



# CHAPTER 4

## Configuring SSL Initiation

---

This chapter describes how to configure a context on the Cisco 4700 Series Application Control Engine (ACE) appliance as an SSL client for SSL initiation.

This chapter contains the following major sections:

- [SSL Initiation Overview](#)
- [ACE SSL Initiation Configuration Prerequisites](#)
- [SSL Initiation Configuration Quick Start](#)
- [Creating and Defining an SSL Parameter Map](#)
- [Creating and Defining an SSL Proxy Service](#)
- [Creating a Layer 7 Class Map for SSL Initiation](#)
- [Creating a Layer 7 Policy Map for SSL Initiation](#)
- [Creating a Layer 3 and Layer 4 Class Map for SSL Initiation](#)
- [Creating a Layer 3 and Layer 4 Policy Map for SSL Initiation](#)
- [Applying the Policy Map to the VLANs](#)
- [Example of an SSL Initiation Configuration](#)



### Note

To verify that the SSL connection from a server to the ACE was properly initiated, you can monitor the handshake counters in the **show stats crypto client** command output (see [Chapter 6, Displaying SSL Information and Statistics](#)). The handshake counters increment for successful connections. For example, the SSLv3 Full Handshakes counter indicates that the handshake completed successfully and the SSLv3 Resumed Handshakes counter indicates that the

handshake resumed successfully by using a session ID. When traffic is flowing, those numbers should increment. If there are failures, then the alerts sent and received counters should also increment.

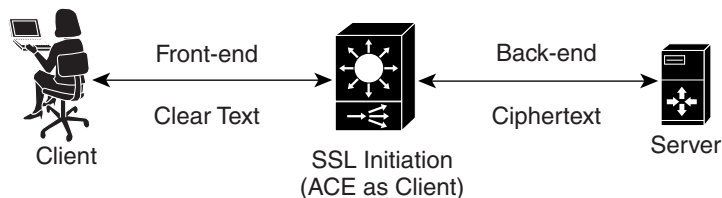
## SSL Initiation Overview

SSL initiation occurs when an ACE, acting as an SSL proxy client, initiates and maintains an SSL connection between itself and an SSL server. In this particular application, the ACE receives clear text from an HTTP client, and encrypts and transmits the data as ciphertext to the SSL server. On the reverse side, the ACE decrypts the ciphertext that it receives from the SSL server and sends the data to the client as clear text.

Figure 4-1 shows the following network connections in which the ACE initiates the SSL connection with the SSL server:

- Client to ACE—HTTP connection between the ACE and the client
- ACE to server—SSL connection between a server and the ACE acting as an SSL proxy client

**Figure 4-1** *SSL Initiation with an SSL Server*

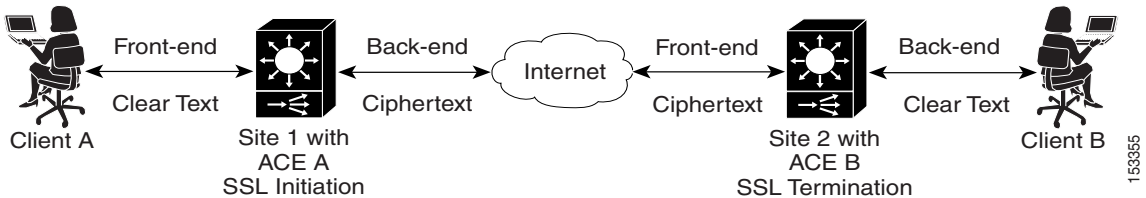


153356

SSL initiation allows you to send clear text between devices within a site for maximum speed, while sending ciphertext through the Internet between sites or to an SSL server for maximum security. For each SSL server or ACE (acting as an SSL proxy server) to which you want to establish an SSL connection from a clear text connection, you must configure an SSL initiation policy service on the ACE that maps to that SSL server or other ACE.

Figure 4-2 shows an SSL initiation flow with another ACE configured for SSL termination. In this case, ACE B acts as a virtual front-end SSL server.

