Commands for the Catalyst 6500 Series Switch SSL Services Module

This chapter contains an alphabetical listing of commands for the Catalyst 6500 series switch SSL Services Module.

For additional SSL Services Module information, refer to the following documentation:

- *Catalyst 6500 Series Switch SSL Services Module Configuration Note*
- *Catalyst 6500 Series Switch SSL Services Module System Message Guide*
- *Catalyst 6500 Series Switch SSL Services Module Installation and Verification Note*

clear ssl-proxy conn

To clear all TCP connections on the entire system, use the **clear ssl-proxy conn** command.

clear ssl-proxy conn [**context** *name* [**module** [*module*]]][**service** *name* [**context** *name* [**module** [*module*]]]]

Syntax Description

context *name*	(Optional) Clears the connections for a specific context.
module *module*	(Optional) Clears the connections for the specified module type. The available options for the module variable are as follows: • **all**—All CPUs • **fdu**—FDU CPU • **ssl1**—SSL1 CPU • **tcp1**—TCP1 CPU • **tcp2**—TCP2 CPU
service *name*	(Optional) Clears the connections for the specified service.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	This command was changed to add the following keywords: • **context** *name* • **module** *module*

Examples

This example shows how to clear the connections for the specified service:

```
ssl-proxy# clear ssl-proxy conn service S6
```

This example shows how to clear all TCP connections on the entire system:

```
ssl-proxy# clear ssl-proxy conn
ssl-proxy#
```

clear ssl-proxy content

To clear all TCP connections on the entire system, use the **clear ssl-proxy conn** command.

clear ssl-proxy content {**all** | **rewrite** | **scanning**} [**module** [*module*]]

Syntax Description

all	Clears all content statistics.
scanning	Clears scanning statistics.
rewrite	Clears rewriting statistics.
module *module*	(Optional) Clears statistics for the specified module type. The available options for the module variable are as follows: • **all**—All CPUs • **fdu**—FDU CPU • **ssl1**—SSL1 CPU • **tcp1**—TCP1 CPU • **tcp2**—TCP2 CPU

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Usage Guidelines

To reset all the content statistics that the SSL Services Module maintains, use the **clear ssl-proxy content all** command.

Examples

This example shows how to clear all of the content statistics:

```
ssl-proxy# clear ssl-proxy content all
```

clear ssl-proxy session

To clear all entries from the session cache, use the **clear ssl-proxy session** command.

clear ssl-proxy session [**service** [*name*] [**context** *name* [**module** [*module*]]]]

Syntax Description

context *name*	(Optional) Clears the session cache for a specific context.
module *module*	(Optional) Clears session cache for the specified module type. The available options for the module variable are as follows: • **all**—All CPUs • **fdu**—FDU CPU • **ssl1**—SSL1 CPU • **tcp1**—TCP1 CPU • **tcp2**—TCP2 CPU
service *name*	(Optional) Clears the session cache for the specified service.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	This command was changed to add the following keywords: • **context** *name* • **module** *module*

Usage Guidelines

To clear all entries from the session cache for all services, use the **clear ssl-proxy session** command without options.

Examples

This example shows how to clear the entries from the session cache for the specified service on the SSL Services Module:

```
ssl-proxy# clear ssl-proxy session service S6
```

This example shows how to clear all entries in the session cache that are maintained on the SSL Services Module:

```
ssl-proxy# clear ssl-proxy session
ssl-proxy#
```

clear ssl-proxy stats

To reset the statistics counters that are maintained in the different system components on the SSL Services Module, use the **clear ssl-proxy stats** command.

clear ssl-proxy stats [**context** [*name*] | **crypto** | **fdu** | **hdr** | **ipc** | **module** [*module*] | **pki** | **service** | **ssl** | **tcp** | **url**]

Syntax Description

context	(Optional) Clears statistics information about the context.
name	(Optional) Specifies the name of the context.
crypto	(Optional) Clears statistics information about the crypto.
fdu	(Optional) Clears statistics information about the FDU.
hdr	(Optional) Clears statistics information about HTTP header insertion.
ipc	(Optional) Clears statistics information about the inter-process communications (IPC).
module *module*	(Optional) Clears statistics information about the specified module type. The available options for the module variable are as follows: • **all**—All CPUs • **fdu**—FDU CPU • **ssl1**—SSL1 CPU • **tcp1**—TCP1 CPU • **tcp2**—TCP2 CPU
pki	(Optional) Clears information about the public key infrastruture (PKI).
service *name*	(Optional) Clears statistics information for a specific service.
ssl	(Optional) Clears statistics information about the SSL.
tcp	(Optional) Clears statistics information about the TCP.
url	(Optional) Clears statistics information about URL rewrite.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	This command was changed to add the following keywords: • **context** *name* • **hdr** • **module** *module* • **url**

Usage Guidelines

To reset all the statistics counters that the SSL Services Module maintains, use the **clear ssl-proxy stats** command without options.

Examples

This example shows how to reset the statistics counters that are maintained in the different system components on the SSL Services Module:

```
ssl-proxy# clear ssl-proxy stats crypto
ssl-proxy# clear ssl-proxy stats ipc
ssl-proxy# clear ssl-proxy stats pki
ssl-proxy# clear ssl-proxy stats service S6
```

This example shows how to clear all the statistic counters that the SSL Services Module maintains:

```
ssl-proxy# clear ssl-proxy stats
ssl-proxy#
```

crypto pki export pem

To export privacy-enhanced mail (PEM) files from the SSL Services Module, use the **crypto pki export pem** command.

crypto pki export *trustpoint_label* **pem** {**terminal** {**des** | **3des**} {**url** *url*}} *pass_phrase*

Syntax Description

trustpoint-label	Name of the trustpoint.
terminal	Displays the request on the terminal.
des	Specifies the 56-bit DES-CBC encryption algorithm.
3des	Specifies the 168-bit DES (3DES) encryption algorithm.
url *url*	Specifies the URL location. Valid values are as follows: • **ftp:**—Exports to the FTP: file system • **null:**—Exports to the NULL: file system • **nvram:**—Exports to the NVRAM: file system • **rcp:**—Exports to the RCP: file system • **scp:**—Exports to the SCP: file system • **system:**—Exports to the system: file system • **tftp:**—Exports to the TFTP: file system
pass-phrase	Pass phrase that is used to protect the private key.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The syntax for this command changed from **crypto ca** to **crypto pki.**

Usage Guidelines

The *pass_phrase* can be any phrase including spaces and punctuation except for the question mark (**?**), which has a special meaning to the Cisco IOS parser.

Pass-phrase protection associates a pass phrase with the key. The pass phrase is used to encrypt the key when it is exported. When this key is imported, you must enter the same pass phrase to decrypt it.

A key that is marked as unexportable cannot be exported.

You can change the default file extensions when prompted. The default file extensions are as follows:

- public key (.pub)
- private key (.prv)
- certificate (.crt)
- CA certificate (.ca)
- signature key (-sign)
- encryption key (-encr)

Note In SSL software release 1.2, only the private key (.prv), the server certificate (.crt), and the issuer CA certificate (.ca) of the server certificate are exported. To export the whole certificate chain, including all the CA certificates, use a PKCS12 file instead of PEM files.

Examples

This example shows how to export a PEM-formatted file on the SSL Services Module:

```
ssl-proxy(config)# crypto ca export TP5 pem url tftp://10.1.1.1/tp99 3des password
% Exporting CA certificate...
Address or name of remote host [10.1.1.1]?
Destination filename [tp99.ca]?
% File 'tp99.ca' already exists.
% Do you really want to overwrite it? [yes/no]: yes
!Writing file to tftp://10.1.1.1/tp99.ca!
% Key name: key1
    Usage: General Purpose Key
% Exporting private key...
Address or name of remote host [10.1.1.1]?
Destination filename [tp99.prv]?
% File 'tp99.prv' already exists.
% Do you really want to overwrite it? [yes/no]: yes
!Writing file to tftp://10.1.1.1/tp99.prv!
% Exporting router certificate...
Address or name of remote host [10.1.1.1]?
Destination filename [tp99.crt]?
% File 'tp99.crt' already exists.
% Do you really want to overwrite it? [yes/no]: yes
!Writing file to tftp://10.1.1.1/tp99.crt!

ssl-proxy(config)#
```

Related Commands

crypto pki import pem

crypto pki import pem

To import a PEM-formatted file to the SSL Services Module, use the **crypto pki import pem** command.

crypto pki import *trustpoint_label* **pem** [**exportable**] {**terminal** | **url** *url* | **usage-keys**} *pass_phrase*

Syntax Description

trustpoint-label	Name of the trustpoint.
exportable	(Optional) Specifies the key that can be exported.
terminal	Displays the request on the terminal.
url *url*	Specifies the URL location. Valid values are as follows: • **ftp:**—Exports to the FTP: file system • **null:**—Exports to the null: file system • **nvram:**—Exports to the NVRAM: file system • **rcp:**—Exports to the RCP: file system • **scp:**—Exports to the SCP: file system • **system:**—Exports to the system: file system • **tftp:**—Exports to the TFTP: file system
pass_phrase	Pass phrase.
usage-keys	Specifies that two special-usage key pairs should be generated, instead of one general-purpose key pair.

Defaults

This command has no default settings.

Command History

Global configuration

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The syntax for this command changed from **crypto ca** to **crypto pki.**

Usage Guidelines

You will receive an error if you enter the pass phrase incorrectly. The *pass_phrase* can be any phrase including spaces and punctuation except for the question mark (**?**), which has a special meaning to the Cisco IOS parser.

Pass-phrase protection associates a pass phrase with the key. The pass phrase is used to encrypt the key when it is exported. When this key is imported, you must enter the same pass phrase to decrypt it.

When importing RSA keys, you can use a public key or its corresponding certificate.

The **crypto pki import pem** command imports only the private key (.prv), the server certificate (.crt), and the issuer CA certificate (.ca). If you have more than one level of CA in the certificate chain, you need to import the root and subordinate CA certificates before this command is issued for authentication. Use cut-and-paste or TFTP to import the root and subordinate CA certificates.

Examples

This example shows how to import a PEM-formatted file from the SSL Services Module:

```
ssl-proxy(config)# crypto pki import TP5 pem url tftp://10.1.1.1/TP5 password
% Importing CA certificate...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.ca]?
Reading file from tftp://10.1.1.1/TP5.ca
Loading TP5.ca from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 1976 bytes]

% Importing private key PEM file...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.prv]?
Reading file from tftp://10.1.1.1/TP5.prv
Loading TP5.prv from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 963 bytes]

% Importing certificate PEM file...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.crt]?
Reading file from tftp://10.1.1.1/TP5.crt
Loading TP5.crt from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 1692 bytes]
% PEM files import succeeded.
ssl-proxy(config)# end
ssl-proxy#
*Apr 11 15:11:29.901: %SYS-5-CONFIG_I: Configured from console by console
```

Related Commands

crypto pki export pem

crypto pki export pkcs12

To export a PKCS12 file from the SSL Services Module, use the **crypto pki export pkcs12** command.

crypto pki export *trustpoint_label* **pkcs12** *file_system* [*pkcs12_filename*] *pass_phrase*

Syntax Description

trustpoint_label	Specifies the trustpoint label.
file_system	Specifies the file system. Valid values are **scp:, ftp:**, **nvram:**, **rcp:**, and **tftp:**
pkcs12_filename	(Optional) Specifies the name of the PKCS12 file to import.
pass_phrase	Specifies the pass phrase of the PKCS12 file.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The syntax for this command changed from **crypto ca** to **crypto pki.**

Usage Guidelines

Imported key pairs cannot be exported.

If you are using SSH, we recommend using SCP (secure file transfer) when exporting a PKCS12 file. SCP authenticates the host and encrypts the transfer session.

If you do not specify *pkcs12_filename*, you will be prompted to accept the default filename (the default filename is the *trustpoint_label*) or enter the filename. For the **ftp:** or **tftp:** value, include the full path in the *pkcs12_filename*.

You will receive an error if you enter the pass phrase incorrectly.

If there is more than one level of CA, the root CA and all the subordinate CA certificates are exported in the PKCS12 file.

Examples

This example shows how to export a PKCS12 file using SCP:

```
ssl-proxy(config)# crypto pki export TP1 pkcs12 scp: sky is blue
Address or name of remote host []? 10.1.1.1
Destination username [ssl-proxy]? admin-1
Destination filename [TP1]? TP1.p12

Password:

Writing TP1.p12 Writing pkcs12 file to scp://admin-1@10.1.1.1/TP1.p12

Password:
!
CRYPTO_PKI:Exported PKCS12 file successfully.
ssl-proxy(config)#
```

crypto pki import pkcs12

To import a PKCS12 file to the SSL Services Module, use the **crypto pki import pkcs12** command.

crypto pki import *trustpoint_label* **pkcs12** *file_system* [*pkcs12_filename*] *pass_phrase*

Syntax Description

trustpoint_label	Name of the trustpoint who issues the certificate that a user is going to export. When you export the PKCS12 file, the trustpoint name is the RSA key name.
file_system	Specifies the file system. Valid values are as follows: • **ftp:**—Imports from the FTP: file system • **nvram:**—Imports from the NVRAM: file system • **rcp:**—Imports from the RCP: file system • **scp:**—Imports from the SCP: file system • **tftp:**—Imports from the TFTP: file system
pkcs12_filename	(Optional) Specifies the name of the PKCS12 file to import.
pass_phrase	Passphrase that is used to encrypt the PKCS12 file for export.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The syntax for this command changed from **crypto ca** to **crypto pki.**

Usage Guidelines

If you are using SSH, we recommend using SCP (secure file transfer) when importing a PKCS12 file. SCP authenticates the host and encrypts the transfer session.

If you do not specify *pkcs12_filename*, you will be prompted to accept the default filename (the default filename is the *trustpoint_label*) or to enter the filename. For the **ftp:** or **tftp:** value, include the full path in the *pkcs12_filename*.

You will receive an error if you enter the pass phrase incorrectly.

If there is more than one level of CA, the root CA and all the subordinate CA certificates are exported in the PKCS12 file.

Security Measures

Keep the PKCS12 file stored in a secure place with restricted access.

An RSA keypair is more secure than a passphrase because the private key in the key pair is not known by multiple parties. When you export an RSA key pair to a PKCS#12 file, the RSA key pair now is only as secure as the passphrase.

To create a good passphrase, be sure to include numbers, as well as both lowercase and uppercase letters. Avoid publicizing the passphrase by mentioning it in e-mail or cell phone communications because the information could be accessed by an unauthorized user.

Examples

This example shows how to import a PKCS12 file using SCP:

```
ssl-proxy(config)# crypto pki import TP2 pkcs12 scp: sky is blue
Address or name of remote host []? 10.1.1.1
Source username [ssl-proxy]? admin-1
Source filename [TP2]? /users/admin-1/pkcs12/TP2.p12

Password:password
Sending file modes:C0644 4379 TP2.p12
!
ssl-proxy(config)#
*Aug 22 12:30:00.531:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
ssl-proxy(config)#
```

crypto key generate rsa (CA)

To generate RSA key pairs, use the **crypto key generate rsa** command in global configuration mode.

crypto key generate rsa [**usage-keys | general-keys**] [*key-pair-label*]

Syntax Description

usage-keys	(Optional) Specifies that two special-usage key pairs should be generated, instead of one general-purpose key pair.
general-keys	(Optional) Specifies that the general-purpose key pair should be generated.
key-pair-label	Optional) Specifies the name of the key pair that router will use. (If this argument is enabled, you must specify either **usage-keys** or **general-keys**.)

Defaults

Rivest, Shamir, and Adelman (RSA) key pairs do not exist.

If key-pair-label is not specified, the fully qualified domain name (FQDN) of the router is used and general-purpose keys are generated.

Command Modes

Global configuration mode

Command History

Release	Modification
11.3 T	This command was introduced.
12.2(8)T	The general-keys keyword and the key-pair-label argument were added.

Usage Guidelines

Use this command to generate RSA key pairs for your Cisco device (such as a router).

RSA keys are generated in pairs—one public RSA key and one private RSA key.

If your router already has RSA keys when you issue this command, you will be warned and prompted to replace the existing keys with new keys.

Note Before issuing this command, ensure your router has a host name and IP domain name configured (with the **hostname** and **ip domain-name** commands). You will be unable to complete the **crypto key generate rsa** command without a host name and IP domain name. (This is not true only when you generate a named-key-pair.)

This command is not saved in the router configuration; however, the keys generated by this command are saved in the private configuration in NVRAM (which is never displayed to the user or backed up to another device).

There are two mutually exclusive styles of RSA key pairs: special-usage keys and general-purpose keys. When you generate RSA key pairs, you will be prompted to select either generate special-usage keys or general-purpose keys.

Examples

Special-Usage Keys

If you generate special-usage keys, two pairs of RSA keys will be generated. One pair will be used with any Internet Key Exchange (IKE) policy that specifies RSA signatures as the authentication method, and the other pair used with any IKE policy that specifies RSA-encrypted nonces as the authentication method. (You configure RSA signatures or RSA-encrypted nonces in your IKE policies as described in the *Cisco IOS Security Configuration Guide*.)

A certification authority (CA) is used only with IKE policies specifying RSA signatures, not with IKE policies specifying RSA-encrypted nonces. (However, you could specify more than one IKE policy and have RSA signatures specified in one policy and RSA-encrypted nonces in another policy.)

If you plan to have both types of RSA authentication methods in your IKE policies, you might prefer to generate special-usage keys. With special-usage keys, each key is not unnecessarily exposed. (Without special-usage keys, one key is used for both purposes, increasing that key's exposure.)

General-Purpose Keys

If you generate general-purpose keys, only one pair of RSA keys will be generated. This pair will be used with IKE policies specifying either RSA signatures or RSA-encrypted nonces. Therefore, a general-purpose key pair might be used more frequently than a special-usage key pair.

Named Key Pairs

If you generate a named key pair using the *key-pair-label* argument, you must also specify the **usage-keys** keyword or the **general-keys** keyword. Named key pairs allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

Modulus Length

When you generate RSA keys, you will be prompted to enter a modulus length. A longer modulus could offer stronger security but takes longer to generate (see Table 1 for sample times) and takes longer to use. A length of less than 512 bits is normally not recommended. (In certain situations, the shorter modulus may not function properly with IKE, so Cisco recommends using a minimum modulus of 1024 bits.)

Table 1 Sample Times Required to Generate RSA Keys

	Modulus Length			
Router	**360 bits**	**512 bits**	**1024 bits**	**2048 bits**
Cisco 2500	11 seconds	20 seconds	4 minutes, 38 seconds	longer than 1 hour
Cisco 4700	less than 1 second	1 second	4 seconds	50 seconds

Examples

The following example generates special-usage RSA keys.

```
crypto key generate rsa usage-keys
The name for the keys will be: myrouter.example.com

Choose the size of the key modulus in the range of 360 to 2048 for your Signature Keys.
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus[512]? <return>
Generating RSA keys.... [OK].

Choose the size of the key modulus in the range of 360 to 2048 for your Encryption Keys.
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus[512]? <return>
Generating RSA keys.... [OK].
```

The following example generates general-purpose RSA keys. (Note, you cannot generate both special-usage and general-purpose keys; you can generate only one or the other.)

```
crypto key generate rsa
The name for the keys will be: myrouter.example.com

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keys. Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus[512]? <return>
Generating RSA keys.... [OK].
```

The following example generates the general-purpose RSA key pair “exampleCAkeys”:

```
crypto key generate rsa general-purpose exampleCAkeys
crypto ca trustpoint exampleCAkeys
 enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
 rsakeypair exampleCAkeys 1024 1024
```

Related Commands

Command	Description
	Specifies which key pair to associate with the certificate.

crypto key zeroize rsa

To delete all RSA keys from your router, use the **crypto key zeroize rsa** command in global configuration mode.

crypto key zeroize rsa [*key-pair-label*]

Syntax Description

key-pair-label	(Optional) Specifies the name of the key pair that router will delete.

Defaults

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
11.3 T	This command was introduced.
12.2(8)T	The *key-pair-label* argument was added.

Usage Guidelines

This command deletes all Rivest, Shamir, and Adelman (RSA) keys that were previously generated by your router unless you include the *key-pair-label* argument, which will delete only the specified RSA key pair. If you issue this command, you must also perform two additional tasks for each trustpoint that is associated with the key pair that was deleted:

- Ask the certification authority (CA) administrator to revoke your router's certificates at the CA; you must supply the challenge password you created when you originally obtained the router's certificates using the **crypto ca enroll** command.
- Manually remove the router's certificates from the configuration by removing the configured trustpoint (using the **no crypto ca trustpoint** *name* command.)

Note This command cannot be undone (after you save your configuration), and after RSA keys have been deleted, you cannot use certificates or the CA or participate in certificate exchanges with other IP Security (IPSec) peers unless you reconfigure CA interoperability by regenerating RSA keys, getting the CA's certificate, and requesting your own certificate again.

This command is not saved to the configuration.

Examples

The following example deletes the general-purpose RSA key pair that was previously generated for the router. After deleting the RSA key pair, the administrator contacts the CA administrator and requests that the router's certificate be revoked. The administrator then deletes the router's certificate from the configuration.

```
crypto key zeroize rsa
crypto ca certificate chain
 no certificate
```

Related Commands

Command	Description
certificate	Adds certificates manually.
crypto ca certificate chain	Enters the certificate chain configuration mode.
crypto ca trustpoint	Declares the CA that your router should use.
	Specifies which key pair to associate with the certificate.

crypto key decrypt rsa

To delete the encrypted key and leave only the unencrypted key, use the **crypto key decrypt rsa** command.

crypto key decrypt [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase*

Syntax Description

write	(Optional) Writes the configuration to the startup configuration.
name *key-name*	(Optional) Name of the key.
passphrase *passphrase*	Pass phrase.

Defaults

This command has no default settings.

Command Modes

Global configuration mode

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Usage Guidelines

Entering the **write** keyword immediately saves the unencrypted key to NVRAM. If you do not enter the **write** keyword, you must manually write the configuration to NVRAM; otherwise, the key remains encrypted the next time that the router is reloaded.

Examples

This example shows how to display the administration VLAN and related IP and gateway addresses:

```
ssl-proxy(config)# crypto key decrypt rsa name pki1-72a.cisco.com passphrase cisco1234
WARNING: Configuration with decrypted key not saved.
  Please save it manually as soon as possible to
  save decrypted key
ssl-proxy(config)# end
ssl-proxy# show crypto key mypubkey rsa
Key name: pki1-72a.cisco.com
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  30819F30 0D06092A 864886F7 0D010101 05000381
...
% Key pair was generated at: 15:42:15 PST Jun

ssl-proxy#
```

Related Commands

crypto key encrypt rsa
crypto key lock rsa
crypto key unlock rsa

crypto key encrypt rsa

To encrypt the RSA keys, use the **crypto key encrypt rsa** command.

crypto key encrypt [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase*

Syntax Description

write	(Optional) Writes the configuration to the startup configuration.
name *key-name*	(Optional) Name of the key.
passphrase *passphrase*	Pass phrase.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Usage Guidelines

After you enter this command, the router can continue to use the key; the key remains unlocked.

If you do not enter the **write** keyword, you must manually write the configuration to NVRAM; otherwise, the encrypted key will be lost the next time that the router is reloaded.

