



General VPN Parameters

The ASA implementation of virtual private networking includes useful features that do not fit neatly into categories. This chapter describes some of these features. It includes the following sections:

- [SSL VPN in this chapter refers to the SSL VPN client \(AnyConnect 2.x or its predecessor, SVC 1.x\), unless clientless \(browser-based\) SSL VPN is specified. Configuring IPsec to Bypass ACLs, page 3-1](#)
- [Permitting Intra-Interface Traffic \(Hairpinning\), page 3-2](#)
- [Setting Maximum Active IPsec or SSL VPN Sessions, page 3-3](#)
- [Using Client Update to Ensure Acceptable IPsec Client Revision Levels, page 3-4](#)
- [Implementing NAT-Assigned IP to Public IP Connection, page 3-6](#)
- [Configuring Load Balancing, page 3-12](#)
- [Configuring VPN Session Limits, page 3-17](#)
- [Configuring the Pool of Cryptographic Cores, page 3-18](#)
- [Configuring ISE Policy Enforcement, page 3-22](#)
-

SSL VPN in this chapter refers to the SSL VPN client (AnyConnect 2.x or its predecessor, SVC 1.x), unless clientless (browser-based) SSL VPN is specified. **Configuring IPsec to Bypass ACLs**

To permit any packets that come from an IPsec tunnel without checking ACLs for the source and destination interfaces, enter the **sysopt connection permit-vpn** command in global configuration mode.

You might want to bypass interface ACLs for IPsec traffic if you use a separate VPN concentrator behind the ASA and want to maximize the ASA performance. Typically, you create an ACL that permits IPsec packets by using the **access-list** command and apply it to the source interface. Using an ACL is more secure because you can specify the exact traffic you want to allow through the ASA.

The syntax is **sysopt connection permit-vpn**. The command has no keywords or arguments.

The following example enables IPsec traffic through the ASA without checking ACLs:

```
hostname(config)# sysopt connection permit-vpn
```

**Note**

Decrypted through-traffic is permitted from the client despite having an access group on the outside interface, which calls a **deny ip any any** ACL, while **no sysopt connection permit-vpn** is configured.

Users who want to control access to the protected network via site-to-site or remote access VPN using the **no sysopt permit-vpn** command in conjunction with an access control list (ACL) on the outside interface are not successful.

In this situation, when management-access inside is enabled, the ACL is not applied, and users can still connect to the ASA using SSH. Traffic to hosts on the inside network is blocked correctly by the ACL, but decrypted through-traffic to the inside interface is not blocked.

The **ssh** and **http** commands are of a higher priority than the ACLs. In other words, to deny SSH, Telnet, or ICMP traffic to the box from the VPN session, use **ssh**, **telnet** and **icmp** commands.

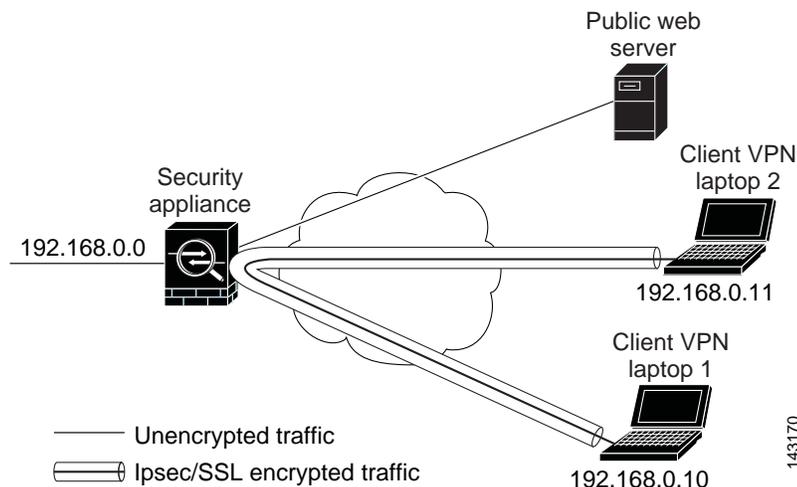
Permitting Intra-Interface Traffic (Hairpinning)

The ASA includes a feature that lets a VPN client send IPsec-protected traffic to another VPN user by allowing such traffic in and out of the same interface. Also called “hairpinning”, this feature can be thought of as VPN spokes (clients) connecting through a VPN hub (ASA).