**Figure 4-2** *SSL Initiation with a Second ACE Running SSL Termination*

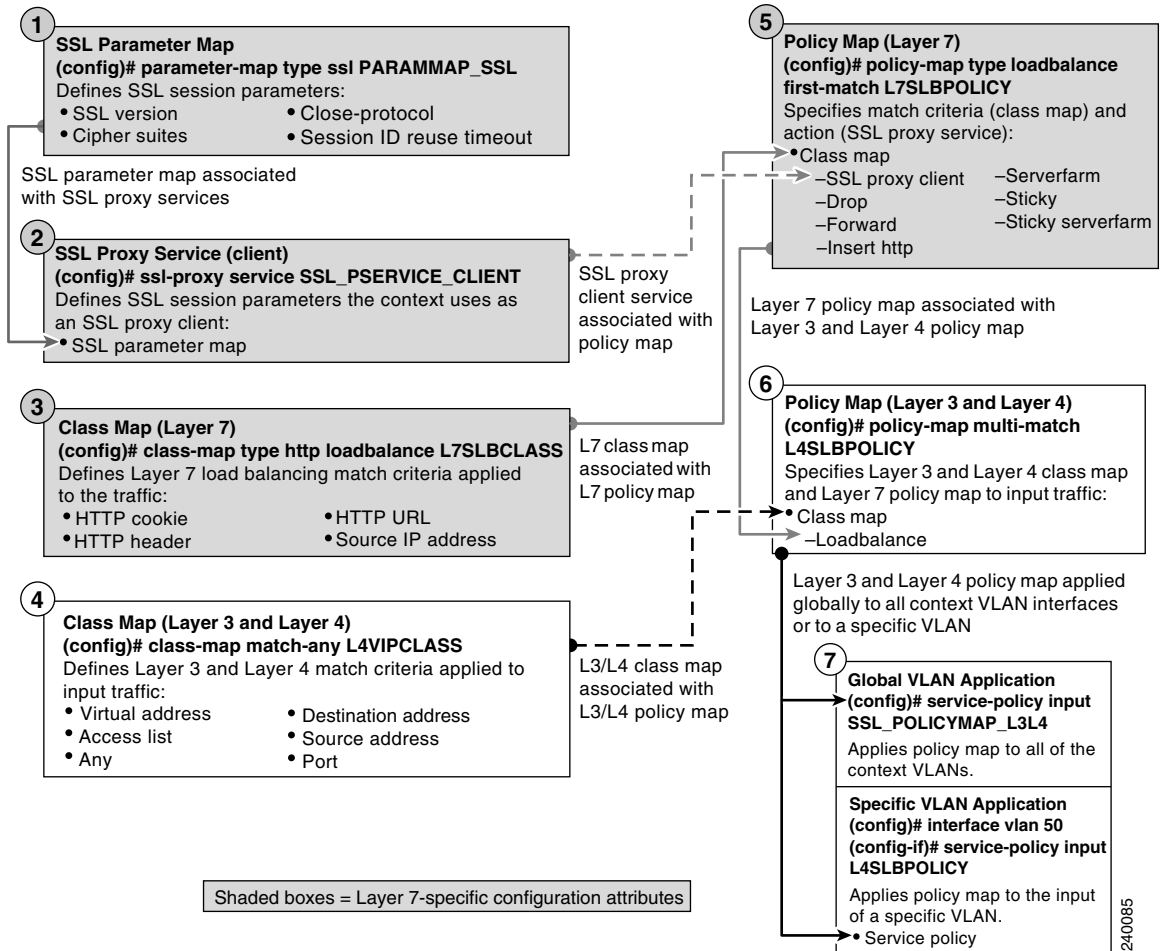


The ACE uses a combination of parameter maps, SSL proxy services, and class maps to build the policy maps that determine the flow of information among the client, the ACE, and the SSL server. For SSL initiation, you configure the ACE so that it is recognized as an SSL client by an SSL server. To accomplish this, you configure the following policy map types:

- **Layer 7 policy map**—This policy map contains an association with a Layer 7 class map and an SSL proxy client service. The class map acts as a traffic filter and looks for traffic that matches the server load-balancing (SLB) criteria that you specify. For SSL initiation, the match criteria is in the form of HTTP load-balancing attributes, such as an HTTP cookie or URL. The SSL proxy client service defines the SSL parameters that the ACE uses during the handshake and subsequent SSL session.
- **Layer 3 and Layer 4 policy map**—You associate the Layer 7 policy map with a Layer 3 and Layer 4 policy map. The ACE applies the Layer 3 and Layer 4 policy map to the context traffic first to determine if the traffic contains specific Layer 3 and Layer 4 match criteria, such as a particular destination, source, or virtual IP address. You specify the match criteria in the Layer 3 and Layer 4 class map that you create and associate with this policy map. When a match is found, the ACE applies the associated Layer 7 policy map to the traffic.

Figure 4-3 provides a basic overview of the process required to build and apply the two types of policy maps that the ACE uses for SSL initiation. The figure also shows how you associate the various components of the policy map configurations with each other.

Figure 4-3 Basic SSL Initiation Configuration Flow Diagram



# ACE SSL Initiation Configuration Prerequisites

Before configuring your ACE for SSL operation, you must first configure it for server load balancing (SLB). During the SLB configuration process, you create the following configuration objects:

- Layer 7 class map
- Layer 3 and Layer 4 class map
- Layer 7 policy map
- Layer 3 and Layer 4 policy map

After configuring SLB, modify the existing SLB class maps and policy maps with the SSL configuration requirements described in this guide for SSL initiation.

To configure your ACE for SLB, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.

## SSL Initiation Configuration Quick Start

[Figure 4-1](#) provides a quick overview of the steps required to configure the ACE for SSL initiation. Each step includes the CLI command or a reference to the procedure required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the sections following [Table 4-1](#).

**Note**

---

The following quick start does not include the procedures for creating a parameter map as shown in [Figure 4-3](#). The ACE uses the default parameter map settings as described in [Table 4-2](#).

---

**Table 4-1** *SSL Initiation Configuration Quick Start***Task and Command Example**

1. If you are operating in multiple contexts, observe the CLI prompt to verify that you are operating in the desired context. If necessary, log directly in to, or change to, the correct context.

```
host1/Admin# changeto c1
host1/C1#
```

The rest of the examples in this table use the Admin context. For details on creating contexts, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

2. Enter configuration mode.

```
host1/Admin# config
host1/Admin(config)#
```

3. Create an SSL proxy client service to associate with the Layer 7 policy map. For the purposes of this Quick Start, you do not define any parameters of the proxy client service; associating this generic proxy client service with the policy map is all that is required to configure the ACE to perform as an SSL client.

```
host1/Admin(config)# ssl-proxy service SSL_PSERVICE_CLIENT
host1/Admin(config-ssl-proxy)# exit
```

4. Create a Layer 7 class map and configure it with the required load-balancing match criteria.

```
host1/Admin(config)# class-map type http loadbalance L7SLBCLASS
host1/Admin(config-cmap-http-lb)# match url XYZ.ORG
host1/Admin(config-cmap-http-lb)# exit
host1/Admin(config)#
```

5. Create a Layer 3 and Layer 4 class map and configure it with the required input traffic match criteria.

```
host1/Admin(config)# class-map match-any L4VIPCLASS
host1/Admin(config-cmap)# match virtual-address 192.168.12.2
255.255.255.0
host1/Admin(config-cmap)# exit
host1/Admin(config)#
```

**Table 4-1** *SSL Initiation Configuration Quick Start (continued)*

---

**Task and Command Example**

---

6. Create a Layer 7 policy map and associate the Layer 7 class map created in Step 4 with it.

```
host1/Admin(config)# policy-map type loadbalance first-match  
L7SLBPOLICY  
host1/Admin(config-pmap-lb)# class L7SLBCLASS  
host1/Admin(config-pmap-lb-c)#
```

---

7. Associate the SSL proxy client service created in Step 3 with the Layer 7 policy map.

```
host1/Admin(config-pmap-lb-c)# ssl-proxy client  
SSL_PSERVICE_CLIENT  
host1/Admin(config-pmap-lb-c)# exit  
host1/Admin(config-pmap-lb)# exit  
host1/Admin(config)#
```

---

8. Create a Layer 3 and Layer 4 policy map and associate the Layer 3 and Layer 4 class map created in Step 5 with it.

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY  
host1/Admin(config-pmap)# class CLASSMAP_L3  
host1/Admin(config-pmap-c)#
```

---

9. Associate the load-balancing Layer 7 policy map created in Step 6 with the Layer 3 and Layer 4 policy map.