Examples

This example shows how to encrypt the RSA key "pki1-72a.cisco.com." Enter the **show crypto key mypubkey rsa** command to verify that the RSA key is encrypted (protected) and unlocked.

```
ssl-proxy(config)# crypto key encrypt rsa name pki1-72a.cisco.com passphrase cisco1234
ssl-proxy(config)# exit
ssl-proxy# show crypto key mypubkey rsa
 Key name:pki1-72a.cisco.com
 Usage:General Purpose Key
 *** The key is protected and UNLOCKED. ***
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E0CC9A 1D23B52C
...
 % Key pair was generated at:00:15:32 GMT Jun 25 2003

ssl-proxy#
```

Related Commands

crypto key decrypt rsa
crypto key lock rsa
crypto key unlock rsa

crypto key export rsa pem

To export a PEM-formatted RSA key to the SSL Services Module, use the **crypto key export rsa pem** command.

crypto key export rsa *keylabel* **pem** {**terminal** | **url** *url*} {{**3des** | **des**} [**exportable**] *pass_phrase*}

Syntax Description

keylabel	Name of the key.
terminal	Displays the request on the terminal.
url *url*	Specifies the URL location. Valid values are as follows: • **ftp:**—Exports to the FTP: file system • **null:**—Exports to the null: file system • **nvram:**—Exports to the NVRAM: file system • **rcp:**—Exports to the RCP: file system • **scp:**—Exports to the SCP: file system • **system:**—Exports to the system: file system • **tftp:**—Exports to the TFTP: file system
3des	Specifies the 168-bit DES (3DES) encryption algorithm.
des	Specifies the 56-bit DES-CBC encryption algorithm.
exportable	(Optional) Specifies that the key can be exported.
pass_phrase	Pass phrase.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The pass phrase can be any phrase including spaces and punctuation except for the question mark (**?**), which has a special meaning to the Cisco IOS parser.

Pass-phrase protection associates a pass phrase with the key. The pass phrase is used to encrypt the key when it is exported. When this key is imported, you must enter the same pass phrase to decrypt it.

Examples

This example shows how to export a key from the SSL Services Module:

```
ssl-proxy(config)# crypto key export rsa test-keys pem url scp: 3des password
% Key name:test-keys
   Usage:General Purpose Key
Exporting public key...
Address or name of remote host []? 7.0.0.7
Destination username [ssl-proxy]? lab
Destination filename [test-keys.pub]?

Password:

Writing test-keys.pub Writing file to scp://lab@7.0.0.7/test-keys.pub
Password:
!
Exporting private key...
Address or name of remote host []? 7.0.0.7
Destination username [ssl-proxy]? lab
Destination filename [test-keys.prv]?

Password:

Writing test-keys.prv Writing file to scp://lab@7.0.0.7/test-keys.prv
Password:
ssl-proxy(config)#
```

crypto key import rsa pem

To import a PEM-formatted RSA key from an external system, use the **crypto key import rsa pem** command.

crypto key import rsa *keylabel* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase*

Syntax Description

keylabel	Name of the key.
usage-keys	(Optional) Specifies that two special-usage key pairs should be generated, instead of one general-purpose key pair.
terminal	Displays the request on the terminal.
url *url*	Specifies the URL location. Valid values are as follows: • **ftp:**—Imports from the FTP: file system • **null:**—Imports from the null: file system • **nvram:**—Imports from the NVRAM: file system • **rcp:**—Imports from the RCP: file system • **scp:**—Imports from the SCP: file system • **system:**—Imports from the system: file system • **tftp:**—Imports from the TFTP: file system
exportable	(Optional) Specifies that the key can be exported.
passphrase	Pass phrase.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The pass phrase can be any phrase including spaces and punctuation except for the question mark (**?**), which has a special meaning to the Cisco IOS parser.

Pass-phrase protection associates a pass phrase with the key. The pass phrase is used to encrypt the key when it is exported. When this key is imported, you must enter the same pass phrase to decrypt it.

Examples

This example shows how to import a PEM-formatted RSA key from an external system and export the PEM-formatted RSA key to the SSL Services Module:

```
ssl-proxy(config)# crypto key import rsa newkeys pem url scp: password
% Importing public key or certificate PEM file...
Address or name of remote host []? 7.0.0.7
Source username [ssl-proxy]? lab
Source filename [newkeys.pub]? test-keys.pub

Password:
Sending file modes:C0644 272 test-keys.pub
Reading file from scp://lab@7.0.0.7/test-keys.pub!
% Importing private key PEM file...
Address or name of remote host []? 7.0.0.7
Source username [ssl-proxy]? lab
Source filename [newkeys.prv]? test-keys.prv

Password:
Sending file modes:C0644 963 test-keys.prv
Reading file from scp://lab@7.0.0.7/test-keys.prv!% Key pair import succeeded.

ssl-proxy(config)#
```

crypto key lock rsa

To lock the encrypted private key, use the **crypto key lock rsa** command.

crypto key lock rsa [**name** *key-name*] **passphrase** *passphrase*

Syntax Description

name *key-name*	(Optional) Name of the key.
passphrase *passphrase*	Pass phrase.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

After the key is locked, it cannot be used to authenticate the router to a peer device. This behavior disables any IPsec or SSL connections that use the locked key.

Any existing IPsec tunnels created on the basis of the locked key will be closed.

If all RSA keys are locked, SSH will automatically be disabled.

Examples

This example shows how to lock the key "pki1-72a.cisco.com." Enter the **show crypto key mypubkey rsa** command to verify that the key is protected (encrypted) and locked.

```
ssl-proxy# crypto key lock rsa name pki1-72a.cisco.com passphrase cisco1234
ssl-proxy# show crypto key mypubkey rsa
Key name:pki1-72a.cisco.com
 Usage:General Purpose Key
 *** The key is protected and LOCKED. ***
 Key is exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
...
% Key pair was generated at: 16:00:11 PST Feb 28 2002

ssl-proxy#
```

Related Commands

crypto key decrypt rsa
crypto key encrypt rsa
crypto key unlock rsa

crypto key unlock rsa

To unlock the encrypted private key, use the **crypto key unlock rsa** command.

crypto key unlock rsa [**name** *key-name*] **passphrase** *passphrase*

Syntax Description

name *key-name*	(Optional) Name of the key.
passphrase *passphrase*	Pass phrase.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Examples

This example shows how to lock the key "pki1-72a.cisco.com." Enter the **show crypto key mypubkey rsa** command to verify that the key is protected (encrypted) and locked.

```
ssl-proxy# crypto key unlock rsa name pki1-72a.cisco.com passphrase cisco1234
...
*Jun 18 00:26:08.275: %STE-5-UPDOWN: ssl-proxy service vip1 changed state to UP
...
ssl-proxy# show crypto key mypubkey rsa
Key name:pki1-72a.cisco.com
 Usage:General Purpose Key
 *** The key is protected and UNLOCKED. ***
 Key is exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
...
% Key pair was generated at: 16:00:11 PST Feb 28 2002

ssl-proxy#
```

Related Commands

crypto key decrypt rsa

crypto key encrypt rsa

crypto key lock rsa

debug ssl-proxy

To turn on the debug flags in different system components, use the **debug ssl-proxy** command. Use the **no** form of this command to turn off the debug flags.

debug ssl-proxy {**app** | **content** [*type*] | **fdu** [*type*] | **flash** [**module** [*module*]] | **health-probe** | **ipc** | **pki** [*type*] | **ssl** [*type*] | **tcp** [*type*] | **vlan**}

Syntax Description

app	Turns on App debugging.
content *type*	Turns on content debugging; (optional) *type* valid values are **detail**, **error**, **ipc**, **module** *module*, **rewriting**, and **scanning**. See the "Usage Guidelines" section for additional information.
fdu *type*	Turns on FDU debugging; (optional) *type* valid values are **cli**, **hash**, **ipc**, and **trace**. See the "Usage Guidelines" section for additional information.
flash	Turns on Flash debugging.
module *module*	Specifies the module to be debugged. The available options for the module variable are as follows: • **fdu**—FDU CPU • **ssl1**—SSL1 CPU • **tcp1**—TCP1 CPU
health-probe	Turns on health probe debugging.
ipc	Turns on IPC debugging.
pki *type*	Turns on PKI debugging; (optional) *type* valid values are **cert**, **events**, **history**, **ipc**, and **key**. See the "Usage Guidelines" section for additional information.
ssl *type*	Turns on SSL debugging; (optional) *type* valid values are **alert**, **error**, **handshake**, and **pkt**. See the "Usage Guidelines" section for additional information.
tcp *type*	Turns on TCP debugging; (optional) *type* valid values are **event**, **packet**, **state**, and **timers**. See the "Usage Guidelines" section for additional information.
vlan	Turns on VLAN debugging.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	This command was changed to add the following keywords: • **content** *type* • **flash** • **health-probe** • **module** *module* • **vlan**

Usage Guidelines

The **content** *type* includes the following values:

- **detail**—content detail
- **error**—content error
- **ipc**—content ipc
- **module** *module*—module to be debugged; *module* includes the following values:
  - **fdu**—fdu cpu
  - **ssl1**—ssl1 cpu
  - **tcp1**—tcp1 cpu
- **rewriting**—content rewriting
- **scanning**—content scanning

The **fdu** *type* includes the following values:

- **cli**—Debugs the FDU CLI.
- **hash**—Debugs the FDU hash.
- **ipc** —Debugs the FDU IPC.
- **trace**—Debugs the FDU trace.

The **pki** *type* includes the following values:

- **certs**—Debugs the certificate management.
- **events**—Debugs events.
- **history**—Debugs the certificate history.
- **ipc**—Debugs the IPC messages and buffers.
- **key**—Debugs key management.

The **ssl** *type* includes the following values:

- **alert**—Debugs the SSL alert events.
- **error**—Debugs the SSL error events.
- **handshake**—Debugs the SSL handshake events.
- **pkt**—Debugs the received and transmitted SSL packets.

Note Use the TCP debug commands only to troubleshoot basic connectivity issues under little or no load conditions (for instance, when no connection is being established to the virtual server or real server).

If you run TCP debug commands, the TCP module displays large amounts of debug information on the console, which can significantly slow down module performance. Slow module performance can lead to delayed processing of TCP connection timers, packets, and state transitions.

The **tcp** *type* includes the following values:

- **events**—Debugs the TCP events.
- **pkt**—Debugs the received and transmitted TCP packets.
- **state**—Debugs the TCP states.
- **timers**—Debugs the TCP timers.

Examples

This example shows how to turn on App debugging:

```
ssl-proxy# debug ssl-proxy app
ssl-proxy#
```

This example shows how to turn on FDU debugging:

```
ssl-proxy# debug ssl-proxy fdu
ssl-proxy#
```

This example shows how to turn on IPC debugging:

```
ssl-proxy# debug ssl-proxy ipc
ssl-proxy#
```

This example shows how to turn on PKI debugging:

```
ssl-proxy# debug ssl-proxy pki
ssl-proxy#
```

This example shows how to turn on SSL debugging:

```
ssl-proxy# debug ssl-proxy ssl
ssl-proxy#
```

This example shows how to turn on TCP debugging:

```
ssl-proxy# debug ssl-proxy tcp
ssl-proxy#
```

This example shows how to turn off TCP debugging:

```
ssl-proxy# no debug ssl-proxy tcp
ssl-proxy#
```

do

To execute EXEC-level commands from global configuration mode or other configuration modes or submodes, use the **do** command.

do *command*

Syntax Description

command	EXEC-level command to be executed.

Defaults

This command has no default settings.

Command Modes

Global configuration or any other configuration mode or submode from which you are executing the EXEC-level command.

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

Caution Do not enter the **do** command in EXEC mode. Interruption of service may occur.

You cannot use the **do** command to execute the **configure terminal** command because entering the **configure terminal** command changes the mode to configuration mode.

You cannot use the **do** command to execute the **copy** or **write** command in the global configuration or any other configuration mode or submode.

Examples

This example shows how to execute the EXEC-level **show interfaces** command from within global configuration mode:

```
ssl-proxy(config)# do show interfaces serial 3/0

Serial3/0 is up, line protocol is up
  Hardware is M8T-RS232
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output 1d17h, output hang never
  Last clearing of "show interface" counters never
.
.
.
ssl-proxy(config)#
```

interface ssl-proxy

To enter the subinterface configuration submode, use the **interface ssl-proxy** command. In interface configuration submode, you can configure a subinterface for the SSL Services Module.

Note The ssl-proxy0 interface is enabled by default and should not be shut down or otherwise configured.

interface 0.*subinterface-number*

Syntax Description

subinterface-number	Subinterface ID; valid values are from 0 to 4294967295.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module. This command replaces the **ssl-proxy vlan** command.

Usage Guidelines

When you upgrade to SSL software release 3.x from SSL software release 2.x or 1.x, the VLAN configuration is converted automatically to an subinterface configuration. For example, **ssl-proxy vlan 3** is converted to **interface ssl-proxy0.3**.

Note The ssl-proxy0 interface is enabled by default and should not be shut down or otherwise configured.

Table 2-2 lists the commands that are available in subinterface configuration submode.

Table 2-2 Subinterface Configuration Submode Command Descriptions

Syntax	Description
default	Sets a command to its defaults.
description	Allows you to enter a description for the subinterface.
encapsulation dot1q *vlan_ID* [**native**]	Sets the encapsulation type for the interface. Enter the **native** keyword to make this a native VLAN.
exit	Exits from the subinterface configuration submode.
ip address *ipaddress subnet* [**secondary**]	Configures the subinterface with an IP address and a subnet mask. Enter the **secondary** keywork to make this IP address a secondary address.
no	Negates a command or sets its defaults.

Table 2-2 Subinterface Configuration Submode Command Descriptions (continued)

Syntax	Description
[**no**] **shutdown**	Shuts down the subinterface. Use the **no** form of this command to put the subinterface in service.
standby [*group-number*] {**authentication** **text** *string*} | {**delay minimum** [*min-delay*] **reload** [*reload-delay*]} | {**ip** [*ip-address* [**secondary**]]} | {**mac-address** *mac-address*} | {**mac-refresh** *seconds*} | {**name** *group-name*} | {**preempt** [**delay**{**minimum** *delay* | **reload** *delay* | **sync** *delay*}]} | {**priority** *priority*} | {**redirects** [**enable** | **disable**] [**timers** *advertisement holddown*] [**unknown**]} | {**timers** [**msec**] *hellotime* [**msec**] *holdtime*} | {**track** *object-number* [**decrement** *priority*]} | [**version** {**1** | **2**}]	Configures redundancy on the subinterface. See the following commands for valid values: • **standby authentication** • **standby delay minimum reload** • **standby ip** • **standby mac-address** • **standby mac-refresh** • **standby name** • **standby preempt** • **standby priority** • **standby redirects** • **standby timers** • **standby track** • **standby use-bia** • **standby version**
timeout absolute *minutes seconds*	Sets the session timeout values for this interface. Valid values for *minutes* are from 0 to 71582787 minutes. Valid values for *seconds* are from 0 to 59 seconds.

The valid values for configuring HSRP are as follows:

- *group-number*—(Optional) Group number on the interface for which HSRP is being activated; valid values are from 0 to 255 for HSRP version 1; valid values are from 0 to 4095 for HSRP version 2. See the "standby version" section on page 2-132 for information about changing the HSRP version. If you do not specify a *group-number*, group **0** is used.
- **ip** *ip-addr*—Specifies the IP address of the HSRP interface.
- **priority** *priority*— Specifies the priority for the HSRP interface. Increase the priority of at least one interface in the HSRP group. The interface with the highest priority becomes active for that HSRP group.
- **prempt** —Enables preemption. When you enable preemption, if the local router has a hot standby priority that is higher than the current active router, the local router attempts to assume control as the active router. If you do not configure preemption, the local router assumes control as the active router only if it receives information indicating that no router is in the active state (acting as the designated router).
- **delay**—(Optional) Specifies the preemption delay. When a router first comes up, it does not have a complete routing table. If it is configured to preempt, it becomes the active router but cannot provide adequate routing services. You can configure a delay before the preempting router actually preempts the currently active router.

- *type time*—Specifies the preemption type and delay; valid values are as follows:
  - **minimum** *time*—Specifies the minimum delay period in delay seconds; valid values are from 0 to 3600 seconds (1 hour).
  - **reload** *time*—Specifies the preemption delay after a reload only.
  - **sync** *time*—Specifies the maximum synchronization period in delay seconds.
- **timers** [**msec**] *hellotime holdtime*—Configures the time between hello packets and the time before other routers declare the active hot standby or standby router to be down; valid values are as follows:
  - **msec**—(Optional) Interval in milliseconds. Millisecond timers allow for faster failover.
  - *hellotime*—Hello interval (in seconds); valid values are from 1 to 254 seconds. If you specify the **msec** keyword, the hello interval is in milliseconds; valid values are from 15 to 999 milliseconds. The default is 3 seconds.
  - *holdtime*—Time (in seconds) before the active or standby router is declared to be down; valid values are from x to 255; x is the *hellotime* plus 50 milliseconds and is rounded up to the nearest 1 second. If you specify the **msec** keyword, the holdtime is in milliseconds; valid values are from y to 3000 milliseconds; y is greater than or equal to 3 times the *hellotime* and is not less than 50 milliseconds. The default is 10 seconds.

Examples

This example shows how to enter the subinterface configuration submode:

```
ssl-proxy (config)# interface ssl-proxy 0.6
ssl-proxy (config-subif)#
```

This example shows how to configure the specified subinterface with an IP address and subnet mask:

```
ssl-proxy (config-subif)# ip address 208.59.100.18 255.0.0.0
ssl-proxy (config-subif)#
```

This example shows how to configure the HSRP on the SSL module:

```
ssl-proxy(config)# interface ssl-proxy 0.100
ssl-proxy(config-subif)# ip address 10.1.0.20 255.255.255.0
ssl-proxy(config-subif)# standby 1 ip 10.1.0.21
ssl-proxy(config-subif)# standby 1 priority 110
ssl-proxy(config-subif)# standby 1 preempt
ssl-proxy(config-subif)# standby 2 ip 10.1.0.22
ssl-proxy(config-subif)# standby 2 priority 100
ssl-proxy(config-subif)# standby 2 preempt
ssl-proxy(config-subif)# end
ssl-proxy#
```

Related Commands

show interfaces ssl-proxy
show ssl-proxy vlan

natpool

To define a pool of IP addresses, which the SSL Services Module uses for implementing the client NAT, use the **natpool** command.

natpool *nat-pool-name start_ip_addr end_ip_addr* **netmask** *netmask*

Syntax Description

nat-pool-name	NAT pool name.
start-ip-addr	First IP address in the pool.
end-ip-addr	Last IP address in the pool.
netmask *netmask*	Specifies the netmask address.

Defaults

This command has no default settings.

Command Modes

Context subcommand mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The **natpool** command (entered in context subcommand mode) replaces the **ssl-proxy natpool** command (entered in global subcommand mode).

Examples

This example shows how to define a pool of IP addresses:

```
ssl-proxy(config)# ssl-proxy context Example
ssl-proxy (config-context)# natpool NP2 207.59.10.01 207.59.10.08 netmask 255.0.0.0
ssl-proxy (config-context)#
```

Related Commands

show ssl-proxy natpool

policy health-probe tcp

To enter the TCP health probe configuration submode, use the **policy health-probe** command. In TCP health probe configuration submode, you can define the TCP health probe policy that is applied.

policy health-probe tcp *policy-name*

Syntax Description

policy-name	TCP health probe policy name.

Defaults

The defaults are as follows:

- **failed-interval** is 60 seconds.
- **interval** is 30 seconds.
- **maximum-retry** is 0.
- **open-timeout** is 80 seconds.
- **port** is the port of the server IP address that you configured in the SSL server proxy service.

Command Modes

Context subcommand mode

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Usage Guidelines

Table 2-10 lists the commands that are available in TCP health probe policy configuration submode.

Table 2-3 TCP Health Probe Submode Command Descriptions

Syntax	Description
interval *seconds*	(Optional) Allows you to set the interval between probes in seconds (from the end of the previous probe to the beginning of the next probe) when the server is healthy. The default is 30 seconds. The valid range is from 30 to 300 seconds.
failed-interval *seconds*	(Optional) Allows you to set the time between health checks after the service has been marked as failed. The default is 60 seconds. The valid range is from 30 to 3600 seconds.
maximum-retry *retries*	(Optional) Sets the number of failed probes that are allowed before marking the service as failed. The default is 0 retries. The valid range is from 1 to 5 retries.

Table 2-3 TCP Health Probe Submode Command Descriptions (continued)

Syntax	Description
open-timeout *seconds*	(Optional) Allows you to set the maximum time to wait to establish a TCP connection. The default is 80 seconds. The valid range is from 70 to 120 seconds.
port *port_number*	(Optional) Allows you to configure an optional port for the health probe. Valid values are from 1 to 65535. By default, the TCP health probe uses the server IP address and port for the SSL server proxy service. Enter the **port** command to specify a different port for the health probe. If you configured the SSL server proxy service with **no nat server**, the TCP health probe uses the virtual IP address that you configured on the SSL server proxy service instead of the server IP address. **Note** TCP health probe is not supported when you configure a wildcard proxy and **no nat server** on the SSL server proxy service. See the "service" section on page 2-58 for information on configuring the SSL server proxy service.

Examples

This example shows how to configure TCP health probe to check whether service at port 80 is up and running on server IP address 19.0.0.1:

```
ssl-proxy(config)# ssl-proxy context ssl
ssl-proxy(config-context)# service ssl-1
ssl-proxy(config-ctx-ssl-proxy)# virtual ipddr 7.100.100.180 protocol tcp port 443
ssl-proxy(config-ctx-ssl-proxy)# server ipaddr 19.0.0.1 protocol tcp port 80
ssl-proxy(config-ctx-ssl-proxy)# certificate rsa general-purpose trustpoint cert1024
ssl-proxy(config-ctx-ssl-proxy)# policy health-probe tcp probe1
ssl-proxy(config-ctx-ssl-proxy)# inservice
ssl-proxy(config-ctx-ssl-proxy)# exit
ssl-proxy(config-context)# policy health-probe tcp probe1
ssl-proxy(config-ctx-tcp-probe)# end
ssl-proxy#
```

This example shows the state of the SSL proxy service when the health probe has failed:

Note The proxy service is down until service at port 81 is up and running again.

```
ssl-proxy# show ssl-proxy service ssl-1 context ssl
Service id: 0, bound_service_id: 256
Virtual IP: 7.100.100.180, port: 443
Server IP: 19.0.0.1, port: 81
TCP Health Probe Policy: probe1
rsa-general-purpose certificate trustpoint: cert1024
  Certificate chain for new connections:
    Certificate:
       Key Label: cert1024.key, 1024-bit, exportable
       Key Timestamp: 05:18:23 UTC Dec 30 2005
       Serial Number: 12F332E200000000000D
    Root CA Certificate:
       Serial Number: 6522F512C30E078447D8AFC35567B101
  Certificate chain complete
```

```
Context name: ssl
Context Id  : 1
Admin Status: up
Operation Status: down
Proxy status: Health Probe Failed
```

This example shows how to configure TCP health probe to check whether service at port 81 is up and running on server IP address 19.0.0.1:

```
ssl-proxy(config-context)# service sslоffload
ssl-proxy(config-ctx-ssl-proxy)# virtual ipaddr 7.100.100.180 protocol tcp port 443
ssl-proxy(config-ctx-ssl-proxy)# server ipaddr 19.0.0.1 protocol tcp port 80
ssl-proxy(config-ctx-ssl-proxy)# certificate rsa general-purpose trustpoint cert1024
ssl-proxy(config-ctx-ssl-proxy)# policy health-probe tcp probe1
ssl-proxy(config-ctx-ssl-proxy)# nat client natpool
ssl-proxy(config-ctx-ssl-proxy)# inservice
ssl-proxy(config-ctx-ssl-proxy)# exit
ssl-proxy(config-context)# policy health-probe tcp probe1
ssl-proxy(config-ctx-tcp-probe)# 81
Warning: Port in the service sslоffload configuration (80) differs from the port in the
health probe configuration (81)
ssl-proxy(config-ctx-tcp-probe)# exit
ssl-proxy(config-context)#
```

This example shows how to configure TCP health probe to check whether service at port 80 is up and running on virtual IP address 7.100.100.180:

```
ssl-proxy(config-context)# service sslоffload
ssl-proxy(config-ctx-ssl-proxy)# virtual ipaddr 7.100.100.180 protocol tcp port 443
ssl-proxy(config-ctx-ssl-proxy)# server ipaddr 19.0.0.1 protocol tcp port 80
ssl-proxy(config-ctx-ssl-proxy)# certificate rsa general-purpose trustpoint cert1024
ssl-proxy(config-ctx-ssl-proxy)# policy health-probe tcp probe1
ssl-proxy(config-ctx-ssl-proxy)# no nat server
ssl-proxy(config-ctx-ssl-proxy)# nat client natpool
ssl-proxy(config-ctx-ssl-proxy)# inservice
ssl-proxy(config-ctx-ssl-proxy)# exit
ssl-proxy(config-context)# policy health-probe tcp probe1
ssl-proxy(config-ctx-tcp-probe)# exit
ssl-proxy(config-context)#
```

This example shows how to configure TCP health probe to check whether service at port 444 is up and running on virtual IP address 7.100.100.180:

```
ssl-proxy(config-context)# service sslоffload
ssl-proxy(config-ctx-ssl-proxy)# virtual ipaddr 7.100.100.180 protocol tcp port 443
ssl-proxy(config-ctx-ssl-proxy)# server ipaddr 19.0.0.1 protocol tcp port 80
ssl-proxy(config-ctx-ssl-proxy)# certificate rsa general-purpose trustpoint cert1024
ssl-proxy(config-ctx-ssl-proxy)# policy health-probe tcp probe1
ssl-proxy(config-ctx-ssl-proxy)# no nat server
ssl-proxy(config-ctx-ssl-proxy)# nat client natpool
ssl-proxy(config-ctx-ssl-proxy)# inservice
ssl-proxy(config-ctx-ssl-proxy)# exit
ssl-proxy(config-context)# policy health-probe tcp probe1
ssl-proxy(config-ctx-tcp-probe)# 444
ssl-proxy(config-ctx-tcp-probe)# exit
Warning: Port in the service sslоffload configuration (80) differs from the port in the
health probe configuration (444)
ssl-proxy(config-context)#
```

Related Commands

show ssl-proxy policy
show ssl-proxy service

policy http-header

To enter the HTTP header insertion configuration submode, use the **policy http-header** command.

policy http-header *http-header-policy-name*

Syntax Description

http-header-policy-name	HTTP header policy name.