In another application, hairpinning can redirect incoming VPN traffic back out through the same interface as unencrypted traffic. This would be useful, for example, to a VPN client that does not have split tunneling but needs to both access a VPN and browse the web.

Figure 3-1 shows VPN Client 1 sending secure IPsec traffic to VPN Client 2 while also sending unencrypted traffic to a public web server.

Figure 3-1 VPN Client Using Intra-Interface Feature for Hairpinning



To configure this feature, use the **same-security-traffic** command in global configuration mode with its **intra-interface** argument.

The command syntax is **same-security-traffic permit {inter-interface | intra-interface}**.

The following example shows how to enable intra-interface traffic:

```
hostname(config)# same-security-traffic permit intra-interface
hostname(config)#
```



Note

You use the **same-security-traffic** command, but with the **inter-interface** argument, to permit communication between interfaces that have the same security level. This feature is not specific to IPsec connections. For more information, see the “Configuring Interface Parameters” chapter of this guide.

To use hairpinning, you must apply the proper NAT rules to the ASA interface, as discussed in the following section.

NAT Considerations for Intra-Interface Traffic

For the ASA to send unencrypted traffic back out through the interface, you must enable NAT for the interface so that publicly routable addresses replace your private IP addresses (unless you already use public IP addresses in your local IP address pool). The following example applies an interface PAT rule to traffic sourced from the client IP pool:

```
hostname(config)# ip local pool clientpool 192.168.0.10-192.168.0.100
hostname(config)# object network vpn_nat
hostname(config-network-object)# subnet 192.168.0.0 255.255.255.0
hostname(config-network-object)# nat (outside,outside) interface
```

When the ASA sends encrypted VPN traffic back out this same interface, however, NAT is optional. The VPN-to-VPN hairpinning works with or without NAT. To apply NAT to all outgoing traffic, implement only the commands above. To exempt the VPN-to-VPN traffic from NAT, add commands (to the example above) that implement NAT exemption for VPN-to-VPN traffic, such as:

```
hostname(config)# nat (outside,outside) source static vpn_nat vpn_nat destination static
vpn_nat vpn_nat
```

For more information on NAT rules, see the “Applying NAT” chapter of this guide.

Setting Maximum Active IPsec or SSL VPN Sessions

To limit VPN sessions to a lower value than the ASA allows, enter the **vpn-sessiondb** command in global configuration mode:

```
vpn-sessiondb {max-anyconnect-premium-or-essentials-limit <number> |
max-other-vpn-limit <number>}
```

The **max-anyconnect-premium-or-essentials-limit** keyword specifies the maximum number of AnyConnect sessions, from 1 to the maximum sessions allowed by the license.

The **max-other-vpn-limit** keyword specifies the maximum number of VPN sessions other than AnyConnect client sessions, from 1 to the maximum sessions allowed by the license. This includes the Cisco VPN client (IPsec IKEv1), Lan-to-Lan VPN, and clientless SSL VPN sessions.

This limit affects the calculated load percentage for VPN Load Balancing.

The following example shows how to set a maximum Anyconnect VPN session limit of 450:

```
hostname(config)# vpn-sessiondb max-anyconnect-premium-or-essentials-limit 450
hostname(config)#
```

Using Client Update to Ensure Acceptable IPsec Client Revision Levels



Note

The information in this section applies to IPsec connections only.

The client update feature lets administrators at a central location automatically notify VPN client users that it is time to update the VPN client software and the VPN 3002 hardware client image.

Remote users might be using outdated VPN software or hardware client versions. You can use the **client-update** command at any time to enable updating client revisions; specify the types and revision numbers of clients to which the update applies; provide a URL or IP address from which to get the update; and, in the case of Windows clients, optionally notify users that they should update their VPN client version. For Windows clients, you can provide a mechanism for users to accomplish that update. For VPN 3002 hardware client users, the update occurs automatically, with no notification. This command applies only to the IPsec remote-access tunnel-group type.

To perform a client update, enter the **client-update** command in either general configuration mode or tunnel-group ipsec-attributes configuration mode. If the client is already running a software version on the list of revision numbers, it does not need to update its software. If the client is not running a software version on the list, it should update. The following procedure explains how to perform a client update:

Step 1 In global configuration mode, enable client update by entering this command:

```
hostname(config)# client-update enable
hostname(config)#
```

Step 2 In global configuration mode, specify the parameters for the client update that you want to apply to all clients of a particular type. That is, specify the type of client, the URL or IP address from which to get the updated image, and the acceptable revision number or numbers for that client. You can specify up to four revision numbers, separated by commas.