```
host1/Admin(config-pmap-c)# loadbalance L7SLBPOLICY  
host1/Admin(config-pmap-c)# exit  
host1/Admin(config-pmap)# exit  
host1/Admin(config)#
```

---

**Table 4-1** *SSL Initiation Configuration Quick Start (continued)*

---

**Task and Command Example**

---

10. Apply the Layer 3 and Layer 4 policy map to the input traffic of the desired interface as follows:

Apply the policy map globally to all VLANs in the context.

```
host1/Admin(config)# service-policy input L4SLBPOLICY
```

Apply the policy map to a specific VLAN within the context.

```
host1/Admin(config)# interface vlan 50  
host1/Admin(config-if)# service-policy input L4SLBPOLICY
```

- 
11. Display the running configuration to verify that the information that you just added is configured properly.

```
host1/Admin(config-if)# do show running-config
```

- 
12. (Optional) Save the configuration changes to flash memory by copying the running configuration to the startup configuration.

```
host1/Admin(config-if)# do copy running-config startup-config
```

---



# Creating and Defining an SSL Parameter Map

An SSL parameter map defines the SSL session parameters that the ACE applies to an SSL proxy service. Creating an SSL parameter map allows you to apply the same SSL session parameters to different proxy services. [Table 4-2](#) describes each SSL session parameter with its default value.

**Table 4-2** *SSL Session Parameters of an SSL Parameter Map*

SSL Session Parameter	Description	Default Value/Behavior
Cipher suites	Defines the cipher suites that the ACE supports during the SSL handshake (see <a href="#">Table 4-3</a> for a list of available cipher suites the ACE supports)	The ACE supports all of the available cipher suites
Authentication-failure ignore	Enables the ACE to ignore expired or invalid server certificates and to continue setting up the back-end connection in an SSL initiation configuration.	The ACE terminates the SSL handshake when a certificate failure is encountered.
Close-protocol	Defines how the ACE executes close-notify messages	<b>none</b> —The ACE sends a close notify alert message to the client/server when closing a session but has no expectation of receiving one back from the client/server
Rehandshake	Enables rehandshake, allowing the ACE to send an SSL HelloRequest message to its peer to restart SSL handshake negotiation	Disabled

**Table 4-2** *SSL Session Parameters of an SSL Parameter Map (continued)*

SSL Session Parameter	Description	Default Value/Behavior
Version	Defines the SSL and TLS versions that the ACE supports during the SSL handshake	The ACE supports versions SSL3 and TLS1
Session cache timeout	Defines the amount of time that the SSL session ID remains valid before the ACE requires a new SSL handshake to establish a new SSL session	Disabled



**Note**

If you want an SSL proxy service to use the default values for the SSL session parameters, you do not need to create an SSL parameter map or associate one with the proxy service. When you do not associate a parameter map with the SSL proxy service, the ACE automatically applies the default values for the session parameters listed in [Table 4-2](#) to the proxy service.

The parameter map SSL configuration mode includes the **queue-delay timeout** command. The queue delay applies only to encrypted data that the ACE sends to the client. For this reason, this timer has no effect on SSL initiation connections handled by the ACE.

You can create an SSL parameter map by using the **parameter-map type ssl** command in configuration mode.

The syntax of this command is as follows:

```
parameter-map type ssl parammap_name
```

The *parammap\_name* argument is the name of the SSL parameter map. Enter an unquoted alphanumeric string with no spaces and a maximum of 64 characters.

For example, to create the SSL parameter map PARAMMAP\_SSL, enter:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
```

After you create an SSL proxy parameter map, the CLI enters parameter map SSL configuration mode.

```
host1/Admin(config-parammap-ssl)#
```

If you exit out of the parameter map SSL configuration mode without defining any of its SSL session parameters, the ACE configures the parameter map with the default values listed in [Table 4-2](#).

To delete an existing SSL parameter map, enter:

```
host1/Admin(config)# no parameter-map type ssl PARAMMAP_SSL
```

This section contains the following topics:

- [Defining a Description of the SSL Parameter Map](#)
- [Adding a Cipher Suite](#)
- [Ignoring Expired or Invalid Server Certificates](#)
- [Defining the Close-Protocol Behavior](#)
- [Enabling SSL Session Rehandshake](#)
- [Defining the SSL and TLS Versions](#)
- [Configuring the SSL Session Cache Timeout](#)

## Defining a Description of the SSL Parameter Map

You can provide a brief summary of the SSL parameter map by using the **description** command in SSL parameter map configuration mode. The syntax of this command is as follows:

```
description text
```

For the *text* argument, enter an unquoted text string with a maximum of 240 alphanumeric characters including spaces.