Defaults

This command has no default settings.

Command Modes

Context subcommand mode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The **policy http-header** command (entered in context subcommand mode) replaces the **ssl-proxy policy http-header** command (entered in global subcommand mode). This command was changed to add the following submode commands: • **client-cert pem** • **alias**
SSL Services Module Release 3.1(5)	The following Context HTTP Header Insert Policy Configuration submode commands were introduced: • **pre-remove-http-hdr**

Usage Guidelines

In HTTP header insertion configuration submode, you can define the HTTP header insertion content policy that is applied to the payload.

HTTP header insertion allows you to insert additional HTTP headers to indicate to the real server that the connection is actually an SSL connection. These headers allow server applications to collect correct information for each SSL session and/or client.

You can insert these header types:

- Client Certificate—Client certificate header insertion allows the back-end server to see the attributes of the client certificate that the SSL module has authenticated and approved. When you specify **client-cert**, the SSL module passes the following headers to the back-end server:

Field To Insert	Description
ClientCert-Valid	Certificate validity state
ClientCert-Error	Error conditions
ClientCert-Fingerprint	Hash output

Field To Insert	Description
ClientCert-Subject-CN	X.509 subject's common name
ClientCert-Issuer-CN	X.509 certificate issuer's common name
ClientCert-Certificate-Version	X.509 certificate version
ClientCert-Serial-Number	Certificate serial number
ClientCert-Data-Signature-Algorithm	X.509 hashing and encryption method
ClientCert-Subject	X.509 subject's distinguished name
ClientCert-Issuer	X.509 certificate issuer's distinguished name
ClientCert-Not-Before	Certificate is not valid before this date
ClientCert-Not-After	Certificate is not valid after this date
ClientCert-Public-Key-Algorithm	The algorithm used for the public key
ClientCert-RSA-Public-Key-Size	Size of the RSA public key
ClientCert-RSA-Modulus-Size	Size of the RSA private key
ClientCert-RSA-Modulus	RSA modulus
ClientCert-RSA-Exponent	The public RSA exponent
ClientCert-X509v3-Authority-Key-Identifier	X.509 authority key identifier
ClientCert-X509v3-Basic-Constraints	X.509 basic constraints
ClientCert-X509v3-Key-Usage	X.509 key usage
ClientCert-X509v3-Subject-Alternative-Name	X.509 subject alternative name
ClientCert-X509v3-CRL-Distribution-Points	X.509 CRL distribution points
ClientCert-X509v3-Authority-Information-Access	X.509 authority information access
ClientCert-Signature-Algorithm	Certificate signature algorithm
ClientCert-Signature	Certificate signature

- Client Certificate in PEM format—When you specify **client-cert pem**, the SSL module sends the entire client certificate in PEM format.
- Client IP and Port Address—Network address translation (NAT) removes the client IP address and port information. When you specify **client-ip-port**, the SSL module inserts the client IP address and information about the client port into the HTTP header, allowing the server to see the client IP address and port.
- Custom—When you specify **custom** *custom-string*, the SSL module inserts the user-defined header into the HTTP header.
- Prefix—When you specify **prefix** *prefix-string*, the SSL module adds the specified prefix into the HTTP header to enable the server to identify that the connections are coming from the SSL module, not from other appliances.
- Header alias—Some applications use different names for the standard header. You can create an alias for the standard name of the header so that the same value is passed using the aliased name instead of the standard name that the SSL Services Module sends. If you have specified a prefix for header insertion, the prefix is also applied to the aliased name.

- SSL Session—Session headers, including the session ID, are used to cache client certificates that are based on the session ID. The session headers are also cached on a session basis if the server wants to track connections that are based on a particular cipher suite. When you specify **session**, the SSL Services Module passes information specific to an SSL connection to the back-end server in the form of the following session headers.

Field to insert	Description
Session-Id	The SSL session ID
Session-Cipher-Name	The symmetric cipher suite
Session-Cipher-Key-Size	The symmetric cipher key size
Session-Cipher-Use-Size	The symmetric cipher use size
Session-Step-Up	TRUE if the server presented a stepup certificate and the client renegotiated the cipher; otherwise FALSE
Session-Initial-Cipher-Name	If Session-Step-Up is TRUE, the initially negotiated cipher name
Session-Initial-Cipher-Key-Size	If Session-Step-Up is TRUE, the initially negotiated cipher's key size
Session-Initial-Cipher-Use-Size	If Session-Step-Up is TRUE, the initially negotiated cipher's use size

Table 2-4 lists the commands available in HTTP header insertion configuration submode.

Table 2-4 HTTP Header Insertion Configuration Submode Command Descriptions

Syntax	Description
alias *user-defined-name standard-name*	Specifies the alias name of the header. **Note** You can configure only one alias per standard name. You cannot configure the same alias name for multiple standard names.
client-cert [**pem**]	Allows the back-end server to see the attributes of the client certificate that the SSL module has authenticated and approved. **Note** You can insert the headers listed below by entering the **client-cert** command, or you can send the entire client certificate in PEM format by entering the **client-cert pem** command. **Note** The client certificate headers, or the client certificate in PEM format, are inserted only if the policy's service is configured for client authentication. The root CA and intermediate CA certificates will not be inserted the when client certificate is inserted in the HTTP header.

Table 2-4 HTTP Header Insertion Configuration Submode Command Descriptions (continued)

Syntax	Description
client-ip-port	Inserts the client IP address and information about the client port into the HTTP header, allowing the server to see the client IP address and port.
custom *custom-string*	Inserts the *custom-string* header into the HTTP header. The maximum *custom-string* length is 239 characters. If this length is exceeded, an "Incomplete command" error will display. If the string includes spaces, you must enclose it in quotes ("").
prefix	Adds the *prefix-string* to the HTTP header to enable the server to identify the connections that come from the SSL module, not from other appliances
session	Passes information that is specific to an SSL connection to the back-end server as session headers.

Examples

This example shows how to enter the HTTP header insertion configuration submode:

```
ssl-proxy(config)# ssl-proxy context s1
ssl-proxy(config-context)# policy http-header test1
ssl-proxy(config-ctx-http-header-policy)#
```

This example shows how to allow the back-end server to see the attributes of the client certificate that the SSL module has authenticated and approved:

```
ssl-proxy(config-ctx-http-header-policy)# client-cert
ssl-proxy(config-ctx-http-header-policy)#
```

This example shows how to insert the client IP address and information about the client port into the HTTP header, allowing the server to see the client IP address and port:

```
ssl-proxy(config-ctx-http-header-policy)# client-ip-port
ssl-proxy(config-ctx-http-header-policy)#
```

This example shows how to insert the custom-string header into the HTTP header:

```
ssl-proxy(config-ctx-http-header-policy)# custom "SOFTWARE VERSION:3.1(1)"
ssl-proxy(config-ctx-http-header-policy)# custom "module:SSL MODULE - CATALYST 6500"
ssl-proxy(config-ctx-http-header-policy)# custom
type-of-proxy:server_proxy_1024_bit_key_size
ssl-proxy(config-ctx-http-header-policy)#
```

This example shows how to add the prefix-string into the HTTP header:

```
ssl-proxy(config-ctx-http-header-policy)# prefix SSL-OFFLOAD
ssl-proxy(config-ctx-http-header-policy)#
```

This example shows how to pass information that is specific to an SSL connection to the back-end server as session headers:

```
ssl-proxy(config-ctx-http-header-policy)# session
ssl-proxy(config-ctx-http-header-policy)#
```

This example shows how to create a header alias for the standard "session-cipher-name" header:

```
ssl-proxy(config-ctx-http-header-policy)# alias My-Session-Cipher session-cipher-name
```

In addition to the standard HTTP headers, the following header information is inserted:

This example shows how to remove fields used for http header insert if found:

```
ssl-proxy(config-ctx-http-header-policy)# pre-remove-http-header
ssl-proxy(config-ctx-http-header-policy)#
```


Note The alias name (My-Session-Cipher) is used instead of the standard name (session-cipher-name).

```
SSL-OFFLOAD-Client-IP:7.100.100.1
SSL-OFFLOAD-Client-Port:59008
SSL-OFFLOAD-SOFTWARE VERSION:3.1(1)
SSL-OFFLOAD-module:SSL MODULE - CATALYST 6500
SSL-OFFLOAD-type-of-proxy:server_proxy_1024_bit_key_size
SSL-OFFLOAD-Session-Id:33:FF:2C:2D:25:15:3C:50:56:AB:FA:5A:81:0A:EC:E9:00:00:0A:03:00:60:
 2F:30:9C:2F:CD:56:2B:91:F2:FF
SSL-OFFLOAD-My-Session-Cipher:RC4-SHA
SSL-OFFLOAD-Session-Cipher-Key-Size:128
SSL-OFFLOAD-Session-Cipher-Use-Size:128
SSL-OFFLOAD-Session-Step-Up:FALSE
SSL-OFFLOAD-Session-Initial-Cipher-Key-Size:
SSL-OFFLOAD-Session-Initial-Cipher-Name:
SSL-OFFLOAD-Session-Initial-Cipher-Use-Size:
SSL-OFFLOAD-ClientCert-Valid:1
SSL-OFFLOAD-ClientCert-Error:none
SSL-OFFLOAD-ClientCert-Fingerprint:1B:11:0F:E8:20:3F:6C:23:12:9C:76:C0:C1:C2:CC:85
SSL-OFFLOAD-ClientCert-Subject-CN:a
SSL-OFFLOAD-ClientCert-Issuer-CN:Certificate Manager
SSL-OFFLOAD-ClientCert-Certificate-Version:3
SSL-OFFLOAD-ClientCert-Serial-Number:0F:E5
SSL-OFFLOAD-ClientCert-Data-Signature-Algorithm:sha1WithRSAEncryption
SSL-OFFLOAD-ClientCert-Subject:OID.1.2.840.113549.1.9.2 = ste2-server.cisco.com +
OID.2.5.4.5 = B0FFF22E, CN = a, O = Cisco
SSL-OFFLOAD-ClientCert-Issuer:CN = Certificate Manager, OU = HSS, O = Cisco, L = San Jose,
ST = California, C = US
SSL-OFFLOAD-ClientCert-Not-Before:22:29:26 UTC Jul 30 2003
SSL-OFFLOAD-ClientCert-Not-After:07:00:00 UTC Apr 27 2006
SSL-OFFLOAD-ClientCert-Public-Key-Algorithm:rsaEncryption
SSL-OFFLOAD-ClientCert-RSA-Public-Key-Size:1024 bit
SSL-OFFLOAD-ClientCert-RSA-Modulus-Size:1024 bit
SSL-OFFLOAD-ClientCert-RSA-Modulus:B3:32:3C:5E:C9:D1:CC:76:FF:81:F6:F7:97:58:91:4D:B2:0E:
C1:3A:7B:62:63:BD:5D:F6:5F:68:F0:7D:AC:C6:72:F5:72:46:7E:FD:38:D3:A2:E1:03:8B:EC:F7:C9:9A:
80:C7:37:DA:F3:BE:1F:F4:5B:59:BD:52:72:94:EE:46:F5:29:A4:B3:9B:2E:4C:69:D0:11:59:F7:68:3A:
D9:6E:ED:6D:54:4E:B5:A7:89:B9:45:9E:66:0B:90:0B:B1:BD:F4:C8:15:12:CD:85:13:B2:0B:FE:7E:8D:
F0:D7:4A:98:BB:08:88:6E:CC:49:60:37:22:74:4D:73:1E:96:58:91
SSL-OFFLOAD-ClientCert-RSA-Exponent:00:01:00:01
SSL-OFFLOAD-ClientCert-X509v3-Authority-Key-Identifier:keyid=EE:EF:5B:BD:4D:CD:F5:6B:60:
9D:CF:46:C2:EA:25:7B:22:A5:08:00
SSL-OFFLOAD-ClientCert-X509v3-Basic-Constraints:
SSL-OFFLOAD-ClientCert-Signature-Algorithm:sha1WithRSAEncryption
SSL-OFFLOAD-ClientCert-Signature:87:09:C1:F8:86:C1:15:C5:57:18:8E:B3:0D:62:E1:0F:6F:D4:9D:
75:DA:5D:53:E2:C6:0B:73:99:61:BE:B0:F6:19:83:F2:E5:48:1B:D2:6C:92:83:66:B3:63:A6:58:B4:5C:
0E:5D:1B:60:F9:86:AF:B3:93:07:77:16:74:4B:C5
SSL-OFFLOAD-ClientCert-X509v3-Subject-Alternative-Name:
ipAddress=192.168.1.100,rfc822Name=my@other.com
SSL-OFFLOAD-ClientCert-X509v3-Key-Usage: Digital Signature,Non-Repudiation,Key
Encipherment,
Data Encipherment,Key Agreement,Key Cert Sign,CRL Signature,Encipher Only,Decipher Only
SSL-OFFLOAD-ClientCert-X509v3-Authority-Information-Access: Access Method=OCSP,Access
Location=http://ocsp.my.host/"
SSL-OFFLOAD-ClientCert-X509v3-CRL-Distribution-Points: http://myhost.com/myca.crl
```

Related Commands show ssl-proxy policy

policy ssl

To enter the SSL-policy configuration submode, use the **policy ssl** command. In the SSL-policy configuration submode, you can define the SSL policy for one or more SSL-proxy services.

policy ssl *ssl-policy-name*

Syntax Description

ssl-policy-name	SSL policy name.

Defaults

The defaults are as follows:

- **cipher** is all-strong.
- **close-protocol** is disabled.
- **session-caching** is enabled.
- **version** is all.
- **session-cache size** *size* is 262143 entries.
- **timeout session** *timeout* is 0 seconds.
- **timeout handshake** *timeout* is 0 seconds.
- **cert-req empty** is disabled.
- **tls-rollback** is disabled.
- **renegotiation** is disabled.

Command Modes

Context subcommand mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 1.2(1)	This command was changed to add the following subcommands: • **session-cache size** *size* • **timeout session** *timeout* [**absolute**]

Release	Modification
SSL Services Module Release 2.1(5)	This command was changed to add the following subcommands: • **cert-req empty** • **tls-rollback** [**current** | **any**]
SSL Services Module Release 3.1(1)	The **policy ssl** command (entered in context subcommand mode) replaces the **ssl-proxy policy ssl** command (entered in global subcommand mode). This command was changed to add the following submode commands: • **cipher rsa-exp-with-des40-cbc-sha** • **cipher rsa-exp-with-rc4-40-md5** • **cipher rsa-exp1024-with-des-cbc-sha** • **cipher rsa-exp1024-with-rc4-56-md5** • **cipher rsa-exp1024-with-rc4-56-sha** • **cipher rsa-with-null-md5** • **renegotiation volume** • **renegotiation interval** • **renegotiation wait-time** • **renegotiation optional**

Usage Guidelines

Each SSL-policy configuration submode command is entered on its own line.

Table 2-5 lists the commands available in SSL-policy configuration submode.

Table 2-5 SSL-Policy Configuration Submode Command Descriptions

Syntax	Description
cert-req empty	Allows you to specify that the SSL Services Module backend service always returns the certificate associated with the trustpoint and does not look for a CA-name match.
cipher-suite {**all** | **all-export** | **all-strong** | **rsa-exp-with-des40-cbc-sha** | **rsa-exp-with-rc4-40-md5** | **rsa-exp1024-with-des-cbc-sha** | **rsa-exp1024-with-rc4-56-md5** | **rsa-exp1024-with-rc4-56-sha** | **rsa-with-3des-ede-cbc-sha** | **rsa-with-des-cbc-sha** | **rsa-with-null-md5** | **rsa-with-rc4-128-md5** | **rsa-with-rc4-128-sha**}	Allows you to configure a list of cipher-suites acceptable to the proxy-server.
[**no**] **close-protocol** {**strict** | **none**}	Allows you to configure the SSL close-protocol behavior. Use the **no** form of this command to disable close protocol.
default {**cipher** | **close-protocol** | **session-cache** | **version**}	Sets a command to its default settings.
exit	Exits from SSL-policy configuration submode.

Table 2-5 SSL-Policy Configuration Submode Command Descriptions (continued)

Syntax	Description
help	Provides a description of the interactive help system.
renegotiation volume *size*	Allows you to enable autorenegotiation and specifies the data volume size (in kilobytes). When the encrypted or decrypted data amount exceeds this size, the SSL Services Module sends a renegotiation request. This setting is disabled by default. The valid range is from 1024 to 1073741824 kilobytes.
renegotiation interval *time*	Allows you to enable autorenegotiation and specifies the interval (in seconds). After the set interval, the SSL Services Module sends an renegotiation request. This setting is disabled by default. The valid range is from 60 to 86400 seconds.
renegotiation wait-time *time*	(Optional) When you enable autorenegotiation, this command specifies the amount of time (in seconds) that the SSL Services Module waits for the peer to respond to the renegotiation request. The default is 100 seconds. The valid range is from 10 to 300 seconds.
renegotiation optional	(Optional) When you enable autorenegotiation, the SSL Services Module allows the session to continue if the peer does not respond to the renegotiation request after timeout. This setting is disabled by default and the session is disconnected after timeout.
[**no**] **session-cache**	Allows you to enable the session-caching feature. Use the **no** form of this command to disable session caching.
session-cache size *size*	Specifies the maximum number of session entries to be allocated for a given service; valid values are from 1 to 262143 entries.
timeout handshake *timeout*	Allows you to configure how long the module keeps the connection in the handshake phase; valid values are from 0 to 65535 seconds.
timeout session *timeout* [**absolute**]	Allows you to configure the session timeout. The syntax description is as follows: • *timeout*—Session timeout; valid values are from 0 to 72000 seconds. • **absolute**—(Optional) The session entry is not removed until the configured timeout has completed.
tls-rollback [**current** | **any**]	Allows you to specify if the SSL protocol version number in the TLS/SSL premaster secret message is either the maximum version or the negotiated version (**current**) or if the version is not checked (**any**).
version {**all** | **ssl3** | **tls1**}	Allows you to set the version of SSL to one of the following: • **all**—Both SSL3 and TLS1 versions are used. • **ssl3**—SSL version 3 is used. • **tls1**—TLS version 1 is used.

You can define the SSL policy templates using the **policy ssl** *ssl-policy-name* command and associate a SSL policy with a particular proxy server using the proxy server configuration CLI. The SSL policy template allows you to define various parameters that are associated with the SSL handshake stack.

When you enter the **close-notify strict** command, **t**he SSL Services Module sends a close-notify alert message to the SSL peer, and the SSL Services Module expects a close-notify alert message from the SSL peer. If the SSL Services Module does not receive a close-notify alert, SSL resumption is not allowed for that session.

When you enter the **close-notify none** command, **t**he SSL Services Module does not send a close-notify alert message to the SSL peer, and the SSL Services Module does not expect a close-notify alert message from the SSL peer. The SSL Services Module preserves the session information so that SSL resumption can be used for future SSL connections.

When close-notify is disabled (default), the SSL Services Module sends a close-notify alert message to the SSL peer; however, the SSL peer does not expect a close-notify alert before removing the session. Whether the SSL peer sends the close-notify alert or not, the session information is preserved allowing session resumption for future SSL connections.

The cipher-suite names follow the same convention as the existing SSL stacks.

The cipher-suites that are acceptable to the proxy-server are as follows:

- all-export—All export ciphers
- all-strong—All strong ciphers (default)
- all—All supported ciphers
- RSA-WITH-3DES-EDE-CBC-SHA—RSA with 3des-sha
- RSA-WITH-DES-CBC-SHA—RSA with des-sha
- RSA-WITH-RC4-128-MD5—RSA with rc4-md5
- RSA-WITH-RC4-128-SHA—RSA with rc4-sha
- RSA-EXP-WITH-DES40-CBC-SHA—RSA export with des40-sha
- RSA-EXP-WITH-RC4-40-MD5—RSA export with rc4-md5
- RSA-EXP1024-WITH-DES-CBC-SHA—RSA export1024 with des-sha
- RSA-EXP1024-WITH-RC4-56-MD5—RSA export1024 with rc4-md5
- RSA-EXP1024-WITH-RC4-56-SHA—RSA export1024 with rc4-sha
- RSA-WITH-NULL-MD5—RSA with null-md5

If you enter the **timeout session** *timeout* **absolute** command, the session entry is kept in the session cache for the configured timeout before it is cleaned up. If the session cache is full, the timers are active for all the entries, the **absolute** keyword is configured, and all further new sessions are rejected.

If you enter the **timeout session** *timeout* command without the **absolute** keyword, the specified timeout is treated as the maximum timeout and a best-effort attempt is made to keep the session entry in the session cache. If the session cache runs out of session entries, the session entry that is currently being used is removed for incoming new connections.

When you enter the **cert-req empty** command, the SSL Services Module back-end service always returns the certificate associated with the trustpoint and does not look for a CA-name match. By default, the SSL Services Module always looks for a CA-name match before returning the certificate. If the SSL server does not include a CA-name list in the certificate request during client authentication, the handshake fails.

By default, the SSL Services Module uses the maximum supported SSL protocol version (SSL2.0, SSL3.0, or TLS1.0) in the ClientHello message. Enter the **tls-rollback** [**current** | **any**] command if the SSL client uses the negotiated version instead of the maximum supported version (as specified in the ClientHello message).

When you enter the **tls-rollback current** command, the SSL protocol version can be either the maximum supported version or the negotiated version.

When you enter the **tls-rollback any** command, the SSL protocol version is not checked at all.