If the user's client revision number matches one of the specified revision numbers, there is no need to update the client. This command specifies the client update values for all clients of the specified type across the entire ASA.

Use this syntax:

```
hostname(config)# client-update type type url url-string rev-nums rev-numbers
hostname(config)#
```

The available client types are **win9X** (includes Windows 95, Windows 98 and Windows ME platforms), **winnt** (includes Windows NT 4.0, Windows 2000 and Windows XP platforms), **windows** (includes all Windows based platforms), and **vpn3002** (VPN 3002 hardware client).

If the client is already running a software version on the list of revision numbers, it does not need to update its software. If the client is not running a software version on the list, it should update. You can specify up to three of these client update entries. The keyword **windows** covers all of the allowable Windows platforms. If you specify **windows**, do not specify the individual Windows client types.



Note

For all Windows clients, you must use the protocol `http://` or `https://` as the prefix for the URL. For the VPN 3002 hardware client, you must specify protocol `tftp://` instead.

The following example configures client update parameters for the remote access tunnel group. It designates the revision number 4.6.1 and the URL for retrieving the update, which is `https://support/updates`.

```
hostname(config)# client-update type windows url https://support/updates/ rev-nums 4.6.1
hostname(config)#
```

Alternatively, you can configure client update just for individual tunnel groups, rather than for all clients of a particular type. (See Step 3.)

VPN 3002 clients update without user intervention and users receive no notification message. The following example applies only to VPN 3002 hardware clients. Entered in tunnel-group ipsec-attributes configuration mode the command it configures client update parameters for the IPsec remote access tunnel group **salesgrp**. This example designates the revision number, 4.7 and uses the TFTP protocol for retrieving the updated software from the site with the IP address 192.168.1.1:

```
hostname(config)# tunnel-group salesgrp type ipsec-ra
hostname(config)# tunnel-group salesgrp ipsec-attributes
hostname(config-tunnel-ipsec)# client-update type vpn3002 url tftp:192.168.1.1 rev-nums
4.7
hostname(config-tunnel-ipsec)#
```



Note

You can have the browser automatically start an application by including the application name at the end of the URL; for example: `https://support/updates/vpnclient.exe`.

Step 3 Define a set of client-update parameters for a particular ipsec-ra tunnel group.

In tunnel-group ipsec-attributes mode, specify the tunnel group name and its type, the URL or IP address from which to get the updated image, and a revision number. If the user's client's revision number matches one of the specified revision numbers, there is no need to update the client, for example, for a Windows client enter this command:

```
hostname(config)# tunnel-group remotegrp type ipsec-ra
hostname(config)# tunnel-group remotegrp ipsec-attributes
hostname(config-tunnel-ipsec)# client-update type windows url https://support/updates/
rev-nums 4.6.1
hostname(config-tunnel-ipsec)#
```

Step 4 (Optional) Send a notice to active users with outdated Windows clients that their client needs updating. For these users, a pop-up window appears, offering them the opportunity to launch a browser and download the updated software from the site that you specified in the URL. The only part of this message that you can configure is the URL. (See Step 2 or 3.) Users who are not active get a notification message the next time they log on. You can send this notice to all active clients on all tunnel groups, or you can send it to clients on a particular tunnel group. For example, to notify all active clients on all tunnel groups, enter the following command in privileged EXEC mode:

```
hostname# client-update all
hostname#
```

If the user's client's revision number matches one of the specified revision numbers, there is no need to update the client, and no notification message is sent to the user. VPN 3002 clients update without user intervention and users receive no notification message.

**Note**

If you specify the client-update type as **windows** (specifying all Windows-based platforms) and later want to enter a client-update type of **win9x** or **winnt** for the same entity, you must first remove the windows client type with the **no** form of the command, then use new client-update commands to specify the new client types.

Implementing NAT-Assigned IP to Public IP Connection

In rare situations, you might want to use a VPN peer's real IP address on the inside network instead of an assigned local IP address. Normally with VPN, the peer is given an assigned local IP address to access the inside network. However, you might want to translate the local IP address back to the peer's real public address if, for example, your inside servers and network security is based on the peer's real IP address.

Cisco ASA 55xx introduced a way to translate the VPN client's assigned IP address on the internal/protected network to its public (source) IP address. This feature supports the scenario where the target servers/services on the internal network and network security policy require communication with the VPN client's public/source IP instead of the assigned IP on the internal corporate network.