For example, to specify a description of an SSL parameter map, enter the following command:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL  
host1/Admin(config-parammap-conn)# description SSL parameter map
```

To remove the description from the SSL parameter map, enter:

```
host1/Admin(config-parammap-conn)# no description
```

## Adding a Cipher Suite

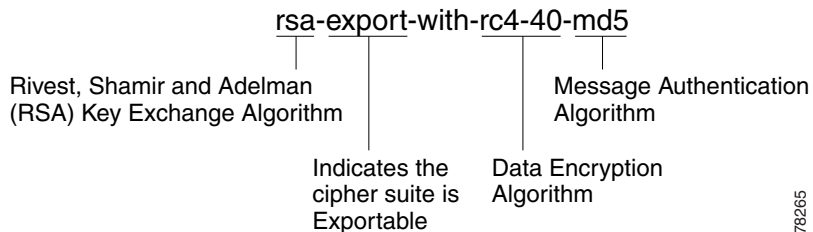
The SSL protocol supports a variety of different cryptographic algorithms, or ciphers, for use in operations such as the following:

- Authenticating the server and client to each other
- Transmitting certificates
- Establishing session keys

Clients and servers may support different cipher suites, or sets of ciphers, depending on various factors, such as the version of SSL that they support, company policies regarding acceptable encryption strength, and government restrictions on export of SSL-enabled software. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suite they will use to authenticate each other, transmit certificates, and establish session keys.

As shown in [Figure 4-4](#), a cipher suite consists of the following three algorithms: key exchange algorithm, data encryption algorithm, and message authentication (hash) algorithm.

**Figure 4-4** *Cipher Suite Algorithms*



78265

**Note**

Exportable cipher suites are those cipher suites that are not as strong as some of the other cipher suites (for example, 3DES or RC4 with 128-bit encryption) as defined by U.S. export restrictions on software products. Exportable cipher suites may be exported to most countries from the United States and provide the strongest encryption available for exportable products.

You can define each of the cipher suites that you want the ACE to support during a secure session by using the **cipher** command in `ssl parameter-map` configuration mode. The cipher suite that you choose depends on your environment and security requirements and must correlate to the certificates and keys that you have loaded on the ACE.

**Note**

By default, the ACE supports all of the cipher suites listed in [Table 4-3](#). The default setting works only when you do not configure the SSL parameter map with any specific ciphers. To return to using the all cipher suites setting, you must delete each specifically defined cipher from the parameter map by using the **no** form of the command.

The syntax of this command is as follows:

```
cipher cipher_name [priority cipher_priority]
```

The keywords and arguments are as follows:

- *cipher\_name*—Name of the cipher suite that you want the ACE to support. [Table 4-3](#) lists the cipher suites that the ACE supports. Enter one of the supported cipher suites from the table.
- **priority**—Assigns a priority level to the cipher suite. The priority level represents the preference ranking of the cipher suite, with 10 being the most preferred and 1 being the least preferred. By default, all configured cipher suites have a priority level of 1. When negotiating which cipher suite to use, the ACE selects from the client list based on the cipher suite configured with the highest priority level. A higher priority level will bias towards the specified cipher suite. For SSL termination applications, the ACE uses the priority level to match cipher suites in the client's ClientHello handshake

message. For SSL initiation applications, the priority level represents the order in which the ACE places the cipher suites in its ClientHello handshake message to the server.

- *cipher\_priority*—Priority level of the cipher suite. Enter a value of 1 to 10. The default priority value is 1.

For example, to add the cipher suite `rsa_with_aes_128_cbc_sha` and assign it a priority 2 level, enter:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# cipher rsa_with_aes_128_cbc_sha
priority 2
```

Repeat the **cipher** command for each cipher suite that you want to include in the SSL parameter map.

To delete a cipher suite from the SSL parameter map, enter:

```
host1/Admin(config-parammap-ssl)# no cipher rsa_with_aes_128_cbc_sha
```

[Table 4-3](#) lists the available cipher suites that the ACE supports and indicates which of the supported cipher suites are exportable from the ACE. The table also lists the authentication certificate and encryption key required by each cipher suite.

If you use the default setting in which the ACE supports all of the ciphers suites listed in [Table 4-3](#), the ACE sends the cipher suites to its peer in the same order as they appear in the table, starting with `RSA_WITH_RC4_128_MD5`.



#### Caution

Cipher suites with “export” in the title indicate that they are intended for use outside of the domestic United States and have encryption algorithms with limited key sizes.

**Table 4-3** *SSL Cipher Suites Supported by the ACE*

Cipher Suite	Exportable	Authentication Certificate Used	Key Exchange Algorithm Used
RSA_WITH_RC4_128_MD5	No	RSA certificate	RSA key exchange
RSA_WITH_RC4_128_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_DES_CBC_SHA	No	RSA certificate	RSA key exchange

Table 4-3 SSL Cipher Suites Supported by the ACE (continued)

Cipher Suite	Exportable	Authentication Certificate Used	Key Exchange Algorithm Used
RSA_WITH_3DES_EDE_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_EXPORT_WITH_RC4_40_MD5	Yes	RSA certificate	RSA key exchange
RSA_EXPORT_WITH_DES40_CBC_SHA	Yes	RSA certificate	RSA key exchange
RSA_EXPORT1024_WITH_RC4_56_MD5	Yes	RSA certificate	RSA key exchange
RSA_EXPORT1024_WITH_DES_CBC_SHA	Yes	RSA certificate	RSA key exchange
RSA_EXPORT1024_WITH_RC4_56_SHA	Yes	RSA certificate	RSA key exchange
RSA_WITH_AES_128_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_AES_256_CBC_SHA	No	RSA certificate	RSA key exchange

## Ignoring Expired or Invalid Server Certificates

You can enable the ACE to ignore expired or invalid server certificates and to continue setting up the back-end connection in an SSL initiation configuration by using the **authentication-failure ignore** command in parameter map SSL configuration mode. This command allows the ACE to ignore the following nonfatal errors with respect to server certificates:

- Certificate not yet valid
- Certificate has expired
- Unable to get issuer certificate
- Certificate revoked

The syntax of this command is as follows:

**authentication-failure ignore**

For example, to ignore expired or invalid server certificates, enter:

```
host1/Admin(config)# parameter-map type ssl SSL_PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# authentication-failure ignore
```

To return to the default setting of disabled, use the **no** form of the command:

```
host1/Admin(config-parammap-ssl)# no authentication-failure ignore
```

## Defining the Close-Protocol Behavior

You can configure how the ACE handles the sending of close-notify messages by using the **close-protocol** command in the `ssl parameter-map` configuration mode.