Examples

This example shows how to enter the SSL-policy configuration submode:

```
ssl-proxy(config)# ssl-proxy context s1
ssl-proxy(config-context)# policy ssl sslpl1
ssl-proxy (config-ctx-ssl-policy)#
```

This example shows how to define the cipher suites that are supported for the SSL-policy:

```
ssl-proxy (config-ctx-ssl-policy)# cipher RSA_WITH_3DES_EDE_CBC_SHA
ssl-proxy (config-ctx-ssl-policy)#
```

This example shows how to enable the SSL-session closing protocol and configure the strict closing protocol behavior:

```
ssl-proxy (config-ctx-ssl-policy)# close-protocol strict
ssl-proxy (config-ctx-ssl-policy)#
```

This example shows how to disable the SSL-session closing protocol:

```
ssl-proxy (config-ctx-ssl-policy)# no close-protocol
ssl-proxy (config-ctx-ssl-policy)#
```

These examples shows how to set a given command to its default setting:

```
ssl-proxy (config-ctx-ssl-policy)# default cipher
ssl-proxy (config-ctx-ssl-policy)# default close-protocol
ssl-proxy (config-ctx-ssl-policy)# default session-cache
ssl-proxy (config-ctx-ssl-policy)# default version
ssl-proxy (config-ctx-ssl-policy)#
```

This example shows how to enable a session cache:

```
ssl-proxy (config-ctx-ssl-policy)# session-cache
ssl-proxy (config-ctx-ssl-policy)#
```

This example shows how to disable a session cache:

```
ssl-proxy (config-ctx-ssl-policy)# no session-cache
ssl-proxy (config-ctx-ssl-policy)#
```

This example shows how to set the maximum number of session entries to be allocated for a given service:

```
ssl-proxy (config-ctx-ssl-policy)# session-cache size 22000
ssl-proxy (config-ctx-ssl-policy)#
```

This example shows how to configure the session timeout to absolute:

```
ssl-proxy (config-ctx-ssl-policy)# timeout session 30000 absolute
ssl-proxy (config-ctx-ssl-policy)#
```

These examples show how to enable the support of different SSL versions:

```
ssl-proxy (config-ctx-ssl-policy)# version all
ssl-proxy (config-ctx-ssl-policy)# version ssl3
ssl-proxy (config-ctx-ssl-policy)# version tls1
ssl-proxy (config-ctx-ssl-policy)#
```

Related Commands show ssl-proxy stats
show ssl-proxy stats ssl

policy tcp

To enter the proxy policy TCP configuration submode, use the **policy tcp** command. In proxy-policy TCP configuration submode, you can define the TCP policy templates.

policy tcp *tcp-policy-name*

Syntax Description

tcp-policy-name	TCP policy name.

Defaults

The defaults are as follows:

- **buffer-share rx** is 32768 bytes.
- **buffer-share tx** is 32768 bytes.
- **delayed-ack-threshold** is 2.
- **delayed-ack-timeout** is 200 seconds.
- **mss** is 1460 bytes.
- **nagle** is enabled.
- **timeout syn** is 75 seconds.
- **timeout reassembly** is 60 seconds.
- **timeout inactivity** is 600 seconds.
- **timeout fin-wait** is 600 seconds.
- **tos carryover** is disabled.

Command Modes

Context subcommand mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 1.2(1)	This command was changed to add the **timeout reassembly** *time* subcommand.
SSL Services Module Release 2.1(4)	This command was changed to add the **tos carryover** subcommand.

Release	Modification
SSL Services Module Release 3.1(1)	The **policy tcp** command (entered in context subcommand mode) replaces the **ssl-proxy policy tcp** command (entered in global subcommand mode). This command was changed to add the following submode commands: • **forced-ack** • **nagle**
SSL Services Module Release 3.1(5)	This command was changed to add the following submode command: • **timeout time-wait**

Usage Guidelines

After you define the TCP policy, you can associate the TCP policy with a proxy server using the proxy-policy TCP configuration submode commands.

Each proxy-policy TCP configuration submode command is entered on its own line.

Table 2-6 lists the commands that are available in proxy-policy TCP configuration submode.

Table 2-6 Proxy-policy TCP Configuration Submode Command Descriptions

Syntax	Description
[**no**] **buffer-share rx** *buffer-limit-in-bytes*	Allows you to configure the maximum size of the receive buffer share per connection; valid values are from 8192 to 262144. Use the **no** form of this command to return to the default setting. **Note** When large encrypted files are transferred by the module, the receive buffer size must be at least the maximum SSL record size of 16384 bytes for reassembly of the SSL record. We recommend a receive buffer size of at least 20000 bytes for optimal performance.
[**no**] **buffer-share tx** *buffer-limit-in-bytes*	Allows you to configure the maximum size of the transmit buffer share per connection; valid values are from 8192 to 262144. Use the **no** form of this command to return to the default setting. **Note** When large encrypted files are transferred by the module, the transmit buffer size must be at least the maximum SSL record size of 16384 bytes for reassembly of the SSL record. We recommend a transmit buffer size of at least 20000 bytes for optimal performance.
default	Sets a command to its default settings.
delayed-ack-threshold *delay*	Allows you to configure the delayed ACK threshold. The default is 2. The valid range is from 1 to 10.
delayed-ack-timeout *timer*	Allows you to configure the delayed ACK timeout. The default is 200 seconds. The valid range is from 50 to 500 seconds.
exit	Exits from proxy-service configuration submode.
forced-ack	Allows you to enable the forced-ACK algorithm.
help	Provides a description of the interactive help system.
[**no**] **mss** *max-segment-size-in-bytes*	Allows you to configure the maximum segment size that the connection identifies in the generated SYN packet; valid values are from 64 to 1460. Use the **no** form of this command to return to the default setting.

Table 2-6 Proxy-policy TCP Configuration Submode Command Descriptions (continued)

Syntax	Description
[**no**] **nagle**	Allows you to enable or disable the Nagle algorithm, which combines many small packets for more efficient transmission. Nagle is enabled by default.
[**no**] **timeout fin-wait** *timeout-in-seconds*	Allows you to configure the FIN wait timeout; valid values are from 75 to 600 seconds. Use the **no** form of this command to return to the default setting.
[**no**] **timeout inactivity** *timeout-in-seconds*	Allows you to configure the inactivity timeout; valid values are from 0 to 960 seconds. This command allows you to set the aging timeout for an idle connection and helps protect the connection resources. Use the **no** form of this command to return to the default setting.
[**no**] **timeout syn** *timeout-in-seconds*	Allows you to configure the connection establishment timeout; valid values are from 5 to 75 seconds. Use the **no** form of this command to return to the default setting.
[**no**] **timeout reassembly** *time*	Allows you to configure the amount of time in seconds before the reassembly queue is cleared; valid values are from 0 to 960 seconds (0 = disabled). If the transaction is not complete within the specified time, the reassembly queue is cleared and the connection is dropped. Use the **no** form of this command to return to the default setting.
[**no**] **tos carryover**	Forwards the type of service (ToS) value to all packets within a flow. **Note** If the policy is configured as a server TCP policy, the ToS value is sent from the server to the client. If the policy is configured as a virtual policy, the ToS value is sent from the client to the server. **Note** The ToS value needs to be learned before it can be propagated. For example, when a ToS value is configured to be propagated from the server to client connection, the server connection must be established before the value is learned and propagated. Therefore, some of the initial packets will not carry the ToS value.

Usage Guidelines

TCP commands that you enter on the SSL Services Module can apply either globally or to a particular proxy server.

You can configure a different maximum segment size for the client side and the server side of the proxy server.

The TCP policy template allows you to define parameters that are associated with the TCP stack.

You can either enter the **no** form of the command or use the **default** keyword to return to the default setting.

Examples

This example shows how to enter the proxy-policy TCP configuration submode:

```
ssl-proxy(config)# ssl-proxy context s1
ssl-proxy(config-context)# ssl-proxy policy tcp tcppl1
ssl-proxy(config-ctx-tcp-policy)#
```

These examples show how to set a given command to its default value:

```
ssl-proxy (config-ctx-tcp-policy)# default timeout fin-wait
```

```
ssl-proxy (config-ctx-tcp-policy)# default inactivity-timeout
ssl-proxy (config-ctx-tcp-policy)# default buffer-share rx
ssl-proxy (config-ctx-tcp-policy)# default buffer-share tx
ssl-proxy (config-ctx-tcp-policy)# default mss
ssl-proxy (config-ctx-tcp-policy)# default timeout syn
ssl-proxy (config-ctx-tcp-policy)#
```

This example shows how to define the FIN-wait timeout in seconds:

```
ssl-proxy (config-ctx-tcp-policy)# timeout fin-wait 200
ssl-proxy (config-ctx-tcp-policy)#
```

This example shows how to define the inactivity timeout in seconds:

```
ssl-proxy (config-ctx-tcp-policy)# timeout inactivity 300
ssl-proxy (config-ctx-tcp-policy)#
```

This example shows how to define the maximum size for the receive buffer configuration:

```
ssl-proxy (config-ctx-tcp-policy)# buffer-share rx 16384
ssl-proxy (config-ctx-tcp-policy)#
```

This example shows how to define the maximum size for the transmit buffer configuration:

```
ssl-proxy (config-ctx-tcp-policy)# buffer-share tx 13444
ssl-proxy (config-ctx-tcp-policy)#
```

This example shows how to define the maximum size for the TCP segment:

```
ssl-proxy (config-ctx-tcp-policy)# mss 1460
ssl-proxy (config-ctx-tcp-policy)#
```

This example shows how to define the initial connection (SYN)-timeout value:

```
ssl-proxy (config-ctx-tcp-policy)# timeout syn 5
ssl-proxy (config-ctx-tcp-policy)#
```

This example shows how to define the reassembly-timeout value:

```
ssl-proxy (config-ctx-tcp-policy)# timeout reassembly 120
ssl-proxy (config-ctx-tcp-policy)#
```

This example shows how to carryover the ToS value to all packets within a flow:

```
ssl-proxy (config-ctx-tcp-policy)# tos carryover
ssl-proxy (config-ctx-tcp-policy)#
```

Related Commands

show ssl-proxy policy

policy url-rewrite

To enter the URL rewrite configuration submode, use the **policy url-rewrite** command. In URL rewrite configuration submode, you can define the URL-rewrite content policy that is applied to the payload.

policy url-rewrite *url-rewrite-policy-name*

Syntax Description

url-rewrite-policy-name	URL rewrite policy name.

Defaults

This command has no default settings.

Command Modes

Context subcommand mode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The **policy url-rewrite** command (entered in context subcommand mode) replaces the **ssl-proxy policy url-rewrite** command (entered in global subcommand mode).

Usage Guidelines

URL rewrite allows you to rewrite redirection links only.

A URL rewrite policy consists of up to 32 rewrite rules for each SSL proxy service.

Table 2-7 lists the commands that are available in proxy-policy configuration submode.

Table 2-7 Proxy-policy Configuration Submode Command Descriptions

default	Sets a command to its default settings.
exit	Exits from proxy-policy configuration submode.
help	Provides a description of the interactive help system.
[**no**] **url** *url-string* [**clearport** *port-number* | **sslport** *port-number*]	Allows you to configure the URL string to be rewritten. Use the **no** form of this command to remove the policy.

url-string—Specifies the host portion of the URL link to be rewritten; it can have a maximum of 251 characters. You can use the asterisk (*) wildcard only as a prefix or a suffix of a *hostname* in a rewrite rule. For example, you can use the *hostname* in one of the following ways:

- www.cisco.com
- *.cisco.com
- wwwin.cisco.*

clearport *port-number*—(Optional) Specifies the port portion of the URL link that is to be rewritten; valid values are from 1 to 65535.

sslport *port-number*—(Optional) Specifies the *port* portion of the URL link that is to be written; valid values are from 1 to 65535.

Enter the **no** form of the command to remove the policy.

Note When a server includes the default HTTP port number 80 in a URL redirect (for example, www.example.com:80), then the **url** command must be configured in the same manner (for example, **url www.example.com:80**). Non-standard port numbers need not be configured as part of the URL, but may instead by configured using the **clearport** keyword.

Examples

This example shows how to enter the URL rewrite configuration submode for the test1 policy:

```
ssl-proxy(config)# ssl-pro context s1
ssl-proxy(config-context)# ssl-proxy policy url-rewrite test1
ssl-proxy(config-ctx-url-rewrite-policy#
```

This example shows how to define the URL rewrite policy for the test1 policy:

```
ssl-proxy(config)# ssl-pro context s1
ssl-proxy(config-context)# ssl-proxy policy url-rewrite test1
ssl-proxy(config-ctx-url-rewrite-policy# url www.cisco.com clearport 80 sslport 443
ssl-proxy(config-ctx-url-rewrite-policy#
```

This example shows how to delete the URL rewrite policy for the test1 policy:

```
ssl-proxy(config)# ssl-pro context s1
ssl-proxy(config-context)# ssl-proxy policy url-rewrite test1
ssl-proxy(config-ctx-url-rewrite-policy# no url www.cisco.com clearport 80 sslport 443
ssl-proxy(config-ctx-url-rewrite-policy#
```

Related Commands

show ssl-proxy policy

pool ca

To enter the certificate authority pool configuration submode, use the **pool ca** command. In the certificate authority pool configuration submode, you can configure a certificate authority pool, which lists the CAs that the module can trust.

pool ca *ca-pool-name*

Syntax Description

ca-pool-name	Certificate authority pool name.

Defaults

This command has no arguments or keywords.

Command Modes

Context subcommand mode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The **pool ca** command (entered in context subcommand mode) replaces the **ssl-proxy pool ca** command (entered in global subcommand mode).

Usage Guidelines

Enter each certificate-authority pool configuration submode command on its own line.

Table 2-8 lists the commands that are available in certificate-authority pool configuration submode.

Table 2-8 Proxy-policy TCP Configuration Submode Command Descriptions

Syntax	Description
ca	Configures a certificate authority. The available subcommand is as follows: **trustpoint** *ca-trustpoint-name*—Configures a certificate-authority trustpoint. Use the **no** form of this command to return to the default setting.
default	Sets a command to its default settings.
exit	Exits from proxy-service configuration submode.
help	Allows you to configure the connection-establishment timeout; valid values are from 5 to 75 seconds. Use the **no** form of this command to return to the default setting.

Examples

This example shows how to add a certificate-authority trustpoint to a pool:

```
ssl-proxy(config)# ssl-proxy context s1
ssl-proxy(config-context)# pool ca test1
ssl-proxy(config-ctx-ca-pool)# ca trustpoint test20
ssl-proxy(config-ctx-ca-pool)#
```

service

To enter the proxy-service configuration submode, use the **service** command.

service *ssl-proxy-name* [**client**]

Syntax Description

ssl-proxy-name	SSL proxy name.
client	(Optional) Allows you to configure the SSL-client proxy services. See the **service client** command.

Defaults

Server NAT is enabled, and client NAT is disabled.

Command Modes

Context subcommand mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 2.1(1)	This command was changed to add the following submode commands: • **authenticate** • **policy urlrewrite** *policy-name* • **trusted-ca** *ca-pool-name* • **sslv2**—See the **server ipaddr** subcommand.
SSL Services Module Release 3.1(1)	The **service** command (entered in context subcommand mode) replaces the **ssl-proxy service** command (entered in global subcommand mode). This command was changed to add the following submode commands: • **policy health-probe tcp** *policy-name* • **policy http-header** *policy-name*

Usage Guidelines

You cannot use the same *service_name* for both the server proxy service and the client proxy service.

In proxy-service configuration submode, you can configure the virtual IP address and port that is associated with the proxy service and the associated target IP address and port. You can also define TCP and SSL policies for both the client side (beginning with the **virtual** keyword) and the server side of the proxy (beginning with the **server** keyword).

In client proxy-service configuration submode, you specify that the proxy service accept clear-text traffic, encrypt it into SSL traffic, and forward it to the back-end SSL server.

In most cases, all of the SSL-server-proxy configurations that are performed are also valid for the SSL-client-proxy configuration, except for the following:

- You must configure a certificate for the SSL-server-proxy but you do not have to configure a certificate for the SSL-client-proxy. If you configure a certificate for the SSL-client-proxy, that certificate is sent in response to the certificate request message that is sent by the server during the client-authentication phase of the handshake protocol.
- The SSL policy is attached to the **virtual** subcommand for the SSL server proxy service; whereas, the SSL policy is attached to the **server** subcommand for the SSL client proxy service.

Enter each proxy-service or proxy-client configuration submode command on its own line.

Table 2-9 lists the commands that are available in proxy-service or proxy-client configuration submode.

Table 2-9 Proxy-service Configuration Submode Command Descriptions

Syntax	Description
authenticate verify {**all** | **signature-only**}	Configures the method for certificate verification. You can specify the following: • **all**—Verifies CRLs and signature authority. • **signature-only**—Verifies the signature only.
certificate rsa general-purpose trustpoint *trustpoint-name*	Configures the certificate with RSA general-purpose keys and associates a trustpoint to the certificate.
default {**certificate** | **inservice** | **nat** | **server** | **virtual**}	Sets a command to its default settings.
description	Allows you to enter a description for proxy service.
exit	Exits from proxy-service or proxy-client configuration submode.
help	Provides a description of the interactive help system.
inservice	Declares a proxy server or client as administratively up.
nat {**server** | **client**}{*natpool-name*}	Specifies the usage of either server NAT or client NAT for the server-side connection that is opened by the SSL Services Module.
policy health-probe tcp *policy-name*	Applies a TCP health probe policy to a proxy server.
policy http-header *policy-name*	Applies an HTTP header insertion policy to a proxy server.
policy urlrewrite *policy-name*	Applies a URL rewrite policy to a proxy server.
server ipaddr *ip-addr* **protocol** *protocol* **port** *portno* [**sslv2**]	Defines the IP address of the target server for the proxy server. You can also specify the port number and the transport protocol. The target IP address can be a virtual IP address of an SLB device or a real IP address of a web server. The **sslv2** keyword specifies the server that is used for handling SSL version 2 traffic.
server policy tcp *server-side-tcp-policy-name*	Applies a TCP policy to the server side of a proxy server. You can specify the port number and the transport protocol.
trusted-ca *ca-pool-name*	Applies a trusted certificate authenticate configuration to a proxy server.
virtual ipaddr *ip-addr* **protocol** *protocol* **port** *portno* [**secondary**]	Defines the virtual IP address of the virtual server to which the STE is proxying. You can also specify the port number and the transport protocol. The valid values for *protocol* are **tcp**; valid values for *portno* is from 1 to 65535. The **secondary** keyword (optional) prevents the STE from replying to the ARP request coming to the virtual IP address.

Table 2-9 Proxy-service Configuration Submode Command Descriptions (continued)

Syntax	Description
virtual policy ssl *ssl-policy-name*	Applies an SSL policy with the client side of a proxy server.
virtual policy tcp *client-side-tcp-policy-name*	Applies a TCP policy to the client side of a proxy server.
vlan *vlan*	Virtual Service VLAN configuration

Both secured and bridge mode between the Content Switching Module (CSM) and the SSL Services Module is supported.

Use the **secondary** keyword (optional) for bridge-mode topology.

Examples

This example shows how to enter the proxy-service configuration submode:

```
ssl-proxy (config)# ssl-proxy context s1
ssl-proxy (config-context)# service S6
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure the method for certificate verification:

```
ssl-proxy (config-ctx-ssl-proxy)# authenticate verify all
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure the certificate for the specified SSL-proxy services:

```
ssl-proxy (config-ctx-ssl-proxy)# certificate rsa general-purpose trustpoint tp1
ssl-proxy (config-ctx-ssl-proxy)#
```

These examples show how to set a specified command to its default value:

```
ssl-proxy (config-ctx-ssl-proxy)# default certificate
ssl-proxy (config-ctx-ssl-proxy)# default inservice
ssl-proxy (config-ctx-ssl-proxy)# default nat
ssl-proxy (config-ctx-ssl-proxy)# default server
ssl-proxy (config-ctx-ssl-proxy)# default virtual
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to apply a trusted-certificate authenticate configuration to a proxy server:

```
ssl-proxy (config-ctx-ssl-proxy)# trusted-ca test1
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure a virtual IP address for the specified virtual server:

```
ssl-proxy (config-ctx-ssl-proxy)# virtual ipaddr 207.59.100.20 protocol tcp port 443
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure the SSL policy for the specified virtual server:

```
ssl-proxy (config-ctx-ssl-proxy)# virtual policy ssl sslpl1
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure the TCP policy for the specified virtual server:

```
ssl-proxy (config-ctx-ssl-proxy)# virtual policy tcp tcppl1
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure a clear-text web server for the SSL Services Module to forward the decrypted traffic:

```
ssl-proxy (config-ctx-ssl-proxy)# server ipaddr 207.50.0.50 protocol tcp port 80
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure a TCP policy for the given clear-text web server:

```
ssl-proxy (config-ctx-ssl-proxy)# server policy tcp tcpp11
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure a NAT pool for the client address that is used in the server connection of the specified service SSL offload:

```
ssl-proxy (config-ctx-ssl-proxy)# nat client NP1
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to enable a NAT server address for the server connection of the specified service SSL offload:

```
ssl-proxy (config-ctx-ssl-proxy)# nat server
ssl-proxy (config-ctx-ssl-proxy)#
```

Related Commands

show ssl-proxy service

service client

To enter the client proxy-service configuration submode, use the **service client** command.

service *ssl-proxy-name* **client**

Syntax Description

ssl-proxy-name	SSL proxy service name.

Defaults

Client NAT is disabled.

Command Modes

Context subcommand mode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The **service client** command (entered in context subcommand mode) replaces the **ssl-proxy service client** command (entered in global subcommand mode). This command was changed to add the following submode commands: • **policy health-probe tcp** • **policy http-header**

Usage Guidelines

You cannot use the same *service_name* for both the server proxy service and the client proxy service.

In client proxy-service configuration submode, you specify that the proxy service accept clear-text traffic, encrypt it into SSL traffic, and forward it to the back-end SSL server.

In most cases, all of the SSL-server-proxy configurations that are performed are also valid for the SSL-client-proxy configuration, except for the following:

- You must configure a certificate for the SSL-server-proxy but you do not have to configure a certificate for the SSL-client-proxy. If you configure a certificate for the SSL-client-proxy, that certificate is sent in response to the certificate request message that is sent by the server during the client-authentication phase of the handshake protocol.
- The SSL policy is attached to the **virtual** subcommand for the SSL server proxy service; whereas, the SSL policy is attached to the **server** subcommand for the SSL client proxy service.

Each proxy-service or proxy-client configuration submode command is entered on its own line.

Table 2-10 lists the commands that are available in proxy-client configuration submode.

Table 2-10 Proxy-client Configuration Submode Command Descriptions

Syntax	Description
certificate rsa general-purpose trustpoint *trustpoint-name*	Configures the certificate with RSA general-purpose keys and associates a trustpoint to the certificate.
default {**certificate** | **inservice** | **nat** | **server** | **virtual**}	Sets a command to its default settings.
description	Allows you to enter a description for the proxy service.
exit	Exits from proxy-client configuration submode.
help	Provides a description of the interactive help system.
inservice	Declares a proxy client as administratively up.
nat {**server** | **client** *natpool-name*}	Specifies the usage of either server NAT or client NAT for the server-side connection that is opened by the SSL Services Module.
policy health-probe tcp *policy-name*	Applies a TCP health probe policy to a proxy server.
policy http-header *policy-name*	Applies an HTTP header insertion policy to a proxy server.
policy urlrewrite *policy-name*	Applies a URL rewrite policy to the proxy server.
server ipaddr *ip-addr* **protocol** *protocol* **port** *portno* [**sslv2**]	Defines the IP address of the target server for the proxy server. You can also specify the port number and the transport protocol. The target IP address can be a virtual IP address of an SLB device or a real IP address of a web server. The **sslv2** keyword enables SSL version 2.
server policy tcp *server-side-tcp-policy-name*	Applies a TCP policy to the server side of a proxy server. You can specify the port number and the transport protocol.
virtual ipaddr *ip-addr* **protocol** *protocol* **port** *portno* [**secondary**]	Defines the IP address of the target server for the proxy server. You can also specify the port number and the transport protocol. The target IP address can be a virtual IP address of an SLB device or a real IP address of a web server.
virtual policy ssl *ssl-policy-name*	Applies an SSL policy with the client side of a proxy server.
virtual policy tcp *client-side-tcp-policy-name*	Applies a TCP policy to the client side of a proxy server.
vlan *vlan*	Virtual Service VLAN configuration.