You can enable this feature on one interface per tunnel group. Object NAT rules are dynamically added and deleted when the VPN session is established or disconnected.

Limitations

Because of routing issues, we do not recommend using this feature unless you know you need it.

- Only supports legacy Cisco VPN client (IKEv1) and AnyConnect clients.
- Return traffic to the public IP addresses must be routed back to the ASA so the NAT policy and VPN policy can be applied.
- Only supports IPv4 assigned and public addresses.
- Multiple peers behind a NAT/PAT device are not supported.
- Does not support load balancing (because of routing issue).
- Does not support roaming.

Detailed Steps

Step 1 In global configuration mode, enter **tunnel general**.

Step 2 Use this syntax to enable the address translation:

```
hostname(config-tunnel-general)# nat-assigned-to-public-ip <interface>
```

This command dynamically installs NAT policies of the assigned IP address to the public IP address of the source. The *interface* determines where to apply NAT.

Step 3 Use this syntax to disable the address translation:

```
hostname(config-tunnel-general)# no nat-assigned-to-public-ip
```

Displaying VPN NAT Policies

Address translation uses the underlying object NAT mechanisms; therefore, the VPN NAT policy displays just like manually configured object NAT policies. This example uses 95.1.226.4 as the assigned IP and 75.1.224.21 as the peer's public IP:

```
prompt# show nat
Auto NAT Policies (Section 2)
1 (outside) to (inside) source static _vpn_nat_95.1.226.4 75.1.224.21
  translate_hits = 315, untranslate_hits = 315

prompt# show nat detail

Auto NAT Policies (Section 2)
1 (outside) to (inside) source static _vpn_nat_95.1.226.4 75.1.224.21
  translate_hits = 315, untranslate_hits = 315
  Source - Origin: 95.1.226.4/32, Translated: 75.1.224.21/32
```

Outside is the interface to which the AnyConnect client connects and *inside* is the interface specific to the new tunnel group.

**Note**

Since VPN NAT policies are dynamic and not added to the configuration, the VPN NAT object and NAT policy are hidden from the show run object and show run nat reports.

Understanding Load Balancing

If you have a remote-access configuration in which you are using two or more ASAs or VPN Concentrators connected on the same network, you can configure these devices to share their session load. This feature is called *load balancing*. To implement load balancing, you group together logically two or more devices on the same private LAN-to-LAN network, private subnet, and public subnet into a *virtual cluster*.

All devices in the virtual cluster carry session loads. Load balancing directs session traffic to the least-loaded device in the cluster, which distributes the load among all devices. It makes efficient use of system resources and provides increased performance and high availability.

One device in the virtual cluster, the *virtual cluster master*, directs incoming traffic to the other devices, called *backup devices*. The virtual cluster master monitors all devices in the cluster, keeps track of how busy each is, and distributes the session load accordingly. The role of virtual cluster master is not tied to a physical device; it can shift among devices. For example, if the current virtual cluster master fails, one of the backup devices in the cluster takes over that role and immediately becomes the new virtual cluster master.

The virtual cluster appears to outside clients as a single *virtual cluster IP address*. This IP address is not tied to a specific physical device. This address belongs to the current virtual cluster master, which makes it virtual. A VPN client attempting to establish a connection connects first to this virtual cluster IP address. The virtual cluster master then sends back to the client the public IP address of the least-loaded available host in the cluster. In a second transaction (transparent to the user), the client connects directly to that host. In this way, the virtual cluster master directs traffic evenly and efficiently across resources.

**Note**

All clients other than the Cisco VPN client or the Cisco 3002 hardware client should connect directly to the ASA as usual; they do not use the virtual cluster IP address.

If a machine in the cluster fails, the terminated sessions can immediately reconnect to the virtual cluster IP address. The virtual cluster master then directs these connections to another active device in the cluster. If the virtual cluster master itself fails, a backup device in the cluster immediately and automatically takes over as the new virtual session master. Even if several devices in the cluster fail, users can continue to connect to the cluster as long as any one device in the cluster is up and available.

Comparing Load Balancing to Failover

Both load balancing and failover are high-availability features, but they function differently and have different requirements. In some circumstances you can use both load balancing and failover. The following sections describe the differences between these features.