The syntax for this command is as follows:

```
close-protocol { disabled | none }
```

The keywords are as follows:

- **disabled**—Specifies that the ACE does not send a close notify alert message to the client/server when closing a session with no expectation of receiving one back from the client/server.
- **none**—Specifies that the ACE sends a close notify alert message to the client/server when closing a session but has no expectation of receiving one back from the client/server.

For example, to set `close-protocol` to `disabled`, enter:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL  
host1/Admin(config-parammap-ssl)# close-protocol disabled
```

To configure the **close-protocol** command to the default setting to send a close notify alert message to the client/server, enter:

```
host1/Admin(config-parammap-ssl)# no close-protocol
```

## Enabling SSL Session Rehandshake

The SSL session rehandshake enables the ACE to send the SSL HelloRequest message to a client to restart SSL handshake negotiation. The rehandshake is useful when you want to ensure security by reestablishing the SSL session.

By default, SSL rehandshake is disabled. To enable the SSL session rehandshake function during a session, use the **rehandshake enable** command in the parameter map SSL configuration mode.

The syntax of this command is:

```
rehandshake enable
```



For example, to enable the SSL rehandshake function, enter:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL  
host1/Admin(config-parammap-ssl) # rehandshake enable
```

To disable the rehandshake function, enter:

```
host1/Admin(config-parammap-ssl) # no rehandshake enable
```

To display the status of the rehandshake enable command, use the **show parameter-map** command.

## Defining the SSL and TLS Versions

You can specify the version of the security protocol that the ACE supports during the SSL handshake with its peer by using the **version** command in parameter map SSL configuration mode.

The syntax of this command is as follows:

```
version {all | ssl3 | tls1}
```

The keywords are as follows:

- **all**—(Default) The ACE supports both SSL Version 3.0 and TLS Version 1.0.
- **ssl3**—The ACE supports only SSL Version 3.0.
- **tls1**—The ACE supports only TLS Version 1.0.

For example, to specify SSL Version 3.0 for the parameter map, enter:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL  
host1/Admin(config-parammap-ssl) # version ssl3
```

To remove a security protocol version from the SSL proxy parameter map, enter:

```
host1/Admin(config-parammap-ssl) # no version tls1
```

## Configuring the SSL Session Cache Timeout

An SSL session ID is created every time that the client and the ACE perform a full SSL key exchange and establish a new master secret key. To quicken the SSL negotiation process between the client and the ACE, the SSL session ID reuse

feature allows the ACE to reuse the secret key information in the session cache. On subsequent connections with the client, the ACE reuses the key stored in the cache from the last negotiated session.

By default, SSL session ID reuse is disabled on the ACE. You can enable session ID reuse by setting a session cache timeout value for the total amount of time that the SSL session ID remains valid before the ACE requires a full SSL handshake to establish a new session.

You can set the session cache timeout by using the **session-cache timeout** command in parameter map SSL configuration mode. The syntax of this command is as follows:

**session-cache timeout** *seconds*

The *seconds* argument is the time in seconds that the ACE reuses the key stored in cache before removing the session IDs. Enter an integer from 0 to 72000 (20 hours). By default, session ID reuse is disabled. A value of 0 causes the ACE to remove the session IDs from the cache when the cache is full and to implement the least-recently used (LRU) timeout policy.

For example, to set the session cache timeout to 600 seconds, enter:

```
host1/Admin(config-parammap-ssl)# session-cache timeout 600
```

To disable the timer and allow the SSL full handshake to occur for each new connection with the ACE, enter:

```
host1/Admin(config-parammap-ssl)# no session-cache timeout
```

To clear the session cache information for the context, use the **clear crypto session-cache** command. The syntax of this command is as follows:

**clear crypto session-cache** [**all**]

The **all** optional keyword clears all session cache information for all contexts. This option is available in the Admin context only.

# Creating and Defining an SSL Proxy Service

The SSL proxy service defines the SSL parameter map that the ACE uses during the SSL handshake. For SSL initiation, you configure the ACE with an SSL proxy *client* service because the ACE acts as an SSL client.

**Note**

---

You do not need to import or associate keys and certificates in an SSL initiation configuration.

---

You can create an SSL proxy client service by using the **ssl-proxy service** command in configuration mode.

The syntax of this command is as follows:

```
ssl-proxy service pservice_name
```

The *pservice\_name* argument is the name of the SSL proxy client service. Enter an unquoted alphanumeric string with no spaces and a maximum of 64 characters.

For example, to create the SSL proxy client service `PSERVICE_CLIENT`, enter:

```
host1/Admin(config)# ssl-proxy service PSERVICE_CLIENT
```

After you create an SSL proxy client service, the CLI enters into SSL proxy configuration mode.

```
host1/Admin(config-ssl-proxy)#
```

To delete an existing SSL proxy client service, enter:

```
host1/Admin(config)# no ssl-proxy PSERVICE_CLIENT
```

This section contains the following topics:

- [Associating an SSL Parameter Map with the SSL Proxy Client Service](#)
- [Configuring an Authentication Group for Server Authentication](#)
- [Configuring Signature Verification on a CRL](#)

## Associating an SSL Parameter Map with the SSL Proxy Client Service

You can associate an SSL parameter map with the SSL proxy client service by using the **ssl advanced-options** command in SSL proxy configuration mode.

The syntax of this command is as follows:

```
ssl advanced-options parammap_name
```

The *parammap\_name* argument is the name of an existing SSL parameter map (see the “[Creating and Defining an SSL Parameter Map](#)” section). Enter an unquoted alphanumeric string with no spaces and a maximum of 64 characters.

For example, to associate the parameter map PARAMMAP\_SSL with the SSL proxy service, enter:

```
host1/Admin(config)# ssl-proxy service PSERVICE_CLIENT  
host1/Admin(config-ssl-proxy)# ssl advanced-options PARAMMAP_SSL
```

To remove the association of an SSL parameter map with the SSL proxy service, enter:

```
host1/Admin(config-ssl-proxy)# no ssl advanced-options PARAMMAP_SSL
```

## Configuring an Authentication Group for Server Authentication

By default, server authentication is always enabled in an SSL initiation configuration. The server must send a certificate to the ACE. The ACE authenticates the certificate by verifying that it is a server certificate and that it has not expired. However, the ACE does not check that the certificate has been signed by an approved CA. If the server certificate has expired, the ACE rejects the backend connection by sending a reset (RST) to the server. Otherwise, the ACE sets up the SSL connection with the server normally.

You can override this behavior by using the **authentication-failure ignore** command in parameter map SSL configuration mode. For details about this command, see the “[Ignoring Expired or Invalid Server Certificates](#)” section.

An authentication group consists of certificates that are trusted as certificate signers (see the “[Configuring a Group of Certificates for Authentication](#)” section in [Chapter 2, Managing Certificates and Keys](#)). When you assign an

authentication group to an SSL-proxy server in an SSL initiation configuration, the ACE checks the server certificate with the certificates in the group, which includes checking the issuer and the signature of the server certificate.

To use an authentication group for server authentication on this SSL-proxy service, use the **authgroup** command in SSL proxy configuration mode. The syntax of the **authgroup** command is as follows:

```
authgroup group_name
```

The *group\_name* argument is the name of an existing certificate authentication group. Enter an unquoted alphanumeric string with no spaces and a maximum of 64 characters.

**Note**

---

When you enable server authentication, a significant performance decrease of the ACE may occur.

---

For example, to specify the certificate authentication group AUTH-CERT1, enter:

```
host1/Admin(config-ssl-proxy) # authgroup AUTH-CERT1
```

To delete a certificate authentication group from the SSL proxy service, enter:

```
host1/Admin(config-ssl-proxy) # no authgroup AUTH-CERT1
```

## Configuring Signature Verification on a CRL

You can configure signature verification on a Certificate Revocation List (CRL) to determine that it is from a trusted certificate authority by using the **crypto crlparams** command in Exec command mode. The syntax of this command is as follows:

```
crypto crlparams crl_name cacert ca_cert_filename
```

The arguments are as follows:

- *crl\_name*—Name of an existing CRL.
- *ca\_cert\_filename*—Name of the CA certificate file used for signature verification.

For example, to configure signature verification on a CRL, enter:

```
host1/Admin(config)# crypto crlparams CRL1 cacert MYCERT.PEM
```

To remove signature verification from a CRL, enter:

```
host1/Admin(config)# no crypto crlparams CRL1
```

## Creating a Layer 7 Class Map for SSL Initiation

The Layer 7 class map that you associate with a policy map acts as a filter for traffic that matches the server load balancing (SLB) criteria that you specify. For SSL initiation, the match criteria is in the form of the following HTTP load-balancing attributes:

- Cookie
- HTTP header
- URL
- Source IP address

You can create a Layer 7 class map by using the **class-map type http loadbalance** command in configuration mode. For details on configuring a Layer 7 class map, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.

## Creating a Layer 7 Policy Map for SSL Initiation

A Layer 7 policy map enables server load balancing on the ACE. This policy map contains an association with a Layer 7 class map and an SSL proxy client service. To use a Layer 7 SLB policy map, you first create the policy map and then define the **match** statements and policy actions. Because Layer 7 policy maps are child policies, you must associate a Layer 7 policy map with the appropriate Layer 3 and Layer 4 policy map to provide an entry point for Layer 7 SLB traffic classification. You cannot directly apply a Layer 7 policy map to an interface; you can apply only a Layer 3 and Layer 4 policy map to an interface or globally to all interfaces in a context.

This section contains the following topics:

- [Creating a Layer 7 Policy Map](#)
- [Associating a Layer 7 Class Map with the Layer 7 Policy Map](#)
- [Specifying Layer 7 SLB Policy Actions](#)

## Creating a Layer 7 Policy Map

You can create a Layer 7 SLB policy map by using the **policy-map** command in configuration mode.

The syntax of this command is as follows:

```
policy-map type loadbalance first-match map_name
```

The keywords and arguments are as follows:

- **type loadbalance**—Specifies a load-balancing policy map.
- **first-match**—Defines the execution for the Layer 7 load-balancing policy map. The ACE executes only the action specified against the first-matching classification.
- *map\_name*—Identifier assigned to the policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to create the policy map L7SLBPOLICY, enter:

```
host1/Admin(config)# policy-map type loadbalance first-match  
L7SLBPOLICY
```

After you create a Layer 7 policy map, the CLI enters policy map load-balancing configuration mode.

```
host1/Admin(config-pmap-lb)#
```

To delete an existing policy map, enter:

```
host1/Admin(config)# no policy-map L7SLBPOLICY
```

## Associating a Layer 7 Class Map with the Layer 7 Policy Map

You can associate a class map with the policy map by using the **class** command in policy map load-balancing configuration mode.

The syntax of this command is as follows:

```
class {name1 | class-default} [insert-before name2]
```

The keywords, arguments, and options are as follows:

- *name1*—Name of a previously defined traffic class, configured with the **class-map** command, to associate traffic with the traffic policy. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
- **class-default**—Specifies the reserved, well-known class map created by the ACE. You cannot delete or modify this class. All traffic that fails to meet the other matching criteria in the named class map belongs to the default traffic class. If no specified classification matches the traffic, then the ACE performs the action specified using the **class class-default** command. The **class-default** class map has an implicit **match any** statement in it that enables it to match all traffic.
- **insert-before** *name2*—(Optional) Places the current class map ahead of an existing class map or match statement specified by the *name2* argument in the policy-map configuration. The ACE does not save the sequence reordering as part of the configuration.

For example, to associate the class map L7SLBCLASS with the policy map, enter:

```
host1/Admin(config-pmap-lb)# class L7SLBCLASS
```

After you associate a class map with the policy map, the CLI enters into policy map load-balancing class configuration mode.