Both secured mode and bridge mode between the Content Switching Module (CSM) and the SSL Services Module are supported.

Use the **secondary** keyword (optional) for the bridge-mode topology.

Examples

This example shows how to enter the client proxy-service configuration submode:

```
ssl-proxy (config)# ssl-proxy context s1
ssl-proxy (config-context)# service S7 client
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure the certificate for the specified SSL-proxy services:

```
ssl-proxy (config-ctx-ssl-proxy)# certificate rsa general-purpose trustpoint tp1
ssl-proxy (config-ctx-ssl-proxy)#
```

These examples show how to set a specified command to its default value:

```
ssl-proxy (config-ctx-ssl-proxy)# default certificate
ssl-proxy (config-ctx-ssl-proxy)# default inservice
ssl-proxy (config-ctx-ssl-proxy)# default nat
ssl-proxy (config-ctx-ssl-proxy)# default server
ssl-proxy (config-ctx-ssl-proxy)# default virtual
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure a virtual IP address for the specified virtual server:

```
ssl-proxy (config-ctx-ssl-proxy)# virtual ipaddr 207.59.100.20 protocol tcp port 443
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure the SSL policy for the specified virtual server:

```
ssl-proxy (config-ctx-ssl-proxy)# virtual policy ssl sslpl1
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure the TCP policy for the specified virtual server:

```
ssl-proxy (config-ctx-ssl-proxy)# virtual policy tcp tcppl1
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure a clear-text web server for the SSL Services Module to forward the decrypted traffic:

```
ssl-proxy (config-ctx-ssl-proxy)# server ipaddr 207.50.0.50 protocol tcp port 80
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure a TCP policy for the given clear-text web server:

```
ssl-proxy (config-ctx-ssl-proxy)# server policy tcp tcppl1
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to configure a NAT pool for the client address that is used in the server connection of the specified service SSL offload:

```
ssl-proxy (config-ctx-ssl-proxy)# nat client NP1
ssl-proxy (config-ctx-ssl-proxy)#
```

This example shows how to enable a NAT server address for the server connection of the specified service SSL offload:

```
ssl-proxy (config-ctx-ssl-proxy)# nat server
ssl-proxy (config-ctx-ssl-proxy)#
```

Related Commands

show ssl-proxy service

show interfaces ssl-proxy

To display information about the configured subinterfaces, use the **show interfaces ssl-proxy** command.

show interfaces ssl-proxy 0.*subinterface*

Syntax Description

subinterface-number	Subinterface ID; valid values are from 0 to 4294967295.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Examples

This example shows how to display information about the configured subinterfaces:

```
ssl-proxy# show ionterfaces 0.3
SSL-Proxy0.3 is up, line protocol is up
  Hardware is STE interface, address is 0001.6445.c744 (bia 00e0.14c1.30e9)
  Internet address is 10.10.0.16/8
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  3.
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never

ssl-proxy#
```

Related Commands

policy tcp

show ssl-proxy buffers

To display information about TCP buffer usage, use the **show ssl-proxy buffers** command.

show ssl-proxy buffers

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display the buffer usage and other information in the TCP subsystem:

```
ssl-proxy# show ssl-proxy buffers
Buffers info for TCP module 1
    TCP data buffers used 2817 limit 88064
    TCP ingress buffer pool size 44032 egress buffer pool size 44032
    TCP ingress data buffers min-thresh 5636096 max-thresh 9017344
    TCP ingress data buffers used Current 0 Max 0
    TCP ingress buffer RED shift 9 max drop prob 10
    Conns consuming ingress data buffers 0
    Buffers with App 0
    TCP egress data buffers used Current 0 Max 0
    Conns consuming egress data buffers 0
    In-sequence queue bufs 0 OOO bufs 0
    Per-flow avg qlen 0 Global avg qlen 0

ssl-proxy#
```

Related Commands

policy tcp

show ssl-proxy certificate-history

To display information about the event history of the certificate, use the **show ssl-proxy certificate-history** command.

show ssl-proxy certificate-history [**service** [*name*]]

Syntax Description

service *name*	Displays all certificate records of a proxy service and (optionally) for a specific proxy service.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The **show ssl-proxy certificate-history** command displays these records:

- Service name
- Key pair name
- Generation or import time
- Trustpoint name
- Certificate subject name
- Certificate issuer name
- Serial number
- Date

A syslog message is generated for each record. The oldest records are deleted after the limit of 512 records is reached.

Examples

This example shows how to display the event history of all the certificate processing:

```
ssl-proxy# show ssl-proxy certificate-history
Record 1, Timestamp:00:00:51, 16:36:34 UTC Oct 31 2002
    Installed Server Certificate, Index 5
    Proxy Service:s1,  Trust Point:t3
    Key Pair Name:k3,  Key Usage:RSA General Purpose, Exportable
    Time of Key Generation:12:27:58 UTC Oct 30 2002
    Subject Name:OID.1.2.840.113549.1.9.2 = simpson5-2-ste.cisco.com,
OID.1.2.840.113549.1.9.8 = 207.79.1.9, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:5D3D1931000100000D99
    Validity Start Time:21:58:12 UTC Oct 30 2002
    End Time:22:08:12 UTC Oct 30 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record

  Record 2, Timestamp:00:01:06, 16:36:49 UTC Oct 31 2002
    Installed Server Certificate, Index 6
    Proxy Service:s5,  Trust Point:t10
    Key Pair Name:k10,  Key Usage:RSA General Purpose, Exportable
    Time of Key Generation:07:56:43 UTC Oct 11 2002
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.9, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:24BC81B7000100000D85
    Validity Start Time:22:38:00 UTC Oct 19 2002
    End Time:22:48:00 UTC Oct 19 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record

  Record 3, Timestamp:00:01:34, 16:37:18 UTC Oct 31 2002
    Installed Server Certificate, Index 7
    Proxy Service:s6,  Trust Point:t10
    Key Pair Name:k10,  Key Usage:RSA General Purpose, Exportable
    Time of Key Generation:07:56:43 UTC Oct 11 2002
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.9, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:24BC81B7000100000D85
    Validity Start Time:22:38:00 UTC Oct 19 2002
    End Time:22:48:00 UTC Oct 19 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record

  Record 4, Timestamp:00:01:40, 16:37:23 UTC Oct 31 2002
    Deleted Server Certificate, Index 0
    Proxy Service:s6,  Trust Point:t6
    Key Pair Name:k6,  Key Usage:RSA General Purpose, Not Exportable
    Time of Key Generation:00:28:28 UTC Mar 1 1993
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.8, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:5CB5CFD6000100000D97
    Validity Start Time:19:30:26 UTC Oct 30 2002
    End Time:19:40:26 UTC Oct 30 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record
% Total number of certificate history records displayed = 4
ssl-proxy#
```

This example shows how to display the certificate record for a specific proxy service:

```
ssl-proxy# show ssl-proxy certificate-history service s6
Record 3, Timestamp:00:01:34, 16:37:18 UTC Oct 31 2002
    Installed Server Certificate, Index 7
    Proxy Service:s6,  Trust Point:t10
    Key Pair Name:k10,  Key Usage:RSA General Purpose, Exportable
    Time of Key Generation:07:56:43 UTC Oct 11 2002
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.9, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:24BC81B7000100000D85
    Validity Start Time:22:38:00 UTC Oct 19 2002
    End Time:22:48:00 UTC Oct 19 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record

  Record 4, Timestamp:00:01:40, 16:37:23 UTC Oct 31 2002
    Deleted Server Certificate, Index 0
    Proxy Service:s6,  Trust Point:t6
    Key Pair Name:k6,  Key Usage:RSA General Purpose, Not Exportable
    Time of Key Generation:00:28:28 UTC Mar 1 1993
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.8, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:5CB5CFD6000100000D97
    Validity Start Time:19:30:26 UTC Oct 30 2002
    End Time:19:40:26 UTC Oct 30 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record
Total number of certificate history records displayed = 2
```

Related Commands

service

show ssl-proxy conn

To display the TCP connections from the SSL Services Module, use the **show ssl-proxy conn** command.

show ssl-proxy conn 4tuple [**local** {**ip** *local-ip-addr local-port*} [**remote** [{**ip** *remote-ip-addr* [**port** *remote-port*]} | {**port** *remote-port* [**ip** *remote-ip-addr*]}]]]

show ssl-proxy conn 4tuple [**local** {**port** *local-port*} [**remote** [{**ip** *remote-ip-addr* [**port** *remote-port*]} | {**port** *remote-port* [**ip** *remote-ip-addr*]}]]]

show ssl-proxy conn 4tuple [**local** {**remote** [{**ip** *remote-ip-addr* [**port** *remote-port*]} | {**port** *remote-port* [**ip** *remote-ip-addr*]}]]

show ssl-proxy conn module *module*

show ssl-proxy conn service *name* [**context** *name*] **module** [*module*]

Syntax Description

4tuple	Displays the TCP connections for a specific address.
local	(Optional) Displays the TCP connections for a specific local device.
ip *local-ip-addr*	IP address of a local device.
local-port	Port number of a local device.
remote	(Optional) Displays the TCP connections for a specific remote device.
ip *remote-ip-addr*	IP address of a remote device.
port *remote-port*	Port number of a remote device.
port *local-port*	(Optional) Displays the TCP connections for a specific local port.
module *module*	(Optional) Displays the information for a specific module. The available options for the module variable are as follows: • **all**—all CPUs • **fdu**—FDU CPU • **ssl1**—SSL1 CPU • **tcp1**—TCP1 CPU
service *name*	Displays the connections for a specific proxy service.
context *name*	(Optional) Displays information about the specified context.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	This command was changed to add the following keywords: • **context** *name* • **module** *module*

Usage Guidelines

The **show ssl-proxy conn** commanddisplays these records:

- Local Address
- Remote Address
- VLAN
- Conid
- Send-Q
- Recv-Q
- State

  The State record indicates the TCP state of the connection between the SSL Services Module and a remote device. The TCP states are described in the following table:

Table 2-11 TCP Connection State Descriptions

LISTEN	This module is wating for a request to initiate a TCP connection.
SYN_SEND	This module has sent a SYN packet to another device in order to initiate the opening of a TCP connection.
SYN_RECEIVED	This module has received a SYN packet from another device that is requesting to open a TCP connection.
ESTABLISHED or ESTAB	The three-way TCP handshake (SYN, SYN/ACK, ACK) has been completed and a TCP connection is now established between this module and another device.
FIN_WAIT_1	This module has sent a FIN packet to a connected device in order to close the TCP connection.
TIME_WAIT or TWAIT	This module has successfully completed a FIN sequence to close a TCP connection with a connected device. The connection will be held in this state for 30-120 seconds to receive any late packets.
CLOSE_WAIT	This module has received a FIN packet from a connected device that is requesting to close the TCP connection.
FIN_WAIT_2	After sending a FIN packet to a connected device in order to close the TCP connection, this module has received an ACK packet and is wating for a FIN packet.
LAST_ACK	At the request of a connected device, this module has closed the TCP connection and is waiting for a final ACK from the other device.

Table 2-11 TCP Connection State Descriptions (continued)

CLOSING	This module has actively closed the TCP connection and is waiting for a final ACK from the other device before entering the TIME_WAIT state.
CLOSED	A TCP connection has been closed with all wait times and acknowledgments completed.

Examples

These examples show different ways to display the TCP connection that is established from the SSL Services Module:

```
ssl-proxy# show ssl-proxy conn
Connections for TCP module 1
Local Address         Remote Address        VLAN Conid  Send-Q Recv-Q State
--------------------- --------------------- ---- ------ ------ ------ ------
2.0.0.10:4430         1.200.200.14:48582    2    0      0      0      ESTAB
1.200.200.14:48582    2.100.100.72:80       2    1      0      0      ESTAB

2.0.0.10:4430         1.200.200.14:48583    2    2      0      0      ESTAB
1.200.200.14:48583    2.100.100.72:80       2    3      0      0      ESTAB

2.0.0.10:4430         1.200.200.14:48584    2    4      0      0      ESTAB
1.200.200.14:48584    2.100.100.72:80       2    5      0      0      ESTAB

2.0.0.10:4430         1.200.200.14:48585    2    6      0      0      ESTAB
1.200.200.14:48585    2.100.100.72:80       2    7      0      0      ESTAB

2.0.0.10:4430         1.200.200.14:48586    2    8      0      0      ESTAB
1.200.200.14:48586    2.100.100.72:80       2    9      0      0      ESTAB

ssl-proxy# show ssl-proxy conn 4tuple local port 443
Connections for TCP module 1
Local Address         Remote Address        VLAN Conid  Send-Q Recv-Q State
--------------------- --------------------- ---- ------ ------ ------ ------
2.50.50.133:443       1.200.200.12:39728    2    113676 0      0      TWAIT
No Bound Connection

2.50.50.133:443       1.200.200.12:39729    2    113680 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:40599    2    113684 0      0      TWAIT
No Bound Connection

2.50.50.132:443       1.200.200.13:48031    2    114046 0      0      TWAIT
No Bound Connection

2.50.50.132:443       1.200.200.13:48032    2    114048 0      0      TWAIT
No Bound Connection

2.50.50.132:443       1.200.200.13:48034    2    114092 0      0      TWAIT
No Bound Connection

2.50.50.132:443       1.200.200.13:48035    2    114100 0      0      TWAIT
No Bound Connection
```

```
ssl-proxy# show ssl-proxy conn 4tuple remote ip 1.200.200.14
Connections for TCP module 1
Local Address         Remote Address        VLAN Conid  Send-Q Recv-Q State
--------------------- --------------------- ---- ------ ------ ------ ------
2.50.50.131:443       1.200.200.14:38814    2    58796  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38815    2    58800  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38817    2    58802  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38818    2    58806  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38819    2    58810  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38820    2    58814  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38821    2    58818  0      0      TWAIT
No Bound Connection

ssl-proxy# show ssl-proxy conn service iis1
Connections for TCP module 1
Local Address         Remote Address        VLAN Conid  Send-Q Recv-Q State
--------------------- --------------------- ---- ------ ------ ------ ------
2.50.50.131:443       1.200.200.14:41217    2    121718 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41218    2    121722 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41219    2    121726 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41220    2    121794 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41221    2    121808 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41222    2    121940 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41223    2    122048 0      0      TWAIT
No Bound Connection
```

show ssl-proxy context

To display context information, use the **show ssl-proxy context** command.

show ssl-proxy context [*name*]

Syntax Description

name	(Optional) Name of the context.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Examples

This example shows how to display all context information on the SSL Services Module:

```
ssl-proxy# show ssl-proxy context

Total number of contexts : 2

Context Name                              VRF             Num Proxies
------------                              ---             -----------
Default                                                   2
c1                                                        200
```

This example shows how to display specific context information on the SSL Services Module:

```
ssl-proxy# show ssl-proxy context Default

Context id              : 0
Number of proxies       : 2
Num max conns allowed   : 65536

Context 'Default' has the following service(s) configured..
s2
s3

ssl-proxy#
```

show ssl-proxy crash-info

To collect information about the software-forced reset from the SSL Services Module, use the **show ssl-proxy crash-info** command.

show ssl-proxy crash-info [**brief** | **details**]

Syntax Description

brief	(Optional) Collects a small subset of software-forced reset information, limited to processor registers.
details	(Optional) Collects the full set of software-forced reset information, including exception and interrupt stacks dump (this process can take up to 10 minutes to complete printing).

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to collect information about the software-forced reset:

```
ssl-proxy# show ssl-proxy crash-info

===== SSL SERVICE MODULE - START OF CRASHINFO COLLECTION =====


------------- COMPLEX 0 [FDU_IOS] ----------------------

NVRAM CHKSUM:0xEB28
NVRAM MAGIC:0xC8A514F0
NVRAM VERSION:1

++++++++++ CORE 0 (FDU) ++++++++++++++++++++++

   CID:0
   APPLICATION VERSION:2003.04.15 14:50:20 built for cantuc
   APPROXIMATE TIME WHEN CRASH HAPPENED:14:06:04 UTC Apr 16 2003
   THIS CORE DIDN'T CRASH
   TRACEBACK:222D48 216894
   CPU CONTEXT  ----------------------------

$0 :00000000, AT :00240008, v0 :5A27E637, v1 :000F2BB1
a0 :00000001, a1 :0000003C, a2 :002331B0, a3 :00000000
t0 :00247834, t1 :02BFAAA0, t2 :02BF8BB0, t3 :02BF8BA0
t4 :02BF8BB0, t5 :00247834, t6 :00000000, t7 :00000001
```

```
s0 :00000000, s1 :0024783C, s2 :00000000, s3 :00000000
s4 :00000001, s5 :0000003C, s6 :00000019, s7 :0000000F
t8 :00000001, t9 :00000001, k0 :00400001, k1 :00000000
gp :0023AE80, sp :031FFF58, s8 :00000019, ra :00216894
LO :00000000, HI :0000000A, BADVADDR :828D641C
EPC :00222D48, ErrorEPC :BFC02308, SREG :34007E03
Cause 0000C000 (Code 0x0):Interrupt exception

CACHE ERROR registers  -------------------

CacheErrI:00000000, CacheErrD:00000000
ErrCtl:00000000, CacheErrDPA:0000000000000000

   PROCESS STACK -----------------------------
      stack top:0x3200000

   Process stack in use:

   sp is close to stack top;

   printing 1024 bytes from stack top:

031FFC00:06405DE0 002706E0 0000002D 00000001  .@]`.'.`...-....
031FFC10:06405DE0 002706E0 00000001 0020B800  .@]`.'.`..... 8.
031FFC20:031FFC30 8FBF005C 14620010 24020004  ..|0.?.\.b..$...
...........
...........
...........
FFFFFFD0:00000000 00000000 00000000 00000000 ................
FFFFFFE0:00627E34 00000000 00000000 00000000 .b~4............
FFFFFFF0:00000000 00000000 00000000 00000006 ................


===== SSL SERVICE MODULE - END OF CRASHINFO COLLECTION =======
```

This example shows how to collect a small subset of software-forced reset information:

```
ssl-proxy# show ssl-proxy crash-info brief


===== SSL SERVICE MODULE - START OF CRASHINFO COLLECTION =====


------------- COMPLEX 0 [FDU_IOS] ----------------------

SKE CRASH INFO Error: wrong MAGIC # 0

CLI detected an error in FDU_IOS crash-info; wrong magic.

------------- COMPLEX 1 [TCP_SSL] ----------------------


Crashinfo fragment #0 from core 2 at offset 0 error:
Remote system reports wrong crashinfo magic.
Bad fragment received. Reception abort.

CLI detected an error in TCP_SSL crash-info;


===== SSL SERVICE MODULE - END OF CRASHINFO COLLECTION =======
```

show ssl-proxy mac address

To display the current MAC address, use the **show ssl-proxy mac address** command.

show ssl-proxy mac address

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display the current MAC address that is used in the SSL Services Module:

```
ssl-proxy# show ssl-proxy mac address
STE MAC address: 00e0.b0ff.f232
ssl-proxy#
```

show ssl-proxy natpool

To display information about the NAT pool, use the **show ssl-proxy natpool** command.

show ssl-proxy natpool [*name*][**context** *name*]

Syntax Description

name	(Optional) NAT pool name.
context *name*	(Optional) Context name.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	This command was changed to add the **context** *name* keyword.

Examples

This example shows how to display information for a specific NAT address pool that is configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy natpool
No context name provided, assuming context 'Default'...
natpool-name        start-ip         end-ip           netmask          use-count
n1                  207.57.110.1     207.57.110.8     255.0.0.0          2
ssl-proxy#
```

This example shows how to display information for a specific NAT address pool that is configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy natpool n1
No context name provided, assuming context 'Default'...
Start ip: 207.57.110.1
End ip: 207.57.110.8
netmask: 255.0.0.0
vlan associated with natpool: 2
SSL proxy services using this natpool:
          S2
          S3
Num of proxies using this natpool: 2
ssl-proxy#
```

Related Commands

natpool

show ssl-proxy policy

To display the configured SSL proxy policies, use the **show ssl-proxy policy** command.

show ssl-proxy policy {**health-probe tcp** [*name*] [**context** *name*] | **http-header** | **ssl** | **tcp** | **url-rewrite**} [*name*]

Syntax Description

health-probe tcp	Displays the configured TCP health probe policies.
name	(Optional) TCP health probe name.
context *name*	(Optional) Displays the TCP health probe policies in this context.
http-header	Displays the configured HTTP header policies.
ssl	Displays the configured SSL policies.
tcp	Displays the configured TCP policies.
url-rewrite	Displays the configured URL rewrite policies.
name	(Optional) Policy name.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 2.1(1)	This command was changed to include the **http-header** and **url-rewrite** keywords.
SSL Services Module Release 3.1(1)	This command was changed to add the **health-probe tcp** keyword.

Examples

This example shows how to display information about the HTTP header policy:

```
ssl-proxy# show ssl-proxy policy http-header h1
No context name provided, assuming context 'Default'...
Prefix                          SSL
Client Certificate Insertion    Not Enabled
Session Header Insertion        All
Client IP/Port Insertion        Not Enabled
Hdr #   Custom Header
 0   "a:"
 1   "b:"
 2   "c:"
 3   "d:"
 4   "e:"
 5   "f:"
```

```
 6   "g:"
 7   "h:"
 8   "i:"
 9   "j:"
10   "k:"
11   "l:"
12   "m:"
13   "n:"

Usage count of this policy: 0

ssl-proxy#
```

This example shows how to display policy information about a specific SSL policy that is configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy policy ssl ssl-policy1
No context name provided, assuming context 'Default'...

Cipher suites: (None configured, default ciphers included)
     rsa-with-rc4-128-md5
     rsa-with-rc4-128-sha
     rsa-with-des-cbc-sha
     rsa-with-3des-ede-cbc-sha
SSL Versions enabled:SSL3.0, TLS1.0
close protocol: default (close_notify sent but not expected from peer)
Session Cache:enabled
Session timeout: 72000 seconds
Renegotiation timeout: 100 seconds
Handshake timeout not configured (never times out)
TLS Rollback: default (version number rollback not allowed)
No. of policy users : 0

ssl-proxy#
```

This example shows how to display policy information about a specific TCP policy that is configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy policy tcp tcp-policy1
No context name provided, assuming context 'Default'...
MSS                1460
SYN timeout        75
Idle timeout       600
FIN wait timeout   75
Reassembly timeout 60
Persist timeout    0
Rx Buffer Share    32768
Tx Buffer Share    65536
TOS Carryover      Disabled
Delayed ACK timer  200
Delayed ACK Threshold 2
Nagle algorithm    Enabled
Forced ACK         Enabled

No. of policy users : 0

ssl-proxy#
```

This example shows how to display information about the URL rewrite policy:

```
ssl-proxy# show ssl-proxy policy url-rewrite urlrw-policy
No context name provided, assuming context 'Default'...
Rule URL                                    Clearport SSLport
   1 wwwin.cisco.com                           80        443
   2 www.cisco.com                             8080      444

Usage count of this policy: 0

ssl-proxy#
```

This example shows how to display information about the TCP health probe policy:

```
ssl-proxy# show ssl-proxy policy health-probe tcp
No context name provided, assuming context 'Default'...