Load Balancing

Load balancing is a mechanism for equitably distributing remote-access VPN traffic among the devices in a virtual cluster. It is based on simple distribution of traffic without taking into account throughput or other factors. A load-balancing cluster consists of two or more devices, one is the virtual master, and the other devices are the backup. These devices do not need to be of the exact same type, or have identical software versions or configurations.

All active devices in a virtual cluster carry session loads. Load balancing directs traffic to the least-loaded device in the cluster, distributing the load among all devices. It makes efficient use of system resources and provides increased performance and high availability.

Failover

A failover configuration requires two identical ASAs connected to each other through a dedicated failover link and, optionally, a stateful failover link. The health of the active interfaces and units is monitored to determine when specific failover conditions are met. If those conditions occur, failover occurs. Failover supports both VPN and firewall configurations.

The ASA supports two failover configurations: Active/Active failover and Active/Standby failover.

With Active/Active failover, both units can pass network traffic. This is not true load balancing, although it might appear to have the same effect. When failover occurs, the remaining active unit takes over passing the combined traffic, based on the configured parameters. Therefore, when configuring Active/Active failover, you must make sure that the combined traffic for both units is within the capacity of each unit.

With Active/Standby failover, only one unit passes traffic, while the other unit waits in a standby state and does not pass traffic. Active/Standby failover lets you use a second ASA to take over the functions of a failed unit. When the active unit fails, it changes to the standby state, while the standby unit changes to the active state. The unit that becomes active assumes the IP addresses (or, for transparent firewall, the management IP address) and MAC addresses of the failed unit and begins passing traffic. The unit that is now in standby state takes over the standby IP addresses of the active unit. If an active unit fails, the standby takes over without any interruption to the client VPN tunnel.

Implementing Load Balancing

Enabling load balancing involves:

- Configuring the load-balancing cluster by establishing a common virtual cluster IP address, UDP port (if necessary), and IPsec shared secret for the cluster. You configure these values identically for every device in the cluster.
- Configuring the participating device by enabling load balancing on the device and defining device-specific properties. These values vary from device to device.

**Note**

VPN load balancing requires an active 3DES/AES license. The ASA checks for the existence of this crypto license before enabling load balancing. If it does not detect an active 3DES or AES license, the ASA prevents the enabling of load balancing and also prevents internal configuration of 3DES by the load balancing system unless the license permits this usage.

Prerequisites

Load balancing is disabled by default. You must explicitly enable load balancing.

You must have first configured the public (outside) and private (inside) interfaces and also have previously configured the interface to which the virtual cluster IP address refers. You can use the **interface** and **nameif** commands to configure different names for these interfaces. Subsequent references in this section use the names outside and inside.

All devices that participate in a cluster must share the same cluster-specific values: IP address, encryption settings, encryption key, and port.

Eligible Platforms

A load-balancing cluster can include ASA models ASA 5512-X (with a Security Plus license) and Model 5515-X and above. You can also include Cisco VPN 3000 series concentrators in the cluster. While mixed configurations are possible, administration is generally simpler if the cluster is homogeneous.

Eligible Clients

Load balancing is effective only on remote sessions initiated with the following clients:

- Cisco AnyConnect VPN client (Release 2.0 and later)
- Cisco VPN Client (Release 3.0 and later)
- Cisco ASA 5505 ASA (when acting as an Easy VPN client)
- Cisco VPN 3002 hardware client (Release 3.5 or later)
- Cisco PIX 501/506E when acting as an Easy VPN client
- Cisco IOS EZVPN client devices supporting IKE-redirect (IOS 831/871)
- Clientless SSL VPN (not a client)

Load balancing works with IPsec clients and SSL VPN client and clientless sessions. All other VPN connection types (L2TP, PPTP, L2TP/IPsec), including LAN-to-LAN, can connect to an ASA on which load balancing is enabled, but they cannot participate in load balancing.

VPN Load-Balancing Algorithm

The master device maintains a sorted list of backup cluster members in ascending IP address order. The load of each backup cluster member is computed as an integer percentage (the number of active sessions). AnyConnect inactive sessions do not count towards the SSL VPN load for load balancing. The master device redirects the IPsec and SSL VPN tunnel to the device with the lowest load until it is 1 percent higher than the rest. When all backup cluster members are 1% higher than the master, the master device redirects to itself.

For example, if you have one master and two backup cluster members, the following cycle applies:



Note All nodes start with 0%, and all percentages are rounded half-up.