```
host1/Admin(config-pmap-lb-c)#
```

The following example shows how to use the **insert-before** option to define the position of a class map in the policy map:

```
host1/Admin(config-pmap-lb)# class L7SLBCLASS insert-before HTTP_CLASS  
host1/Admin(config-pmap-lb-c)#
```



The following example shows how to use the **class class-default** command:

```
host1/Admin(config-pmap-lb)# class class-default
host1/Admin(config-pmap-lb-c)#
```

To remove the association of a class map with the policy map, enter:

```
(config-pmap-lb)# no class L7SLBCLASS
```

## Specifying Layer 7 SLB Policy Actions

After you associate a Layer 7 SLB class map with a Layer 7 SLB policy map or specify inline **match** commands, you need to specify one or more of the following actions that the ACE should take when network traffic matches a class map or inline **match** command:

- Discard requests
- Forward Requests without load balancing
- Enable HTTP header information
- Enable load balancing to a server farm
- Configure a sticky server farm
- Specify the IP differentiated services code point of packets
- Associate an SSL proxy service

This section describes the process of associating an SSL proxy service with the policy map. For details on configuring additional policy actions, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.

You can associate an SSL proxy client service with the policy map by using the **ssl-proxy** command in policy map load-balancing class configuration mode.

The syntax of this command is as follows:

```
ssl-proxy client name
```

The *name* argument is the identifier of an existing SSL proxy client service. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to associate the SSL client proxy service `PSERVICE_CLIENT` with the class map, enter:

```
host1/Admin(config)# policy-map type loadbalance first-match  
L7SLBPOLICY  
host1/Admin(config-pmap-lb)# class L7SLBCLASS  
host1/Admin(config-pmap-lb-c)# ssl-proxy client PSERVICE_CLIENT
```

To remove the association of the SSL client proxy service to the class map, enter:

```
host1/Admin(config-pmap-lb-c)# no ssl-proxy client PSERVICE_CLIENT
```

## Creating a Layer 3 and Layer 4 Class Map for SSL Initiation

The Layer 3 and Layer 4 class map that you associate with a Layer 3 and Layer 4 policy map acts as a filter for traffic that matches the criteria that you specify. For SSL initiation, you can define the match criteria based on one or more of the following traffic characteristics:

- Access list
- Virtual IP address
- Source IP address and subnet mask
- Destination IP address and subnet mask
- TCP/UDP port number or port range

You can create a Layer 3 and Layer 4 class map by using the **class-map** command in the configuration mode. For details on creating and configuring a Layer 3 and Layer 4 class map, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.

# Creating a Layer 3 and Layer 4 Policy Map for SSL Initiation

The Layer 3 and Layer 4 policy map that you create for SSL initiation contains an association with the Layer 7 policy map that the ACE uses for load balancing. Because you can apply only a Layer 3 and Layer 4 policy map directly to a context interface, you need to associate the Layer 7 policy map with the Layer 3 and Layer 4 policy map.

This section contains the following topics:

- [Creating a Layer 3 and Layer 4 Policy Map](#)
- [Associating the Layer 3 and Layer 4 Class Map with the Policy Map](#)
- [Associating a Layer 7 Policy Map with the Class Map](#)

## Creating a Layer 3 and Layer 4 Policy Map

You can create a Layer 3 and Layer 4 policy map by using the **policy-map** command in configuration mode.

The syntax of this command is as follows:

```
policy-map multi-match policy_name
```

The *policy\_name* argument is the name that you assign to the policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to create the policy map L4SLBPOLICY, enter:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
```

After you create a policy map, the CLI enters into policy map configuration mode.

```
host1/Admin(config-pmap)#
```

To delete an existing policy map, enter:

```
host1/Admin(config)# no policy-map L4SLBPOLICY
```

For information on associating an SSL class map with the policy map, see the [“Associating a Layer 7 Class Map with the Layer 7 Policy Map”](#) section.

## Associating the Layer 3 and Layer 4 Class Map with the Policy Map

You can associate the Layer 3 and Layer 4 class map with the policy map by using the **class** command in policy map configuration mode.

The syntax of this command is as follows:

```
class class-map
```

The *class-map* argument is the name of an existing class map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to associate the class map L4SLBCLASS with the policy map, enter:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY  
host1/Admin(config-pmap)# class L4SLBCLASS
```

After you associate a class map with the policy map, the CLI enters policy map class configuration mode.

```
host1/Admin(config-pmap-c)#
```

To remove the association of a class map to the policy map, enter:

```
host1/Admin(config-pmap)# no class L4SLBCLASS
```

## Associating a Layer 7 Policy Map with the Class Map

You can associate a Layer 7 policy map with the Layer 3 and Layer 4 class map by using the **loadbalance** command in policy map class configuration mode. This association nests the Layer 7 policy map within the Layer 3 and Layer 4 policy map that the ACE applies directly to the traffic.

The syntax of this command is as follows:

```
loadbalance policy polycymap
```

The **policy** *polycymap* keyword and argument specify the name of an existing Layer 7 policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to associate the Layer 7 policy map L7SLBPOLICY with the class map, enter:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)# class L4SLBCLASS
host1/Admin(config-pmap-c)# loadbalance policy L7SLBPOLICY
```

To remove the association of the Layer 7 policy map with the class map, enter:

```
host1/Admin(config-pmap-c)# no loadbalance policy L7SLBPOLICY
```

## Applying the Policy Map to the VLANs

This section describes how to apply the Layer 3 and Layer 4 policy map to the VLAN traffic. The ACE allows you to apply the policy globally to all VLANs within the current context or to a specific VLAN in the context.

This section contains the following topics:

- [Applying the Policy Map Globally](#)
- [Applying the Policy Map to a Specific VLAN](#)

## Applying the Policy Map Globally

You can globally apply the policy map to all of the VLANs in the context by using the **service-policy** command in configuration mode.