TCP Health Probe Policy Name                        Usage-Count
      tcp-health                                              1
```

This example shows how to display information about the specified TCP health probe policy:

```
ssl-proxy# show ssl-proxy policy health-probe tcp tcp-health
No context name provided, assuming context 'Default'...

TCP Health Probe Details : tcp-health
Server Port number                80
Interval between probe            30
Interval between failed probe     60
TCP Connection open timeout       80
Maximum retries for success probe 3
No. of policy users               1
SSL proxy services using this policy:
    s3                                      Connected
Usage count of this policy: 1
```

Related Commands

policy health-probe tcp
policy http-header
policy ssl
policy tcp
policy url-rewrite

show ssl-proxy service

To display information about the configured SSL virtual service, use the **show ssl-proxy service** command.

show ssl-proxy service [*name*][**context** *name*]

Syntax Description

name	(Optional) Service name.
context *name*	(Optional) Displays service information for the specifed context name.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	This command was changed to add the **context** *name* keyword.

Examples

This example shows how to display all SSL virtual services that are configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy service
No context name provided, assuming context 'Default'...

Proxy Service Name  Context Name         Admin  Operation
                                         status status
s2                  Default              up     up
s3                  Default              up     up

ssl-proxy#
```

This example shows how to display a specific SSL virtual service that is configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy service S6
No context name provided, assuming context 'Default'...

Service id: 1, bound_service_id: 257
Virtual IP: 10.10.1.104, port: 443
Server IP: 10.10.1.100, port: 80
Virtual SSL Policy: SSL1_PLC
Server TCP Policy: nagle
TCP Health Probe Policy: tcp-health
```

```
Nat pool: n2
rsa-general-purpose certificate trustpoint: tptest
  Certificate chain for new connections:
    Certificate:
       Key Label: mytp, 1024-bit, not exportable
       Key Timestamp: 07:21:09 UTC Apr 20 2005
       Serial Number: 0FE5
    Root CA Certificate:
       Serial Number: 01
  Certificate chain complete
Context name: Default
Context Id  : 0
Admin Status: up
Operation Status: up

ssl-proxy#
```

This example shows how to display a specific SSL virtual service on a specific context that is configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy service s2 context c1
Service id: 214, bound_service_id: 470
Virtual IP: 10.12.0.2, port: 443
Server IP: 10.0.207.203, port: 80
TCP Health Probe Policy: h1
rsa-general-purpose certificate trustpoint: mytp
  Certificate chain for new connections:
    Certificate:
       Key Label: mytp, 1024-bit, not exportable
       Key Timestamp: 07:21:09 UTC Apr 20 2005
       Serial Number: 0FE5
    Root CA Certificate:
       Serial Number: 01
  Certificate chain complete
Context name: c1
Context Id  : 167
Admin Status: up
Operation Status: up

ssl-proxy#
```

Related Commands

service
service client

show ssl-proxy stats

To display information about the statistics counter, use the **show ssl-proxy stats** command.

show ssl-proxy stats [*type*]

Syntax Description

type	(Optional) Information type; valid values are **content**, **context, crypto**, **fdu**, **hdr**, **ipc**, **module**, **pki**, **service**, **ssl**, **tcp**, and **url**. See the "Usage Guidelines" section for additional information.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 1.2(1)	The output of the **show ssl-proxy stats** command was changed to include information about the session allocation failure and session limit-exceed table.
SSL Services Module Release 3.1(1)	This command was changed to add the following keywords: • **content** • **context** • **hdr** • **module** *module* • **url**

Usage Guidelines

The *type* values are defined as follows:

- **content**—Displays content scan object statistics.
- **context**—Displays context statistics information.
- **crypto**—Displays crypto statistics.
- **fdu**—Displays FDU statistics.
- **hdr**—Displays HTTP header insertion statistics.
- **ipc**—Displays IPC statistics.

- **module** *module*—Displays statistics for the specified module; module type includes the following:
  - **all**—all CPUs
  - **fdu**—FDU CPU
  - **ssl1**—SSL1 CPU
  - **tcp1**—TCP1 CPU
- **pki**—Displays PKI statistics.
- **service**—Displays proxy service statistics.
- **ssl**—Displays SSL detailed statistics.
- **tcp**—Displays TCP detailed statistics.
- **url**—Displays URL rewrite statistics.

Examples

This example shows how to display ssl-proxy statistics:

```
ssl-proxy# show ssl-proxy stats
TCP Statistics:
    Conns initiated     : 14415157      Conns accepted       : 14415157
    Conns established   : 27748020      Conns dropped        : 14414667
    Conns Allocated     : 14415157      Conns Deallocated    : 14415157
    Conns closed        : 28830314      SYN timeouts         : 1081918
    Idle timeouts       : 0             Total pkts sent      : 2621810445
    Data packets sent   : 4048216786    Data bytes sent      : 406938953
    Total Pkts rcvd     : 4175351636    Pkts rcvd in seq     : 4182198218
    Bytes rcvd in seq   : 2528520209

SSL Statistics:
    conns attempted     : 14415157      conns completed     : 14415157
    full handshakes     : 14415157      resumed handshakes  : 0
    active conns        : 0             active sessions     : 0
    renegs attempted    : 0             conns in reneg      : 0
    handshake failures  : 0             data failures       : 0
    fatal alerts rcvd   : 0             fatal alerts sent   : 0
    no-cipher alerts    : 0             ver mismatch alerts : 0
    no-compress alerts  : 0             bad macs received   : 0
    pad errors          : 0             session fails       : 0

FDU Statistics:
    IP Reass in progress: 0             Frag Svc full Drops : 0
    IPFlow create Drops : 0             TTL expired Drops   : 0
    IP Frag Drops       : 0             Frag reass complete : 0
    Frag nodes freed    : 0             IP Version Drops    : 0
    IP Addr Discards    : 0             Serv_Id Drops       : 4266052
    Conn Id Drops       : 0             Bound Conn Drops    : 0
    Vlan Id Drops       : 0             TCP HW Checksum     : 4179625097
    TCP SW Checksum     : 0             TCP Checksum Drops  : 0
    Hash Full Drops     : 0             Hash Alloc Fails    : 0
    Flow Creates        : 28830314      Flow Deletes        : 28830314
    Conn Id allocs      : 14415157      Conn Id deallocs    : 14415157
    Tagged Pkts Drops   : 0             Non-Tagg Pkts Drops : 0
    Add ipcs            : 3             Delete ipcs         : 0
    Disable ipcs        : 0             Enable ipcs         : 0
    Unsolicited ipcs    : 0             Duplicate Add ipcs  : 0
    IOS Broadcast Pkts  : 5330879       IOS Unicast Pkts    : 3752
    IOS Multicast Pkts  : 0             IOS Total Pkts      : 5334631
    IOS Congest Drops   : 0             SYN Discards        : 7400
    UDP datagrams Rcvd  : 0             UDP datagrams Sent  : 0
```

```
    UDP HW Checksum     : 0             UDP SW Checksum     : 0
    UDP Flow Misses     : 0             UDP Length Errors   : 0
    TCP 5-tuple reuse   : 0             FDU Reset Drops     : 0

ssl-proxy#
```

This example shows how to display ssl statistics:

```
ssl-proxy# show ssl-proxy stats ssl
SSL Statistics:
    conns attempted     : 14415157      conns completed     : 14415157
    conns in handshake  : 0             conns in data       : 0
    renegs attempted    : 0             conns in reneg      : 0
    active sessions     : 0             max handshake conns : 505
    rand bufs allocated : 803902        cached rand buf miss: 0
    current device q len: 0             max device q len    : 327
    sslv2 forwards      : 0             cert reqs processed : 0
    fatal alerts rcvd   : 0             fatal alerts sent   : 0
    stale packet drops  : 0             service_id discards : 0
    session reuses      : 0             hs handle in use    : 0
    netscape step-ups   : 0             SGC step-ups        : 0
    alloc msg received  : 14415157      delete  msg received: 14415157
    delayed conn delete : 57            timer expires       : 0
    multi timer expires : 0             callwheel NULL list : 0
    bad clnt session id : 0             expired session id  : 0

    SSL3 Statistics:
      full handshakes    : 5882754       resumed handshakes : 0
      handshake failures : 0             data failures      : 0
      bad macs received  : 0             pad errors         : 0
      conns established with cipher rsa-with-rc4-128-md5          : 0
      conns established with cipher rsa-with-rc4-128-sha          : 0
      conns established with cipher rsa-with-des-cbc-sha          : 0
      conns established with cipher rsa-with-3des-ede-cbc-sha     : 5882754
      conns established with cipher rsa-with-null-md5             : 0
      conns established with cipher rsa-exp1024-with-des-cbc-sha  : 0
      conns established with cipher rsa-exp1024-with-rc4-56-sha   : 0
      conns established with cipher rsa-exp1024-with-rc4-56-md5   : 0
      conns established with cipher rsa-exp-with-rc4-40-md5       : 0
      conns established with cipher rsa-exp-with-des40-cbc-sha    : 0

    TLS1 Statistics:
      full handshakes    : 8532403       resumed handshakes : 0
      handshake failures : 0             data failures      : 0
      bad macs received  : 0             pad errors         : 0
      conns established with cipher rsa-with-rc4-128-md5          : 8532403
      conns established with cipher rsa-with-rc4-128-sha          : 0
      conns established with cipher rsa-with-des-cbc-sha          : 0
      conns established with cipher rsa-with-3des-ede-cbc-sha     : 0
      conns established with cipher rsa-with-null-md5             : 0
      conns established with cipher rsa-exp1024-with-des-cbc-sha  : 0
      conns established with cipher rsa-exp1024-with-rc4-56-sha   : 0
      conns established with cipher rsa-exp1024-with-rc4-56-md5   : 0
      conns established with cipher rsa-exp-with-rc4-40-md5       : 0
      conns established with cipher rsa-exp-with-des40-cbc-sha    : 0

SSL error statistics:
    session alloc fails : 0             session limit exceed: 0
    handshake init fails: 0             renegotiation fails : 0
    no-cipher alerts    : 0             ver mismatch alerts : 0
    no-compress alerts  : 0             multi buf rec errors: 0
    ssl peer closes     : 0             non-ssl peer closes : 0
    unexpected record   : 0             rec formatting error: 0
    rsa pkcs pad errors : 0             premaster errors    : 0
```

```
    failed rsa reqs     : 0             failed random reqs  : 0
    failed key-material : 0             failed master-secret: 0
    failed md5 hash     : 0             failed tls exp key  : 0
    failed tls exp iv   : 0             failed srvr key exch: 0
    failed update hash  : 0             failed finish hash  : 0
    failed encrypts     : 0             failed decrypts     : 0
    bad record version  : 0             bad record size     : 0
    cert verify errors  : 0             unsupported certs   : 0
    conn aborted        : 0             empty cert records  : 0
    overload drops      : 0             hs limit exceeded   : 0
    hs handle mem fails : 0             conn reuse error    : 0
    dev invalid params  : 0             dev failed requests : 0
    dev timeout         : 0             dev busy            : 0
    dev cancelled       : 0             no dev fails        : 0
    dev resource fails  : 0             dev unknown errors  : 0
    dev conn ctx fails  : 0             dev cmd ctx fails   : 0
    mem alloc fails     : 0             buf alloc fails     : 0
    invalid cipher algo : 0             invalid hash algo   : 0
    unaligned buf addr  : 0             unaligned buf len   : 0
    internal error      : 0             unknown ipcs        : 0
    double free attempts: 0             alert-send fails    : 0

SSL Crypto Statistics:
    blocks encrypted    : 4138812209    blocks decrypted    : 347762407
    bytes encrypted     : 1789594329    bytes decrypted     : 1217509979
    crypto failures     : 0
    device dma errors   : 0
    PushMCR_nopkts      : 2532831724    PushMCR_pushed      : 0
    PushMCR_full        : 2115169602    PushMCR_push        : 146595780
    GetFreeMCR_busy     : 0             GetFreeMCR_dma_error: 0
    GetFreeMCR_no_rsrc  : 0             GetFreeMCR_success  : 45670677

SSL last 5 sec average Statistics:
    full handshakes     : 0             resumed handshakes  : 0
    handshake failures  : 0             data failures       : 0
    bytes encrypted     : 0             bytes decrypted     : 0

SSL last 1 min average Statistics:
    full handshakes     : 0             resumed handshakes  : 0
    handshake failures  : 0             data failures       : 0
    bytes encrypted     : 0             bytes decrypted     : 0

SSL last 5 min average Statistics:
    full handshakes     : 0             resumed handshakes  : 0
    handshake failures  : 0             data failures       : 0
    bytes encrypted     : 0             bytes decrypted     : 0

SSL PKI Statistics:
    number of malloc    : 1455663450    number of free      : 1455663198
    ssl buf allocated   : 7             ssl buf freed       : 1

    Peer Certificate Verify Statistics:
    cert approved       : 0             cert disapproved    : 0
    peer cert empty     : 0             total num of request: 0
    req being processed : 0             req pending         : 0
    longest queue       : 0             longest pending     : 0
    verify congestion   : 0             req dropped, q full : 0
    no memory for verify: 0             verify data error   : 0
    verify context error: 0             context delete error: 0
    timer expired error : 0             timer expired count : 0
    late verify result  : 0             timer turned on     : 0
    timer turned off    : 0             context created     : 0
    context deleted     : 0
```

```
    High Priority IPC:
    ipc request received: 1455663049    ipc request dropped : 0
    ipc req duplicated  : 0             ipc req fragment err: 0
    ipc req parm len err: 0             ipc req op code err : 0
    ipc req cert len err: 0             ipc response sent   : 1455663049
    ipc resp no memory  : 0             ipc resp no ssl buf : 0
    ipc buffer allocated: 0             ipc buffer freed    : 0
    ipc buf alloc failed: 0             ipc send msg failed : 0

    Normal Priority IPC:
    ipc buffer allocated: 0             ipc buffer freed    : 0
    ipc request sent    : 0             ipc request received: 0
    ipc buf alloc failed: 0             ipc send msg failed : 0
    ipc requests dropped: 0

    Subject_Name Allocation:
    subject_name allocs : 0             subject_name frees  : 0
    subject_name memory : 0

    Session Queue Sizes:
    ssl_free_sess_q_size: 262144        ssl_free_sess_active_timer_q_size: 0
    ssl_delete_conn_q_size: 0

SSL Queue Sizes:
    bcm_cmd_ctx_pool_size   : 64            bcm_asym_cmd_ctx_pool_sz: 9000
    bcm_info_pool_size      : 65538         buf_desc_free_q_size    : 94710
    cert_result_free_q_size : 11048         delete_conn_q_size      : 0
    event_q_size            : 0             free_conn_q_size        : 65536
    free_sess_q_size        : 262144        free_sess_active_tmr_qsz: 0
    global_pending_q_size   : 0             to_app_ctx_pool_size    : 512
    ste_asym_req_q_size     : 0             ste_free_req_ctx_pool_sz: 20480
    ste_sym_req_q_size      : 0             available ctx count     : 64
    ctx cleanup count       : 0             device reset count      : 0


SSL Random Buffer Info:
    psuedo_rand_req_pending : 0             rand_req_pending        : 0
    pseudo_rand_req_count   : 297
    curr_psuedo_rand_buf    : 0x0AEBF2A4    curr_rand_buf           : 0x0AEBF220
    psuedo_rand_buf_a       : 0x0AEBF278    psuedo_rand_buf_a_rx_sz : 3044
    psuedo_rand_buf_b       : 0x0AEBF2A4    psuedo_rand_buf_b_rx_sz : 3884
    rand_buf_a              : 0x0AEBF220    rand_buf_a_rx_size      : 4064
    rand_buf_b              : 0x0AEBF24C    rand_buf_b_rx_size      : 4064
```

This example shows how to display the TCP statistics:

```
ssl-proxy# show ssl-proxy stats tcp
TCP Statistics:
 Connection related :
    Initiated             : 14415157   Accepted             : 14415157
    Established           : 27748020   Dropped              : 14414667
    Dropped before est    : 1082294    Closed               : 28830314
    Persist timeout drops : 0          Rxmt timeout drops   : 0
    Current TIME-WAIT     : 0          Current ESTABLISHED  : 0
    Maximum TIME-WAIT     : 17254      Maximum ESTABLISHED  : 1911
    Conns Allocated       : 14415157   Conns Deallocated    : 14415157
    Conn Deletes sent     : 28830314   Credit Updates       : 0
    Credit Enable Req     : 0          Credit Disable Req   : 0
    Probe resets          : 0
 Timer related :
    RTT estimates         : 4272088584 RTT est. updates     : 4282749990
    delayed acks sent     : 16011985   FIN-WAIT2 timeouts   : 0
    Retransmit timeouts   : 6263673    Persist Timeouts     : 0
    SYN timeouts          : 1081918    Idle Timeouts        : 0
```

```
   Reassembly timeouts   : 0
Packet Transmit related :
   Total packets         : 2621810445 Data packets         : 4048216786
   Data bytes sent       : 406938953  Retransmitted pkts   : 1015590
   Retransmitted bytes   : 184932379  Ack only pkts        : 54472646
   Window probes         : 0          URG only pkts        : 0
   Window Update pkts    : 2764579114 Cntrl pkts (S/F/R)   : 48493321
   Tx TOS - normal       : 2621810161 Tx TOS - Min. Cost   : 0
   Tx TOS - max. rel.    : 0          Tx TOS - Max. thru.  : 0
   Tx TOS - min. delay   : 0          Tx TOS - invalid     : 0
Packet Receive related :
   Total packets         : 4175351636 In seq data pkts     : 4182198218
   In seq data bytes     : 2528520209 Bad Offset           : 0
   Too short             : 0          Dup-only data pkts   : 1180058
   Dup-only data bytes   : 1528138278 Part. dup. data pkts : 0
   Part. Dup. data bytes : 0          OOO data pkts        : 0
   OOO data bytes rcvd   : 0          Pkts after rx win    : 0
   Bytes after rx window : 0          Pkts after close     : 0
   Window Probes         : 0          Duplicate ACKs       : 16539602
   ACKs for unsent data  : 0          ACK-only pkts        : 1528035
   Bytes acked by acks   : 433284808  Window Update pkts   : 23
   PAWS dropped pkts     : 0          Hdr pred. ACKs       : 4109450673
   Hdr pred. data pkts   : 4123685217 TCB cache misses     : 1114110201
   3 dup-only pkts       : 9          Partial Acks         : 6
   Rx TOS - normal       : 4171511079 Rx TOS - Min. Cost   : 0
   Rx TOS - max. rel.    : 0          Rx TOS - Max. thru.  : 0
   Rx TOS - min. delay   : 0          Rx TOS - invalid     : 0
   Unrecognized Options  : 0          Unaligned MSS        : 0
   Unaligned Timestamp   : 0          Unaligned SACK       : 0
   Forced ACKs           : 0          RST ACK's sent       : 0
   Recycled Client Conns : 14415157   Recycled Server Conns: 14415157

Packet Drop statistics :
   Per-flow limit drops  : 0          Aggregate tail drops : 0
   Aggregate random drps : 0          Egress Bufpool drops : 0

Connection Drop/Close statistics :
   Active                : 13332005   Passive              : 81
   App closed early      : 0          Client Reuse         : 0
   Client RST Rcvd       : 1184       Server RST Rcvd      : 83
   App aborted client    : 1082001    App aborted server   : 13333042
   Unexp. SYNs           : 0          Server Refused       : 0
   Conn Bufpool Drops    : 0          Invalid MSS Drops    : 0
   User clear Drops      : 0          Unexp. Data Rcvd     : 0
   Server Reuse          : 0          Conn init failures   : 0
   SYN Timeout           : 1081918    Age Timeout          : 0
   Reass Timeout         : 0          FinWait2 Timeout     : 0
   Rexmit Timeout        : 0          Persist Timeout      : 0
   RST Closed            : 0          ACK Closed           : 0
   NOSYN Closed          : 0          MSS                  : 0
   Conn Pool Fails       : 0          No Buffers           : 0

Debug Statistics :
   Unaccounted Buffers   : 0          Invalid Conns        : 0
   Output Failures       : 0          Header Bufpool Fails : 0
   MAC channel Fails     : 0          DM Channel Fails     : 0
   Invalid App Opcodes   : 0          MAC Bufpool Fails    : 0
   MAC BufDesc Fails     : 0          Recycle Conn Fails   : 0
   DM chan congested     : 0          MAC chan congested   : 0
   Connid_alloc Deallocs : 0          Connid_alloc Failures: 0
   Connid_free Bad_Connid: 0          Connid_free Dups     : 0
   AppConnEntry GC Frees : 0          RST rcvd in SYN state: 0
   RST rcvd in EST state : 1150       RST rcvd in FW1 state: 3
   RST rcvd in FW2 state : 0          RST rcvd in CWT state: 57
```

```
    RST rcvd in CLG state : 0          RST rcvd in LCK state: 57
    Lcte Free Pool Count  : 262144     AppConn Free Pool Cnt: 65536
```

This example shows how to display the PKI statistics:

```
ssl-proxy# show ssl-proxy stats pki
Authentication request timeout: 180 seconds
Max in process: 50 (requests)
Max queued before dropping: 500 (requests)
Certificate Authentication & Authorization Statistics:
  Requests started: 0
  Requests finished: 0
  Requests pending to be processed: 0
  Requests waiting for CRL: 0
  Signature only requests: 0
  Valid certificates: 0
  Certificate date out of range: 0
  Total number of invalid certificates: 0
  Approved with warning (no crl check): 0
  Number of times polling CRL: 0
  Failed to get CRL: 0
  Not authorized (e.g. denied by ACL): 0
  Root certificates not self-signed: 0
  Verify requests failed (e.g. CRL operation failed): 0
  Number of times polling OCSP: 0
  OCSP invalid response date: 0
  Unknown failure: 0
  Empty certificate chain: 0
  No memory to process requests: 0
  DER encoded certificates missing: 0
  Bad DER certificate length: 0
  Failed to get key from certificate: 0
  Issuer CA not in trusted CA pool: 0
  Issuer CA certificates not valid yet: 0
  Expired issuer CA certificates: 0
  Peer certificates not valid yet: 0
  Expired peer certificates: 0
  Peer certificates revoked: 0
  Auth failures logged     : 0
  Auth failures allowed    : 0
  Revoked certs allowed    : 0
  Internal buffer overflow: 0
Peer certificate cache size: 0 (entries), aging timeout: 15 (minutes)
Peer certificate cache statistics:
  In use: 0 (entries)
  Cache hit: 0
  Cache miss: 0
  Cache allocated: 0
  Cache freed: 0
  Cache entries expired: 0
  Cache error: 0
  Cache full (wrapped around): 0
  No memory for caching: 0
Certificate Expiration Warning statistics:
  Proxy service certificates expiring: 0
  CA certificates expiring: 0
  CA pool certificates expiring: 0
  Proxy service certificates expiring SNMP traps sent: 0
Certificate headers statistics:
  Certificate headers formed: 0
  Errors in forming headers: 0
  Prefix error: 0
Key Certificate Table Current Usage (cannot be cleared):
  Total number of entries in table: 8192
```

```
  Entries in use: 5
  Free entries: 8187
  Complete service entries: 2
  Incomplete new/renew service entries: 0
  Retiring service entries: 0
  Obsolete service entries: 0
  Complete intermediate CA cert: 2
  Complete root CA cert: 1
  Obsolete intermediate CA cert: 0
  Obsolete root CA cert: 0
PKI Accumulative Counters (cannot be cleared):
  Proxy service trustpoint added: 2
  Proxy service trustpoint deleted: 0
  Proxy service trustpoint modified: 0
  Keypair added: 2
  Keypair deleted: 0
  Wrong key type: 0
  Service certificate added: 2
  Service certificate deleted: 0
  Service certificate rolled over: 0
  Service certificate completed: 2
  Intermediate CA certificate added: 2
  Intermediate CA certificate deleted: 0
  Root CA certificate added: 1
  Root CA certificate deleted: 0
  Certificate overwritten: 0
  No free table entries: 0
  Rollover failed: 0
Certificate History Statistics (cannot be cleared):
  History records written: 0
  History records deleted: 0
  History records malloc: 0
  History records free: 0
  History records errors: 0
  History records currently kept in memory: 0
  History records have been cleared: 0 times
PKI IPC Counters for normal priority messages:
  Request buffer sent: 0
  Request buffer received: 0
  Request duplicated: 0
  Request send failed: 0
  Response buffer sent: 0
  Response buffer received: 0
  Response timeout: 0
  Response failed: 0
  Response with error reported by SSL Processor: 0
  Response with no request: 0
  Response duplicated: 0
  Message type error: 0
  Message length error: 0
PKI IPC Counters for high priority messages:
  Request buffer sent: 1455695939
  Request buffer received: 0
  Request duplicated: 0
  Request send failed: 0
  Response buffer sent: 0
  Response buffer received: 1455695938
  Response timeout: 0
  Response failed: 0
  Response with error reported by SSL Processor: 0
  Response with no request: 0
  Response duplicated: 0
  Message type error: 0
  Message length error: 0
```

```
PKI Memory Usage Counters:
  Malloc count: 2911392424
  Free count: 2911392363
  Malloc failed: 0
  High Priority IPC:
  Ipc alloc count: 2911391878
  Ipc free count: 72120518
  Ipc alloc failed: 0
  Normal Priority IPC:
  Ipc alloc count: 0
  Ipc free count: 0
  Ipc alloc failed: 0
Ephemeral Key Generation Statistics:
  512 bit ephemeral keys : 14
  1024 bit ephemeral keys: 14
ssl-proxy#
```