1. The master device takes the connection if all members have a load at 1% higher than the master.
2. If the master does not take the connection, the session is taken by whichever backup device has the least load percentage.
3. If all members have the same percentage load, the backup device with the least number of sessions gets the session.
4. If all members have the same percentage load and the same number of sessions, the device with the least IP addresses gets the session.

VPN Load-Balancing Cluster Configurations

A load-balancing cluster can consist of ASAs of the same release, of mixed releases, as well as VPN 3000 concentrators, or a mixture of these, subject to the following restrictions:

- Load-balancing clusters that consist of same release ASAs or all VPN 3000 concentrators can run load balancing for a mixture of IPsec, AnyConnect, and clientless SSL VPN sessions.
- Load-balancing clusters that consist of both same release ASAs and VPN 3000 concentrators can run load balancing for a mixture of IPsec, AnyConnect, and clientless SSL VPN client and clientless sessions.
- Load-balancing clusters that include mixed release ASAs or same release ASAs and VPN 3000 concentrators or both can support only IPsec sessions. In such a configuration, however, the ASAs might not reach their full IPsec capacity. [Scenario 1: Mixed Cluster with No SSL VPN Connections](#), illustrates this situation.

Since Release 7.1(1), IPsec and SSL VPN sessions count or weigh equally in determining the load that each device in the cluster carries. This is a change from the load-balancing calculation for the ASA Release 7.0(x) software and the VPN 3000 concentrator. Both platforms use a weighting algorithm that on some hardware platforms calculates the SSL VPN session load differently from the IPsec session load.

The virtual master of the cluster assigns session requests to the members of the cluster. The ASA regards all sessions, SSL VPN or IPsec, as equal and assigns them accordingly. You can configure the number of IPsec and SSL VPN sessions to allow up to the maximum allowed by your configuration and license. See [Configuring VPN Session Limits](#) for a description of how to set these limits.

We have tested up to ten nodes in a load-balancing cluster. Larger clusters might work, but we do not officially support such topologies.

Some Typical Mixed Cluster Scenarios

If you have a mixed configuration—that is, if your load-balancing cluster includes devices running a mixture of ASA software releases or at least one ASA running ASA Release 7.1(1) or later and a VPN 3000 concentrator—the difference in weighting algorithms becomes an issue if the initial cluster master fails and another device takes over as master.

The following scenarios illustrate the use of VPN load balancing in clusters consisting of a mixture of ASAs running ASA Release 7.1(1) and ASA Release 7.0(x) software, as well as VPN 3000 series concentrators.

Scenario 1: Mixed Cluster with No SSL VPN Connections

In this scenario, the cluster consists of a mixture of ASAs and VPN 3000 concentrators. Some of the ASA cluster peers are running ASA Release 7.0(x), and some are running Release 7.1(1). The pre-7.1(1) and VPN 3000 peers do not have any SSL VPN connections, and the 7.1(1) cluster peers have only the base SSL VPN license, which allows two SSL VPN sessions, but there are no SSL VPN connections. In this case, all the connections are IPsec, and load balancing works fine.

The two SSL VPN licenses have a very small effect on the user's taking advantage of the maximum IPsec session limit, and then only when a VPN 3000 concentrator is the cluster master. In general, the smaller the number of SSL VPN licenses is on a ASA in a mixed cluster, the smaller the effect on the ASA 7.1(1) device being able to reach its IPsec session limit in a scenario where there are only IPsec sessions.

Scenario 2: Mixed Cluster Handling SSL VPN Connections

Suppose, for example, an ASA running ASA Release 7.1(1) software is the initial cluster master and then that device fails. Another device in the cluster takes over automatically as master and applies its own load-balancing algorithm to determine processor loads within the cluster. A cluster master running ASA Release 7.1(1) software cannot weight session loads in any way other than what that software provides. Therefore, it cannot assign a combination of IPsec and SSL VPN session loads properly to ASA devices running earlier versions nor to VPN 3000 concentrators. Conversely, a VPN 3000 concentrator acting as the cluster master cannot assign loads properly to an ASA Release 7.1(1) ASA. The following scenario illustrates this dilemma.

This scenario is similar to the previous one, in that the cluster consists of a mixture of ASAs and VPN 3000 concentrators. Some of the ASA cluster peers are running ASA Release 7.0(x) and some are running Release 7.1(1). In this case, however, the cluster is handling SSL VPN connections as well as IPsec connections.