The syntax of this command is as follows:

```
service-policy input policy_name
```

The *policy\_name* argument is the name of an existing policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to globally apply the policy map L4SLBPOLICY to all of the context VLANs, enter:

```
host1/Admin(config)# service-policy input L4SLBPOLICY
```

To globally remove a policy map from all VLANs, enter:

```
host1/Admin(config)# no service-policy input L4SLBPOLICY
```

## Applying the Policy Map to a Specific VLAN

To apply a policy map to a specific VLAN interface, you must enter interface configuration mode by using the **interface** command in configuration mode.

The syntax of this command is as follows:

```
interface vlan vlan
```

The *vlan* argument is the context VLAN number. Enter an integer from 2 to 4094.

For example, to enter into interface configuration mode for VLAN 10, enter:

```
host1/Admin(config)# interface vlan 10  
host1/Admin(config-if)#
```

You can apply the policy map to the interface by using the **service-policy** command in interface configuration mode.

The syntax of this command is as follows:

```
service-policy input policy-name
```

The *policy-name* argument is the name of an existing policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to apply the policy map L4SLBPOLICY to VLAN 10, enter:

```
host1/Admin(config)# interface vlan 10  
host1/Admin(config-if)# service-policy input L4SLBPOLICY
```

To remove the policy from the interface, enter:

```
host1/Admin(config-if)# no service-policy input L4SLBPOLICY
```

## Example of an SSL Initiation Configuration

The following example illustrates a running configuration of the ACE acting as a SSL proxy client, initiating and maintaining an SSL connection between itself and an SSL server. The ACE receives clear text from an HTTP client, and then encrypts and transmits the data as cipher text to the SSL server. On the reverse

side, the ACE decrypts the cipher text that it receives from the SSL server and sends the data to the client as clear text. The SSL initiation configuration appears in bold in the example.

```
access-list ACL1 line 10 extended permit ip any any
```

```
probe http GEN-HTTP
  port 80
  interval 50
  faildetect 5
  expect status 200 200
```

```
serverfarm host SFARM1
  description SERVER FARM 1 FOR SSL INITIATION
  probe GEN_HTTP
  rserver SERVER1 443
    inservice
  rserver SERVER2 443
    inservice
  rserver SERVER3 443
    inservice
  rserver SERVER4 443
    inservice
```

```
serverfarm host SFARM2
  description SERVER FARM 2 FOR SSL TERMINATION
  probe GEN_HTTP
  rserver SERVER5 443
    inservice
  rserver SERVER6 443
    inservice
  rserver SERVER7 443
    inservice
  rserver SERVER8 443
    inservice
```

```
parameter-map type http PARAMMAP_HTTP
  server-conn reuse
  case-insensitive
  persistence-rebalance
parameter-map type ssl PARAMMAP_SSL_INITIATION
  cipher RSA_WITH_RC4_128_MD5
  cipher RSA_WITH_RC4_128_SHA
  cipher RSA_WITH_DES_CBC_SHA
  cipher RSA_WITH_3DES_EDE_CBC_SHA
  cipher RSA_WITH_AES_128_CBC_SHA
  cipher RSA_WITH_AES_256_CBC_SHA
  cipher RSA_EXPORT_WITH_RC4_40_MD5
```

### Example of an SSL Initiation Configuration

```

cipher RSA_EXPORT1024_WITH_RC4_56_MD5
cipher RSA_EXPORT_WITH_DES40_CBC_SHA
cipher RSA_EXPORT1024_WITH_DES_CBC_SHA
cipher RSA_EXPORT1024_WITH_RC4_56_SHA
version all
parameter-map type connection TCP_PARAM
syn-data drop
exceed-mss allow

ssl-proxy service SSL_PSERVICE_CLIENT
ssl advanced-options PARAMMAP_SSL_INITIATION

class-map type http loadbalance match-all L7_SERVER_CLASS
description Sticky for SSL Testing
2 match http url .*\.jpg
3 match source-address 192.168.130.0 255.255.255.0
class-map type http loadbalance match-all L7_SLB-HTTP_CLASS
2 match http url .*
3 match source-address 192.168.130.0 255.255.255.0
class-map match-all L4_SSL-INIT_CLASS
description SSL Initiation VIP
2 match virtual-address 192.168.130.12 tcp eq www
policy-map type loadbalance first-match L7_SSL-INIT_POLICY
class L7_SERVER_CLASS
serverfarm SFARM1
insert-http SRC_IP header-value "%is"
insert-http I_AM header-value "SSL_INIT"
insert-http SRC_Port header-value "%ps"
insert-http DEST_IP header-value "%id"
insert-http DEST_Port header-value "%pd"
ssl-proxy client SSL_PSERVICE_CLIENT
class L7_SLB-HTTP_CLASS
serverfarm SFARM2
insert-http SRC_IP header-value "%is"
insert-http I_AM header-value "SSL_INIT"
insert-http DEST_Port header-value "%pd"
insert-http DEST_IP header-value "%id"
insert-http SRC_Port header-value "%ps"
ssl-proxy client SSL_PSERVICE_CLIENT
policy-map multi-match L4_SSL-VIP_POLICY
class L4_SSL-INIT_CLASS
loadbalance vip inservice
loadbalance policy L7_SSL-INIT_POLICY
loadbalance vip icmp-reply active
appl-parameter http advanced-options PARAMMAP_HTTP
connection advanced-options TCP_PARAM

interface vlan 120

```



```
description Upstream VLAN_120 - Clients and VIPs
ip address 192.168.120.1 255.255.255.0
fragment chain 20
fragment min-mtu 68
access-group input ACL1
nat-pool 1 192.168.120.70 192.168.120.80 netmask 255.255.255.0 pat
service-policy input L4_SSL-VIP_POLICY
no shutdown
ip route 10.1.0.0 255.255.255.0 192.168.120.254
```

■ Example of an SSL Initiation Configuration