This example shows how to display FDU statistics:

```
ssl-proxy# show ssl-proxy stats fdu
FDU Statistics:
    IP Reass in progress: 0             Frag Svc full Drops : 0
    IPFlow create Drops : 0             TTL expired Drops   : 0
    IP Frag Drops       : 0             Frag reass complete : 0
    Frag nodes freed    : 0             IP Version Drops    : 0
    IP Addr Discards    : 0             Serv_Id Drops       : 4266052
    Conn Id Drops       : 0             Bound Conn Drops    : 0
    Vlan Id Drops       : 0             TCP HW Checksum     : 4179625097
    TCP SW Checksum     : 0             TCP Checksum Drops  : 0
    Hash Full Drops     : 0             Hash Alloc Fails    : 0
    Flow Creates        : 28830314      Flow Deletes        : 28830314
    Conn Id allocs      : 14415157      Conn Id deallocs    : 14415157
    Tagged Pkts Drops   : 0             Non-Tagg Pkts Drops : 0
    Add ipcs            : 3             Delete ipcs         : 0
    Disable ipcs        : 0             Enable ipcs         : 0
    Unsolicited ipcs    : 0             Duplicate Add ipcs  : 0
    IOS Broadcast Pkts  : 5331232       IOS Unicast Pkts    : 3937
    IOS Multicast Pkts  : 0             IOS Total Pkts      : 5335169
    IOS Congest Drops   : 0             SYN Discards        : 7400
    UDP datagrams Rcvd  : 0             UDP datagrams Sent  : 0
    UDP HW Checksum     : 0             UDP SW Checksum     : 0
    UDP Flow Misses     : 0             UDP Length Errors   : 0
    TCP 5-tuple reuse   : 0             FDU Reset Drops     : 0


FDU Debug Counters:
    Inv. Conn Drops     : 0             Inv. Conn Pkt Drops : 0
    Inv. UDP Pkt Drops  : 0             Inv. TCP opcodes    : 0
    UDP Broadcast Drops : 0
```

This example shows how to display the HTTP header insertion statistics:

```
ssl-proxy# show ssl-proxy stats hdr
Header Insert Statistics:
    Session Headers Inserted : 0          Custom Headers Inserted : 0
    Session Id's Inserted    : 10149105   Client Cert. Inserted   : 0
    Client IP/Port Inserted  : 0          PEM Cert. Inserted      : 0
    Aliased Hdrs Inserted    : 0          Request boundry found   : 10149105
    Content Length Headers   : 0          Chunked Headers         : 0
    Content Length Splt Bufs : 0          Content Length Read Errs: 0
    Buffers allocated        : 0          Buffers Scanned         : 16031859
    Insertion Points Found   : 10149105   Hdrs Spanning Records   : 5882754
    End of Header Found      : 10149105   Buffers Accumulated     : 16031859
    Multi-buffer IP Port     : 0          Multi-buffer Session Id : 0
```

```
    Multi-buffer Session Hdr : 0          Multi-buffer Custom Hdr : 0
    HTTP Struct Allocs       : 14415156   HTTP Struct Frees       : 14415156
    No End of Hdr Detected   : 0          Payload no HTTP header  : 0
    Desc Alloc Failed        : 0          Buffer Alloc Failed     : 0
    Client Cert Errors       : 10149105   Malloc failed           : 0
    Service Errors           : 0          Conn Entry Invalid      : 0
    Scan Internal Error      : 0          Database Not Initialized: 0
    Unsupported headers      : 0          Client Cert. Insrt Basic: 0
    Missing Subject Name Errs: 0          Chunk Parse Errors      : 0
    Http headers removed     : 0          Http header removal errs: 0
```

This example shows how to display context statistics:

```
ssl-proxy# show ssl-proxy stats context
Context name : Def
TCP Context Statistics
======================
Current conns ACTIVE              : 0
Num conns DROPPED (hit max limit) : 0
Maximum conns ESTABLISHED         : 0

Context name : Default
TCP Context Statistics
======================
Current conns ACTIVE              : 0
Num conns DROPPED (hit max limit) : 0
Maximum conns ESTABLISHED         : 1150
```

This example shows how to display the URL rewrite statistics:

```
ssl-proxy# show ssl-proxy stats url
URL Rewrite Statistics:
    Rewrites Succeeded   : 0          Rewrites Failed      : 0
    Rsp Scan Incomplete  : 0          URL Scan Incomplete  : 0
    Invalid Conn Entry   : 0          URL Mismatch         : 0
    URL Object Error     : 0          Dbase not initialized: 0
    3xx URL Not Rewritten: 0          Scan Internal Error  : 0
    Scan Dbase not Init. : 0          Slash Delim not found: 0
```

This example shows how to display content statistics:

```
ssl-proxy# show ssl-proxy stats content
Scan object statistics in CPU: SSL1
    Objects in use       : 0
    Obj alloc failures   : 0
    Max obj in use       : 73
```

show ssl-proxy status

To display information about the SSL Services Module proxy status, use the **show ssl-proxy status** command.

show ssl-proxy status [**fdu** | **ssl** | **tcp**]

Syntax Description

fdu	(Optional) Displays the FDU status.
ssl	(Optional) Displays the SSL status.
tcp	(Optional) Displays the TCP status.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 1.2(1)	The output of the **show ssl-proxy status** command was changed to include statistics that are displayed at a 5-second, 1-minute, and 5-minute traffic rate for CPU utilization.
SSL Services Module Release 3.1(1)	This command was changed to add the following keywords: • **fdu** • **ssl** • **tcp**

Examples

This example shows how to display the status of the SSL Services Module:

```
ssl-proxy# show ssl-proxy status
FDU cpu is alive!
FDU cpu utilization:
    % process util   : 0             % interrupt util : 0

    proc cycles : 0x2DB3980C         int cycles  : 0x2ADACD71
    total cycles: 0x4E75127FCEA4
    % process util (5 sec)   : 0             % interrupt util (5 sec) : 0
    % process util (1 min)  : 0             % interrupt util (1 min): 0
    % process util (5 min)  : 0             % interrupt util (5 min) : 0
```

```
TCP cpu is alive!
TCP cpu utilization:
    % process util   : 0             % interrupt util : 0

    proc cycles : 0x2E42C686         int cycles  : 0x47F7C36A91
    total cycles: 0x4E799DB3F5F8
    % process util (5 sec)   : 0             % interrupt util (5 sec) : 0
    % process util (1 min)  : 0             % interrupt util (1 min): 0
    % process util (5 min)  : 0             % interrupt util (5 min) : 0

SSL cpu is alive!
SSL cpu utilization:
    % process util   : 0             % interrupt util : 0

    proc cycles : 0x9E396A4          int cycles  : 0xDB85C98B
    total cycles: 0x4E798224EDC1
    % process util (5 sec)   : 0             % interrupt util (5 sec) : 0
    % process util (1 min)  : 0             % interrupt util (1 min): 0
    % process util (5 min)  : 0             % interrupt util (5 min) : 0
```

This example shows how to display the status of the TCP CPU on the SSL Services Module:

```
ssl-proxy# show ssl-proxy status tcp
TCP cpu is alive!
TCP cpu utilization:
    % process util   : 0             % interrupt util : 0

    proc cycles : 0x2E45DAEE         int cycles  : 0x47FC7C2AC5
    total cycles: 0x4E7EC4499DC8
    % process util (5 sec)   : 0             % interrupt util (5 sec) : 0
    % process util (1 min)  : 0             % interrupt util (1 min): 0
    % process util (5 min)  : 0             % interrupt util (5 min) : 0
```

show ssl-proxy version

To display the current image version, use the **show ssl-proxy version** command.

show ssl-proxy version

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display the image version that is currently running on the SSL Services Module:

```
ssl-proxy# show ssl-proxy version
Cisco IOS Software, SVCSSL Software (SVCSSL-K9Y9-M)
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Mon 09-Jan-06 16:54 by integ

ROM: System Bootstrap, Version 12.2(11)YS1 RELEASE SOFTWARE

ssl-proxy uptime is 1 day, 15 hours, 57 minutes
System returned to ROM by power-on
System image file is "tftp://10.1.1.1/unknown"
AP Version 3.1(1)

ssl-proxy#
```

show ssl-proxy vlan

To display VLAN information, use the **show ssl-proxy vlan** command.

show ssl-proxy vlan [*vlan-id*][**debug**][**module** *module*]

Syntax Description

vlan-id	(Optional) VLAN ID. Displays information for a specific VLAN; valid values are from 1 to 1005.
debug	(Optional) Displays debug information.
module *module*	(Optional) Displays statistics for the specified module; module type includes the following: • **all**—all CPUs • **fdu**—FDU CPU • **ssl1**—SSL1 CPU • **tcp1**—TCP1 CPU

Defaults

This command has no default settings.

Command Modes

EXEC

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	This command was changed to add the **module** *module* keyword.

Examples

This example shows how to display all the VLANs that are configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy vlan
VLAN index 2:
   Associated with interface SSL-Proxy0.2 (UP)
   IP addr 207.10.0.16 NetMask 255.0.0.0
VLAN index 3:
   Associated with interface SSL-Proxy0.3 (UP)
   IP addr 208.10.0.16 NetMask 255.0.0.0
VLAN index 4:
   Associated with interface SSL-Proxy0.4 (UP)
   IP addr 209.10.0.16 NetMask 255.0.0.0
ssl-proxy#
```

Related Commands

interface ssl-proxy

snmp-server enable

To configure the SNMP traps and informs, use the **snmp-server enable** command. Use the **no** form of this command to disable SNMP traps and informs.

snmp-server enable {**informs** | **traps** {**ipsec** | **isakmp** | **snmp** | {**ssl-proxy** [**cert-expiring**] [**oper-status**]}}}

no snmp-server enable {**informs** | **traps** {**ipsec** | **isakmp** | **snmp** | {**ssl-proxy** [**cert-expiring**] [**oper-status**]}}}

Syntax Description

informs	Enables SNMP informs.
traps	Enables SNMP traps.
ipsec	Enables IPsec traps.
isakmp	Enables ISAKMP traps.
snmp	Enables SNMP traps.
ssl-proxy	Enables SNMP SSL proxy notification traps.
cert-expiring	(Optional) Enables SSL proxy certificate-expiring notification traps.
oper-status	(Optional) Enables SSL proxy operation-status notification traps.

Defaults

This command has no default setting.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Examples

This example shows how to enable SNMP informs:

```
ssl-proxy (config)# snmp-server enable informs
ssl-proxy (config)#
```

This example shows how to enable SSL-proxy traps:

```
ssl-proxy (config)# snmp-server enable traps ssl-proxy
ssl-proxy (config)#
```

This example shows how to enable SSL-proxy notification traps:

```
ssl-proxy (config)# snmp-server enable traps ssl-proxy cert-expiring oper-status
ssl-proxy (config)#
```

ssl pre-remove-http-hdr

To remove existing headers prior to inserting a new header, use the **ssl pre-remove-http-hdr** command. Use the **no** form of this command to ignore headers before insertion.

ssl pre-remove-http-hdr

no ssl pre-remove-http-hdr

Defaults

The default behavior for this command is to ignore the existing headers before inserting a new header.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 3.1(5)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

This command requests that the SSLM search HTTP messages for all http headers that the SSLM can insert except for custom headers. If any headers are found, they are removed. The command does not search for header prefixes or aliases. This command might impact SSLM performance based on the number of headers present.

Examples

This example shows how to remove existing headers:

```
ssl-proxy (config)# policy http-header example
ssl-proxy (config)# pre-remove-http-hdr
!
```

ssl-proxy context

To enter the SSL context submode and define the virtual SSL context, use the **ssl-proxy context** command. Use the **no** form of this command to remove any commands that you have entered in the SSL context subcommand mode from the configuration.

ssl-proxy context [*name*]

no ssl-proxy context *name*

Syntax Description

name	Name of the context.

Defaults

The default context name is "Default."

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The *name* argument is case sensitive.

After you enter the **ssl-proxy context** command, the prompt changes to the following:

```
ssl-proxy(config-context)#
```

After you enter the context submode, you can use the context submode commands listed in Table 2-12 to configure the context services.

Table 2-12 Context Submode Commands

Command	Purpose and Guidelines	Defaults
default	Set a command to its defaults	
description *description*	(Optional) Allows you to enter a short description for this context.	
exit	Exit from context configuration mode.	
maxconns *connections*	(Optional) Configures the maximum number of connections for this context. Valid values are from 1 to 65536.	65536
natpool *name start_ip_addr end_ip_addr* **netmask** *netmask*	Configures the NAT pool settings. See the "natpool" section on page 2-35.	
policy health-probe tcp *policy-name*	Configures the TCP health probe policy. See the "policy health-probe tcp" section on page 2-36.	

Table 2-12 Context Submode Commands (continued)

Command	Purpose and Guidelines	Defaults
policy http-header *policy-name*	Configures the HTTP header insertion policy. See the "policy http-header" section on page 2-39.	
policy ssl *policy-name*	Configures the SSL policy. See the"policy ssl" section on page 2-45.	
policy tcp *policy-name*	Configures the TCP policy. See the "policy tcp" section on page 2-51.	
policy url-rewrite *policy-name*	Configures the URL rewrite policy. See the "policy url-rewrite" section on page 2-55.	
pool ca *name*	Configures a pool of resources. See the "pool ca" section on page 2-57.	
service *service_name*	Enters SSL proxy service subcommand mode and lets you configure the SSL client or server proxy service. See the "service" section on page 2-58 for information about SSL proxy services.	
vrf-name *name*	Configures the VRF associated with this context.	

Examples

This example shows how to configure the context "hubble":

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy context hubble
ssl-proxy(config-context)# vrf-name hubble
ssl-proxy(config-context)# service hubble
ssl-proxy(config-ctx-ssl-proxy)# virtual ipaddr 3.100.100.108 protocol tcp port 443
ssl-proxy(config-ctx-ssl-proxy)# server ipaddr 5.100.100.41 protocol tcp port 80
ssl-proxy(config-ctx-ssl-proxy)# certificate rsa general-purpose trustpoint shuttle
ssl-proxy(config-ctx-ssl-proxy)# nat client hubble
ssl-proxy(config-ctx-ssl-proxy)# inservice
ssl-proxy(config-ctx-ssl-proxy)# exit
ssl-proxy(config-context)# natpool hubble 5.100.100.20 5.100.100.27 netmask 255.255.255.0
ssl-proxy(config-context)# policy health-probe tcp probe1
ssl-proxy(config-ctx-tcp-probe)# port 80
ssl-proxy(config-ctx-tcp-probe)# exit
ssl-proxy(config-context)#
ssl-proxy(config-context)# description Example context
ssl-proxy(config-context)# end
ssl-proxy#
```

ssl-proxy crypto selftest

To initiate a cryptographic self-test, use the **ssl-proxy crypto selftest** command. Use the **no** form of this command to disable the testing.

ssl-proxy crypto selftest [**time-interval** *seconds*]

no ssl-proxy crypto selftest

Syntax Description

time-interval *seconds*	(Optional) Sets the time interval between test cases; valid values are from 1 to 8 seconds.

Defaults

3 seconds

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The **ssl-proxy crypto selftest** command enables a set of crypto algorithm tests to be run on the SSL processor in the background. Random number generation, hashing, encryption and decryption, and MAC generation are tested with a time interval between test cases.

This test is run only for troubleshooting purposes. Running this test will impact run-time performance.

To display the results of the self-test, enter the **show ssl-proxy stats crypto** command.

Examples

This example shows how to start a cryptographic self-test:

```
ssl-proxy (config)# ssl-proxy crypto selftest
ssl-proxy (config)#
```

ssl-proxy device-check

To check the health of the crypto device, use the **ssl-proxy device-check** command.

ssl-proxy device-check interval *milliseconds* **reset-limit** *number*

Syntax Description

interval *milliseconds*	Device check interval in milliseconds. The range is from 10 to 60000. 0 = device check disabled.
reset-limit *number*	Number of consecutive resets before rebooting. The range is from 0 to 60. 0 = unlimited.

Defaults

The device check is disabled.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 3.1(5)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

This command is normally disabled (device check interval is 0). If the command is enabled, the SSLM checks the crypto device at every interval for proper operation. If there are outstanding requests older than the request interval, the crypto device is reset to return to operational status. A reset limit can also be configured. If the reset limit is set to default (zero), there is no limit. If the reset limit is non zero, the SSLM reboots if the device is reset for more than the reset-limit number of consecutive poll intervals.

Examples

This example shows how to set the device-check interval to 20 milliseconds, and reset-limit to 0:

```
ssl-proxy (config)# ssl-proxy device-check interval 20 reset-limit 0
```

This example shows how to check the number of resets that have occurred using the **show ssl-proxy stats ssl** command. Note the 'device reset count' in the output.

```
ssl-proxy# show ssl-proxy stats ssl
SSL Queue Sizes:
    bcm_cmd_ctx_pool_size   : 64            bcm_asym_cmd_ctx_pool_sz: 9000
    bcm_info_pool_size      : 65538         buf_desc_free_q_size    : 94710
    cert_result_free_q_size : 11048         delete_conn_q_size      : 0
    event_q_size            : 0             free_conn_q_size        : 65536
    free_sess_q_size        : 262144        free_sess_active_tmr_qsz: 0
    global_pending_q_size   : 0             to_app_ctx_pool_size    : 512
    ste_asym_req_q_size     : 0             ste_free_req_ctx_pool_sz: 20480
    ste_sym_req_q_size      : 0             available ctx count     : 64
    ctx cleanup count       : 0             device reset count      : 0
```

ssl-proxy disable-eth-pad

To disable the padding of Ethernet payload to even length, use the **ssl-proxy disable-eth-pad** command.

ssl-proxy disable-eth-pad

Defaults

Ethernet payload padding is enabled by default.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 3.1(5)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

When enabled, this command instructs the SSLM not to pad odd-length Ethernet payloads by one byte.

ssl-proxy mac address

To configure a MAC address, use the **ssl-proxy mac address** command.

ssl-proxy mac address *mac-addr*

Syntax Description

mac-addr	MAC address; see the "Usage Guidelines" section for additional information.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

Enter the MAC address in this format: H.H.H.

Examples

This example shows how to configure a MAC address:

```
ssl-proxy (config)# ssl-proxy mac address 00e0.b0ff.f232
ssl-proxy (config)#
```

Related Commands

show ssl-proxy mac address

ssl-proxy pki

To configure and define the PKI implementation on the SSL Services Module, use the **ssl-proxy pki** command. Use the **no** form of this command to disable the logging and clear the memory.

ssl-proxy pki {{**authenticate** {**timeout** *seconds*}} | {**cache** {{**size** *entries*} | {**timeout** *minutes*}}} | {**certificate** {**check-expiring** {**interval** *hours*}}} | **history**}

no ssl-proxy pki {**authenticate** | **cache** | **certificate** | **history**}

Syntax Description

authenticate	Configures the certificate authentication and authorization.
timeout *seconds*	Specifies the timeout in seconds for each request; valid values are from 1 to 600 seconds.
cache	Configures the peer-certificate cache.
size *entries*	Specifies the maximum number of cache entries; valid values are from 0 to 5000 entries.
timeout *minutes*	Specifies the aging timeout value of entries; valid values are from 1 to 600 minutes.
certificate	Configures the check-expiring interval.
check-expiring interval *hours*	Specifies the check-expiring interval; valid values are from 0 to 720 hours.
history	Key and certificate history.

Defaults

The default settings are as follows:

- **timeout** *seconds*—**180** seconds
- **size** *entries*—**0** entries
- **timeout** *minutes*—**15** minutes
- **interval** *hours*—**0** hours, do not check

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 2.1(1)	This command was changed to add the following keywords: • **authenticate** • **cache** • **certificate**

Usage Guidelines

The **ssl-proxy pki history** command enables logging of certificate history records per-proxy service into memory and generates a syslog message per record. Each record tracks the addition or deletion of a key pair or certificate into the proxy services key and the certificate table.

When the index of the table changes, this command logs the following information:

- Key pair name
- Trustpoint label
- Service name
- Subject name
- Serial number of the certificate

Up to 512 records can be stored in the memory at one time.

Examples

This example shows how to specify the timeout in seconds for each request:

```
ssl-proxy (config)# ssl-proxy pki authenticate timeout 200
ssl-proxy (config)#
```

This example shows how to specify the cache size:

```
ssl-proxy (config)# ssl-proxy pki cache size 50
ssl-proxy (config)#
```

This example shows how to specify the aging timeout value of entries:

```
ssl-proxy (config)# ssl-proxy pki cache timeout 20
ssl-proxy (config)#
```

This example shows how to specify the check-expiring interval:

```
ssl-proxy (config)# ssl-proxy pki certificate check-expiring interval 100
ssl-proxy (config)#
```

This example shows how to enable PKI event-history:

```
ssl-proxy (config)# ssl-proxy pki history
ssl-proxy (config)#
```

Related Commands

show ssl-proxy stats

ssl-proxy crypto key unlock rsa

To unlock the key automatically after a reload, use the **ssl-proxy crypto key unlock rsa** command.

ssl-proxy crypto key unlock rsa *key-name* **passphrase** *passphrase*

Syntax Description

key-name	Name of the key.
passphrase	Pass phrase.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Examples

This example shows how to unlock the keys automatically after a reload:

```
ssl-proxy(config)# ssl-proxy crypto key unlock rsa pki1-72a.cisco.com passphrase cisco1234
ssl-proxy(config)#
```

ssl-proxy ip-frag-ttl

To adjust the IP fragment reassembly timer, use the **ssl-proxy ip-frag-ttl** command.

ssl-proxy ip-frag-ttl *time*

Syntax Description

time	(Optional) Adjust the IP fragment reassembly timer; valid values are from 3 to 120 seconds.

Defaults

time is 6 seconds.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 3.1(1)	Support for this command was introduced on the Catalyst 6500 series SSL Services Module.

