is hashed using SHA-256, and stored as the **key-hash** command. Therefore, to remove the string, use the **no key-hash** command.

key-string *key_string*

Syntax Description

key_string	Enters the public key.

Command Default

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Server configuration	• Yes	• Yes	• Yes	—	• Yes

Command History

Release	Modification
9.1(5)	This command was added.

Usage Guidelines

You can copy files to and from the ASA using the on-board SCP client. The ASA stores the SSH host key for each SCP server to which it connects. You can manually add or delete servers and their keys from the ASA database if desired.

For each server, you can specify the **key-string** (public key) or **key-hash** (hashed value) of the SSH host. The *key_string* is the Base64 encoded RSA public key of the remote peer. You can obtain the public key value from an open SSH client; that is, from the .ssh/id_rsa.pub file. After you submit the Base64 encoded public key, that key is then hashed via SHA-256.

Examples

The following example adds a host string key for the server at 10.7.8.9:

```
ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.7.8.9
ciscoasa(config-ssh-pubkey-server)# key-string
Enter the base 64 encoded RSA public key.
End with the word "exit" on a line by itself
ciscoasa(config-ssh-pubkey-server-string)# c1:b1:30:29:d7:b8:de:6c:97:77:10:d7:46:41:63:87
ciscoasa(config-ssh-pubkey-server-string)# exit
```

The following example shows the saved hashed key:

```
ciscoasa(config-ssh-pubkey-server)# show running-config ssh
ssh scopy enable
ssh stricthostkeycheck
ssh pubkey-chain
  server 10.7.8.9
    key-hash sha256
65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19
```

Related Commands

Command	Description
copy	Copies a file to or from the ASA.
key-hash	Enters a hashed SSH host key.
key-string	Enters a public SSH host key.
ssh pubkey-chain	Manually adds or deletes servers and their keys from the ASA database.
ssh stricthostkeycheck	Enables SSH host key checking for the on-board Secure Copy (SCP) client.

kill

To terminate a Telnet session, use the **kill** command in privileged EXEC mode.

kill*telnet_id*

Syntax Description

telnet_id	Specifies the Telnet session ID.

Command Default

No default behaviors or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added.

Usage Guidelines

The **kill** command lets you terminate a Telnet session. Use the who command to see the Telnet session ID. When you kill a Telnet session, the ASA lets any active commands terminate and then drops the connection without warning.

Examples

The following example shows how to terminate a Telnet session with the ID “2”. First, the who command is entered to display the list of active Telnet sessions. Then the **kill 2** command is entered to terminate the Telnet session with the ID “2”.

```
ciscoasa# who
2: From 10.10.54.0

ciscoasa# kill 2
```

Related Commands

Command	Description
telnet	Configures Telnet access to the ASA.
who	Displays a list of active Telnet sessions.