If a device that is running software earlier than ASA Release 7.1(1) is the cluster master, the master applies the protocol and logic in effect prior to Release 7.1(1). That is, sessions might be directed to load-balancing peers that have exceeded their session limit. In that case, the user is denied access.

If the cluster master is a device running ASA Release 7.0(x) software, the old session-weighting algorithm applies only to the pre-7.1(1) peers in the cluster. No one should be denied access in this case. Because the pre-7.1(1) peers use the session-weighting algorithm, they are more lightly loaded.

An issue arises, however, because you cannot guarantee that the 7.1(1) peer is always the cluster master. If the cluster master fails, another peer assumes the role of master. The new master might be any of the eligible peers. Because of the unpredictability of the results, we recommend that you avoid configuring this type of cluster.

Configuring Load Balancing

To use load balancing, configure the following elements for each device that participates in the cluster:

- Public and private interfaces
- VPN load-balancing cluster attributes



Note

All participants in the cluster must have an identical cluster configuration, except for the device priority within the cluster.



Note

The Local CA feature is not supported if you use Active/Active stateful failover or VPN load-balancing. The Local CA cannot be subordinate to another CA; it can act only as the Root CA.

Configuring the Public and Private Interfaces for Load Balancing

To configure the public (outside) and private (inside) interfaces for the load-balancing cluster devices, do the following steps:

- Step 1** Configure the public interface on the ASA by entering the **interface** command with the **lbpublic** keyword in vpn-load-balancing configuration mode. This command specifies the name or IP address of the public interface for load balancing for this device:

```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# interface lbpublic outside
hostname(config-load-balancing)#
```

- Step 2** Configure the private interface on the ASA by entering the **interface** command with the **lbprivate** keyword in vpn-load-balancing configuration mode. This command specifies the name or IP address of the private interface for load balancing for this device:

```
hostname(config-load-balancing)# interface lbprivate inside
hostname(config-load-balancing)#
```

- Step 3** Set the priority to assign to this device within the cluster. The range is from 1 to 10. The priority indicates the likelihood of this device becoming the virtual cluster master, either at startup or when an existing master fails. The higher you set the priority (for example, 10), the more likely it is that this device becomes the virtual cluster master.

```
hostname(config-load-balancing)# priority number
hostname(config-load-balancing)#
```

For example, to assign this device a priority of 6 within the cluster, enter the following command:

```
hostname(config-load-balancing)# priority 6
hostname(config-load-balancing)#
```

- Step 4** If you want to apply network address translation for this device, enter the **nat** command with the NAT assigned address for the device. You can define an IPv4 and an IPv6 address or specify the device's hostname.

```
hostname(config-load-balancing)# nat ipv4_address ipv_address
hostname(config-load-balancing)#
```

For example, to assign this device a NAT address of 192.168.30.3 and 2001:DB8::1, enter the following command:

```
hostname(config-load-balancing)# nat 192.168.30.3 2001:DB8::1
hostname(config-load-balancing)#
```

Configuring the Load Balancing Cluster Attributes

To configure the load-balancing cluster attributes for each device in the cluster, do the following steps:

- Step 1** Set up VPN load balancing by entering the **vpn load-balancing** command in global configuration mode:

```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)#
```

This enters vpn-load-balancing configuration mode, in which you can configure the remaining load-balancing attributes.

- Step 2** Configure the IP address or the fully qualified domain name of the cluster to which this device belongs. This command specifies the single IP address or FQDN that represents the entire virtual cluster. Choose an IP address that is within the public subnet address range shared by all the ASAs in the virtual cluster. You can specify an IPv4 or IPv6 address.

```
hostname(config-load-balancing)# cluster ip address ip_address
hostname(config-load-balancing)#
```

For example, to set the cluster IP address to IPv6 address, 2001:DB8::1, enter the following command:

```
hostname(config-load-balancing)# cluster ip address 2001:DB8::1
hostname(config-load-balancing)#
```

- Step 3** Configure the cluster port. This command specifies the UDP port for the virtual cluster in which this device is participating. The default value is 9023. If another application is using this port, enter the UDP destination port number that you want to use for load balancing.

```
hostname(config-load-balancing)# cluster port port_number
hostname(config-load-balancing)#
```

For example, to set the cluster port to 4444, enter the following command:

```
hostname(config-load-balancing)# cluster port 4444
hostname(config-load-balancing)#
```

- Step 4** (Optional) Enable IPsec encryption for the cluster. The default is no encryption. This command enables or disables IPsec encryption. If you configure this check attribute, you must first specify and verify a shared secret. The ASAs in the virtual cluster communicate via LAN-to-LAN tunnels using IPsec. To ensure that all load-balancing information communicated between the devices is encrypted, enable this attribute.

```
hostname(config-load-balancing)# cluster encryption
hostname(config-load-balancing)#
```



Note When using encryption, you must have previously configured the load-balancing inside interface. If that interface is not enabled on the load-balancing inside interface, you get an error message when you try to configure cluster encryption.