Examples

This example shows how to configure the IP reassembly timeout to 60 seconds:

```
ssl-proxy(config)# ssl-proxy ip-frag-ttl 60
ssl-proxy(config)#
```

ssl-proxy ssl ratelimit

To prohibit new connections during overload conditions, use the **ssl-proxy ssl ratelimit** command. Use the **no** form of this command to allow new connections if memory is available.

ssl-proxy ssl ratelimit

no ssl-proxy ssl ratelimit

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to prohibit new connections during overload conditions:

```
ssl-proxy (config)# ssl-proxy ssl ratelimit
ssl-proxy (config)#
```

This example shows how to allow new connections during overload conditions if memory is available:

```
ssl-proxy (config)# no ssl-proxy ssl ratelimit
ssl-proxy (config)#
```

standby authentication

To configure an authentication string for HSRP, use the **standby authentication** command. Use the **no** form of this command to delete an authentication string.

standby [*group-number*] **authentication text** *string*

no standby [*group-number*] **authentication text** *string*

Syntax Description

group-number	(Optional) Group number on the interface to which this authentication string applies. Valid values are from 0 to 255 for HSRP version 1; valid values are from 0 to 4095 for HSRP version 2. See the "standby version" section on page 2-132 for information about changing the HSRP version.
text *string*	Specifies the authentication string, which can be up to eight characters.

Defaults

The defaults are as follows:

- *group-number* is **0**.
- *string* is **cisco**.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

HSRP ignores unauthenticated HSRP messages.

The authentication string is sent unencrypted in all HSRP messages. You must configure the same authentication string on all routers and access servers on a cable to ensure interoperation. Authentication mismatch prevents a device from learning the designated hot standby IP address and the hot standby timer values from the other routers that are configured with HSRP.

When you use group number 0, no group number is written to NVRAM, providing backward compatibility.

Examples

This example shows how to configure "word" as the authentication string to allow hot standby routers in group 1 to interoperate:

```
ssl-proxy (config-subif)# standby 1 authentication text word
ssl-proxy (config-subif)#
```

standby delay minimum reload

To configure a delay before the HSRP groups are initialized, use the **standby delay minimum reload** command. Use the **no** form of this command to disable the delay.

standby delay minimum [*min-delay*] **reload** [*reload-delay*]

no standby delay minimum [*min-delay*] **reload** [*reload-delay*]

Syntax Description

min-delay	(Optional) Minimum time (in seconds) to delay HSRP group initialization after an interface comes up; valid values are from 0 to 10000 seconds.
reload-delay	(Optional) Time (in seconds) to delay after the router has reloaded; valid values are from 0 to 10000 seconds.

Defaults

The defaults are as follows:

- *min-delay* is **1** second.
- *reload-delay* is **5** seconds.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

The *min-delay* applies to all subsequent interface events.

The *reload-delay* applies only to the first interface-up event after the router has reloaded.

If the active router fails or you remove it from the network, the standby router automatically becomes the new active router. If the former active router comes back online, you can control whether it takes over as the active router by using the **standby preempt** command.

However, in some cases, even if you do not use the **standby preempt** command, the former active router resumes the active role after it reloads and comes back online. Use the **standby delay minimum reload** command to set a delay for HSRP group initialization. This command allows time for the packets to get through before the router resumes the active role.

We recommend that you use the **standby delay minimum reload** command if the **standby timers** command is configured in milliseconds or if HSRP is configured on a VLAN interface of a switch.

In most configurations, the default values provide sufficient time for the packets to get through and configuring longer delay values is not necessary.

The delay is canceled if an HSRP packet is received on an interface.

Examples

This example shows how to set the minimum delay to 30 seconds and the delay after the first reload to 120 seconds:

```
ssl-proxy(config)# interface ssl-proxy 0.100
ssl-proxy (config-subif)# standby delay minimum 30 reload 120
ssl-proxy (config-subif)#
```

Related Commands

show standby delay
standby preempt
standby timers

standby ip

To activate HSRP, use the **standby ip** command. Use the **no** form of this command to disable HSRP.

standby [*group-number*] **ip** [*ip-address* [**secondary**]]

no standby [*group-number*] **ip** [*ip-address*]

Syntax Description

group-number	(Optional) Group number on the interface for which HSRP is being activated.
ip-address	(Optional) IP address of the hot standby router interface.
secondary	(Optional) Indicates the IP address is a secondary hot standby router interface.

Defaults

The defaults are as follows:

- *group-number* is 0.
- HSRP is disabled by default.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

The **standby ip** command allows you to configure primary and secondary HSRP addresses.

The **standby ip** command activates HSRP on the configured interface. If you specify an IP address, that address is used as the designated address for the hot standby group. If you do not specifiy an IP address, the designated address is learned through the standby function. So that HSRP can elect a designated router, at least one router on the cable must have been configured with, or have learned, the designated address. Configuring the designated address on the active router always overrides a designated address that is currently in use.

When you enable the **standby ip** command on an interface, the handling of proxy ARP requests is changed (unless proxy ARP was disabled). If the hot standby state of the interface is active, proxy ARP requests are answered using the MAC address of the hot standby group. If the interface is in a different state, proxy ARP responses are suppressed.

When you use group number 0, no group number is written to NVRAM, providing backward compatibility.

Examples

This example shows how to activate HSRP for group 1 on Ethernet interface 0. The IP address that is used by the hot standby group is learned using HSRP.

```
ssl-proxy (config-subif)# standby 1 ip
ssl-proxy (config-subif)#
```

This example shows how to indicate that the IP address is a secondary hot standby router interface:

```
ssl-proxy (config-subif)# standby ip 1.1.1.254
ssl-proxy (config-subif)# standby ip 1.2.2.254 secondary
ssl-proxy (config-subif)# standby ip 1.3.3.254 secondary
```

standby mac-address

To specify a virtual MAC address for HSRP, use the **standby mac-address** command. Use the **no** form of this command to revert to the standard virtual MAC address (0000.0C07.AC*xy)*.

standby [*group-number*] **mac-address** *mac-address*

no standby [*group-number*] **mac-address**

Syntax Description

group-number	(Optional) Group number on the interface for which HSRP is being activated. The default is 0.
mac-address	MAC address.

Defaults

If this command is not configured, and the **standby use-bia** command is not configured, the standard virtual MAC address is used: 0000.0C07.AC*xy*, where *xy* is the group number in hexadecimal. This address is specified in RFC 2281, *Cisco Hot Standby Router Protocol (HSRP)*.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

This command cannot be used on a Token Ring interface.

You can use HSRP to help end stations locate the first-hop gateway for IP routing. The end stations are configured with a default gateway. However, HSRP can provide first-hop redundancy for other protocols. Some protocols, such as Advanced Peer-to-Peer Networking (APPN), use the MAC address to identify the first hop for routing purposes. In this case, it is often necessary to be able to specify the virtual MAC address; the virtual IP address is unimportant for these protocols. Use the **standby mac-address** command to specify the virtual MAC address.

The specified MAC address is used as the virtual MAC address when the router is active.

This command is intended for certain APPN configurations. The parallel terms are shown in Table 2-13.

Table 2-13 Parallel Terms Between APPN and IP

APPN	IP
End node	Host
Network node	Router or gateway

In an APPN network, an end node is typically configured with the MAC address of the adjacent network node. Use the **standby mac-address** command in the routers to set the virtual MAC address to the value that is used in the end nodes.

Examples

This example shows how to configure HSRP group 1 with the virtual MAC address:

```
ssl-proxy (config-subif)# standby 1 mac-address 4000.1000.1060
ssl-proxy (config-subif)#
```

Related Commands

show standby
standby version

standby mac-refresh

To change the interval at which packets are sent to refresh the MAC cache when HSRP is running over FDDI, use the **standby mac-refresh** command. Use the **no** form of this command to restore the default value.

standby mac-refresh *seconds*

no standby mac-refresh

Syntax Description

seconds	Number of seconds in the interval at which a packet is sent to refresh the MAC cache; valid values are from 1 to 255 seconds.

Defaults

seconds is **10** seconds.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

This command applies to HSRP running over FDDI only. Packets are sent every 10 seconds to refresh the MAC cache on learning bridges or switches. By default, the MAC cache entries age out in 300 seconds (5 minutes).

All other routers participating in HSRP on the FDDI ring receive the refresh packets, although the packets are intended only for the learning bridge or switch. Use this command to change the interval. Set the interval to 0 if you want to prevent refresh packets (if you have FDDI but do not have a learning bridge or switch).

Examples

This example shows how to change the MAC-refresh interval to 100 seconds. In this example, a learning bridge needs to miss three packets before the entry ages out.

```
ssl-proxy (config-subif)# standby mac-refresh 100
ssl-proxy (config-subif)#
```

standby name

To configure the name of the standby group, use the **standby name** command. Use the **no** form of this command to disable the name.

standby name *group-name*

no standby name *group-name*

Syntax Description

group-name	Name of the standby group.

Defaults

HSRP is disabled.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

The *group-name* argument specifies the HSRP group.

Examples

This example shows how to specifiy the standby name as SanJoseHA:

```
ssl-proxy (config-subif)# standby name SanJoseHA
ssl-proxy (config-subif)#
```

Related Commands

ip mobile home-agent redundancy (refer to the *Cisco IOS Release 12.2 Command Reference)*

standby preempt

To configure HSRP preemption and preemption delay, use the **standby preempt** command. Use the **no** form of this command to restore the default values.

standby [*group-number*] **preempt** [**delay**{**minimum** *delay* | **reload** *delay* | **sync** *delay*}]

no standby [*group-number*] **preempt** [**delay**{**minimum** *delay* | **reload** *delay* | **sync** *delay*}]

Syntax Description

group-number	(Optional) Group number on the interface to which the other arguments in this command apply.
delay	(Optional) Required if either the **minimum**, **reload**, or **sync** keywords are specified.
minimum *delay*	(Optional) Specifies the minimum delay in *delay* seconds; valid values are from 0 to 3600 seconds (1 hour).
reload *delay*	(Optional) Specifies the preemption delay after a reload only.
sync *delay*	(Optional) Specifies the maximum synchronization period in *delay* seconds.

Defaults

The defaults are as follows:

- *group-number* is 0.
- *delay* is 0 seconds; the router preempts immediately. By default, the router that comes up later becomes the standby router.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

The *delay* argument causes the local router to postpone taking over the active role for *delay* (minimum) seconds since that router was last restarted.

When you use this command, the router is configured to preempt, which means that when the local router has a hot standby priority that is higher than the current active router, the local router should attempt to assume control as the active router. If you do not configure preemption, the local router assumes control as the active router only if it receives information indicating no router is in the active state (acting as the designated router).

When a router first comes up, it does not have a complete routing table. If you configure the router to preempt, it becomes the active router, but it cannot provide adequate routing services. You can configure a delay before the preempting router actually preempts the currently active router.

When you use group number 0, no group number is written to NVRAM, providing backward compatibility.

IP-redundancy clients can prevent preemption from taking place. The **standby preempt delay sync** *delay* command specifies a maximum number of seconds to allow IP-redundancy clients to prevent preemption. When this expires, preemption takes place regardless of the state of the IP-redundancy clients.

The **standby preempt delay reload** *delay* command allows preemption to occur only after a router reloads. This provides stabilization of the router at startup. After this initial delay at startup, the operation returns to the default behavior.

The **no standby preempt delay** command disables the preemption delay but preemption remains enabled. The **no standby preempt delay minimum** *delay* command disables the minimum delay but leaves any synchronization delay if it was configured.

Examples

This example shows how to configure the router to wait for 300 seconds (5 minutes) before attempting to become the active router:

```
ssl-proxy (config-subif)# standby preempt delay minimum 300
ssl-proxy (config-subif)#
```

standby priority

To configure the priority for HSRP, use the **standby priority** command. Use the **no** form of this command to restore the default values.

standby [*group-number*] **priority** *priority*

no standby [*group-number*] **priority** *priority*

Syntax Description

group-number	(Optional) Group number on the interface to which the other arguments in this command apply.
priority	Priority value that prioritizes a potential hot standby router; valid values are from 1 to 255, where 1 denotes the lowest priority and 255 denotes the highest priority.

Defaults

The defaults are as follows:

- *group-number* is 0.
- *priority* is 100.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

The router in the HSRP group with the highest priority value becomes the active router.

When you use group number 0, no group number is written to NVRAM, providing backward compatibility.

The assigned priority is used to help select the active and standby routers. Assuming that preemption is enabled, the router with the highest priority becomes the designated active router. In case of ties, the primary IP addresses are compared, and the higher IP address has priority.

The priority of the device can change dynamically if an interface is configured with the **standby track** command and another interface on the router goes down.

Examples

This example shows how to change the router priority:

```
ssl-proxy (config-subif)# standby priority 120
ssl-proxy (config-subif)#
```

Related Commands

standby track

standby redirects

To enable HSRP filtering of Internet Control Message Protocol (ICMP) redirect messages, use the **standby redirects** command. Use the **no** form of this command to disable the HSRP filtering of ICMP redirect messages.

standby redirects [**enable** | **disable**] [**timers** *advertisement holddown*] [**unknown**]

no standby redirects [**unknown**]

Syntax Description

enable	(Optional) Allows the filtering of ICMP redirect messages on interfaces that are configured with HSRP, where the next-hop IP address may be changed to an HSRP virtual IP address.
disable	(Optional) Disables the filtering of ICMP redirect messages on interfaces that are configured with HSRP.
timers	(Optional) Adjusts HSRP-router advertisement timers.
advertisement	(Optional) HSRP-router advertisement interval in seconds; valid values are from 10 to 180 seconds.
holddown	(Optional) HSRP-router holddown interval in seconds; valid values are from 61 to 3600.
unknown	(Optional) Allows sending of ICMP packets to be sent when the next-hop IP address that is contained in the packet is unknown in the HSRP table of real IP addresses and active virtual IP addresses.

Defaults

The defaults are as follows:

- HSRP filtering of ICMP redirect messages is enabled if you configure HSRP on an interface.
- *advertisement* is 60 seconds.
- *holddown* is 180 seconds.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

You can configure the **standby redirects** command globally or on a per-interface basis. When you first configure HSRP on an interface, the setting for that interface inherits the global value. If you explicitly disable the filtering of ICMP redirects on an interface, then the global command cannot reenable this functionality.

The **no standby redirects** command is the same as the **standby redirects disable** command. We do not recommend that you save the **no** form of this command to NVRAM. Because the command is enabled by default, we recommend that you use the **standby redirects disable** command to disable the functionality.

With the **standby redirects** command enabled, the real IP address of a router can be replaced with a virtual IP address in the next-hop address or gateway field of the redirect packet. HSRP looks up the next-hop IP address in its table of real IP addresses versus virtual IP addresses. If HSRP does not find a match, the HSRP router allows the redirect packet to go out unchanged. The host HSRP router is redirected to a router that is unknown, that is, a router with no active HSRP groups. You can specify the **no standby redirects unknown** command to stop these redirects from being sent.

Examples

This example shows how to allow HSRP to filter ICMP redirect messages:

```
ssl-proxy (config-subif)# standby redirects
ssl-proxy (config-subif)#
```

This example shows how to change the HSRP router advertisement interval to 90 seconds and the holddown timer to 270 seconds on interface Ethernet 0:

```
ssl-proxy (config-subif)# standby redirects timers 90 270
ssl-proxy (config-subif)#
```

Related Commands

show standby
show standby redirect

standby timers

To configure the time between hello packets and the time before other routers declare the active hot standby or standby router to be down, use the **standby timers** command. Use the **no** form of this command to return to the default settings.

standby [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*

no standby [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*

Syntax Description

group-number	(Optional) Group number on the interface to which the timers apply.
msec	(Optional) Specifies the interval in milliseconds.
hellotime	Hello interval (in seconds); see the "Usage Guidelines" section for valid values.
holdtime	Time (in seconds) before the active or standby router is declared to be down; see the "Usage Guidelines" section for valid values.

Defaults

The defaults are as follows:

- *group-number* is 0.
- *hellotime* is 3 seconds.
- *holdtime* is 10 seconds.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

The valid values for *hellotime* are as follows:

- If you did not enter the **msec** keyword, valid values are from 1 to 254 seconds.
- If you enter the **msec** keyword, valid values are from 15 to 999 milliseconds.

The valid values for *holdtime* are as follows:

- If you did not enter the **msec** keyword, valid values are from *x* to 255 seconds, where *x* is the *hellotime* and 50 milliseconds and is rounded up to the nearest 1 second.
- If you enter the **msec** keyword, valid values are from *y* to 3000 milliseconds, where *y* is greater than or equal to 3 times the *hellotime* and is not less than 50 milliseconds.

If you specify the **msec** keyword, the hello interval is in milliseconds. Millisecond timers allow for faster failover.

The **standby timers** command configures the time between standby hello packets and the time before other routers declare the active or standby router to be down. Routers or access servers on which timer values are not configured can learn timer values from the active or standby router. The timers configured on the active router always override any other timer settings. All routers in a Hot Standby group should use the same timer values. Normally, holdtime is greater than or equal to three times the value of hellotime. The range of values for holdtime force the holdtime to be greater than the hellotime. If the timer values are specified in milliseconds, the holdtime is required to be at least three times the hellotime value and not less than 50 milliseconds.

Some HSRP state flapping can occasionally occur if the holdtime is set to less than 250 milliseconds, and the processor is busy. It is recommended that holdtime values less than 250 milliseconds be used. Setting the **process-max-time** command to a suitable value may also help with flapping.

The value of the standby timer will not be learned through HSRP hellos if it is less than 1 second.

When group number 0 is used, no group number is written to NVRAM, providing backward compatibility.

Examples

This example sets, for group number 1 on Ethernet interface 0, the time between hello packets to 5 seconds, and the time after which a router is considered to be down to 15 seconds:

```
interface ethernet 0
 standby 1 ip
 standby 1 timers 5 15
```

This example sets, for the hot router interface that is located at 172.19.10.1 on Ethernet interface 0, the time between hello packets to 300 milliseconds, and the time after which a router is considered to be down to 900 milliseconds:

```
interface ethernet 0
 standby ip 172.19.10.1
 standby timers msec 300 msec 900
```

This example sets, for the hot router interface that is located at 172.18.10.1 on Ethernet interface 0, the time between hello packets to 15 milliseconds, and the time after which a router is considered to be down to 50 milliseconds. Note that the holdtime is three times larger than the hellotime because the minimum holdtime value in milliseconds is 50.

```
interface ethernet 0
 standby ip 172.18.10.1
 standby timers msec 15 msec 50
```

standby track

To configure HSRP to track an object and change the hot standby priority based on the state of the object, use the **standby track** command. Use the **no** form of this command to remove the tracking.

standby [*group-number*] **track** *object-number* [**decrement** *priority*]

no standby [*group-number*] **track** *object-number* [**decrement** *priority*]

Syntax Description

group-number	(Optional) Group number to which the tracking applies.
object-number	Object number in the range from 1 to 500 representing the object to be tracked.
decrement *priority*	(Optional) Specifies the amount by which the hot standby priority for the router is decremented (or incremented) when the tracked object goes down (or comes back up).

Defaults

The defaults are as follows:

- *group-number* is **0**.
- *priority* is **10**.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

This command ties the hot standby priority of the router to the availability of its tracked objects. Use the **track interface** or **track ip route** global configuration command to track an interface object or an IP route object. The HSRP client can register its interest in the tracking process by using the **standby track** command commands and take action when the object changes.

When a tracked object goes down, the priority decreases by 10. If an object is not tracked, its state changes do not affect the priority. For each object configured for hot standby, you can configure a separate list of objects to be tracked.

The optional *priority* argument specifies how much to decrement the hot standby priority when a tracked object goes down. When the tracked object comes back up, the priority is incremented by the same amount.

When multiple tracked objects are down, the decrements are cumulative, whether configured with *priority* values or not.

Use the **no standby** *group-number* **track** command to delete all tracking configuration for a group.

When you use group number 0, no group number is written to NVRAM, providing backward compatibility.

The **standby track** command syntax prior to Release 12.2(15)T is still supported. Using the older form will cause a tracked object to be created in the new tracking process. This tracking information can be displayed using the **show track** command.

Examples

This example shows how to track the IP routing capability of serial interface 1/0. HSRP on Ethernet interface 0/0 registers with the tracking process to be informed of any changes to the IP routing state of serial interface 1/0. If the IP state on Serial interface 1/0 goes down, the priority of the HSRP group is reduced by 10.

If both serial interfaces are operational, Router A becomes the HSRP active router because it has the higher priority.

However, if IP routing on serial interface 1/0 in Router A fails, the HSRP group priority is reduced and Router B takes over as the active router, thus maintaining a default virtual gateway service to hosts on the 10.1.0.0 subnet.

Router A Configuration

```
!
track 100 interface serial1/0 ip routing
!
interface Ethernet0/0
  ip address 10.1.0.21 255.255.0.0
  standby 1 ip 10.1.0.1
  standby 1 priority 105
  standby 1 track 100 decrement 10
```

Router B Configuration

```
!
track 100 interface serial1/0 ip routing
!
interface Ethernet0/0
  ip address 10.1.0.22 255.255.0.0
  standby 1 ip 10.1.0.1
  standby 1 priority 100
  standby 1 track 100 decrement 10
```

Related Commands

standby preempt
standby priority

standby use-bia

To configure HSRP to use the burned-in address of the interface as its virtual MAC address instead of the preassigned MAC address (on Ethernet and FDDI) or the functional address (on Token Ring), use the **standby use-bia** command. Use the **no** form of this command to restore the default virtual MAC address.

standby use-bia [**scope interface**]

no standby use-bia

Syntax Description

scope interface	(Optional) Specifies that this command is configured only for the subinterface on which it was entered, instead of the major interface.

Defaults

HSRP uses the preassigned MAC address on Ethernet and FDDI or the functional address on Token Ring.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

You can configure multiple standby groups on an interface when you enter the **standby use-bia** command. Hosts on the interface must have a default gateway configured. We recommend that you set the **no ip proxy-arp** command on the interface. We also recommend that you configure the **standby use-bia** command on a Token Ring interface if there are devices that reject ARP replies with source hardware addresses that are set to a functional address.

When HSRP runs on a multiple-ring, source-routed bridging environment and the HRSP routers reside on different rings, configuring the **standby use-bia** command can prevent confusion about the routing information field (RFI).

Without the **scope interface** keywords, the **standby use-bia** command applies to all subinterfaces on the major interface. You cannot enter the **standby use-bia** command both with and without the **scope interface** keywords at the same time.

Examples

This example shows how to map the virtual MAC address to the virtual IP address:

```
ssl-proxy (config-subif)# standby use-bia
ssl-proxy (config-subif)#
```

standby version

To change the version of the Hot Standby Router Protocol (HSRP), use the **standby version** command:

standby version {**1** | **2**}

Syntax Description

1	Specifies HSRP version 1.
2	Specifies HSRP version 2.

Defaults

The default HSRP version is 1.

Command Modes

Subinterface configuration submode

Command History

Release	Modification
SSL Services Module Release 2.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 3.1(1)	The command mode for this command was changed from Proxy-VLAN to Subinterface.

Usage Guidelines

HSRP version 2 addresses limitations of HSRP version 1 by providing an expanded group number range of 0 to 4095.

HSRP version 2 will not interoperate with HSRP version 1. An interface cannot operate both version 1 and version 2 because both versions are mutually exclusive. You cannot change from version 2 to version 1 if you have configured groups above 255. Using the **no standby version** command sets the HSRP version to the default version, version 1.

If an HSRP version is changed, each group will reinitialize because it now has a new virtual MAC address.

Examples

This example shows how to configure HSRP version 2:

```
ssl-proxy (config-subif)# standby version 2
ssl-proxy (config-subif)#
```