If the load-balancing inside interface was enabled when you configured cluster encryption, but was disabled before you configured the participation of the device in the virtual cluster, you get an error message when you enter the **participate** command (or, in ASDM, check the **Participate in Load Balancing Cluster** check box), and encryption is not enabled for the cluster.

To use cluster encryption, you must enable ISAKMP on the inside interface, using the **crypto isakmp enable** command with the inside interface specified.

- Step 5** If you enable cluster encryption, you must also specify the IPsec shared secret by entering the **cluster key** command. This command specifies the shared secret between IPsec peers when you have enabled IPsec encryption. The value you enter in the box appears as consecutive asterisk characters

```
hostname(config-load-balancing)# cluster key shared_secret
hostname(config-load-balancing)#
```

For example, to set the shared secret to 123456789, enter the following command:

```
hostname(config-load-balancing)# cluster key 123456789
hostname(config-load-balancing)#
```

- Step 6** Enable this device's participation in the cluster by entering the **participate** command:

```
hostname(config-load-balancing)# participate
hostname(config-load-balancing)#
```

Enabling Redirection Using a Fully Qualified Domain Name

To enable or disable redirection using a fully qualified domain name in vpn load-balancing mode, use the **redirect-fqdn enable** command in global configuration mode. This behavior is disabled by default.

By default, the ASA sends only IP addresses in load-balancing redirection to a client. If certificates are in use that are based on DNS names, the certificates will be invalid when redirected to a backup device.

As a VPN cluster master, this ASA can send a fully qualified domain name (FQDN), using reverse DNS lookup, of a cluster device (another ASA in the cluster) instead of its outside IP address when redirecting VPN client connections to that cluster device.

All of the outside and inside network interfaces on the load-balancing devices in a cluster must be on the same IP network.

To do VPN load balancing for SSL or IPsec/IKEv2 connections using FQDNs rather than IP addresses, perform the following configuration steps:

- Step 1** Enable the use of FQDNs for load balancing with the **redirect-fqdn enable** command:

```
redirect-fqdn {enable | disable}
no redirect-fqdn {enable | disable}
```

For example:

```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# redirect-fqdn enable
```

```
hostname (config-load-balancing) #
```

- Step 2** Add an entry for each of your ASA outside interfaces into your DNS server if such entries are not already present. Each ASA outside IP address should have a DNS entry associated with it for lookups. These DNS entries must also be enabled for reverse lookup.
- Step 3** Enable DNS lookups on your ASA with the **dns domain-lookup inside** command on whichever interface has a route to your DNS server.
- Step 4** Define your DNS server IP address on the ASA; for example: **dns name-server 10.2.3.4** (IP address of your DNS server).

The following is an example of a VPN load balancing command sequence that includes an interface command that enables redirection for a fully qualified domain name, specifies the public interface of the cluster as **test** and the private interface of the cluster as **foo**:

```
hostname (config) # interface GigabitEthernet 0/1
hostname (config-if) # ip address 209.165.202.159 255.255.255.0
hostname (config) # nameif test
hostname (config) # interface GigabitEthernet 0/2
hostname (config-if) # ip address 209.165.201.30 255.255.255.0
hostname (config) # nameif foo
hostname (config) # vpn load-balancing
hostname (config-load-balancing) # nat 192.168.10.10
hostname (config-load-balancing) # priority 9
hostname (config-load-balancing) # interface lbpublic test
hostname (config-load-balancing) # interface lbprivate foo
hostname (config-load-balancing) # cluster ip address 209.165.202.224
hostname (config-load-balancing) # cluster key 123456789
hostname (config-load-balancing) # cluster encryption
hostname (config-load-balancing) # cluster port 9023
hostname (config-load-balancing) # redirect-fqdn enable
hostname (config-load-balancing) # participate
```

Frequently Asked Questions About Load Balancing

IP Address Pool Exhaustion

Q: Does the ASA consider IP address pool exhaustion as part of its VPN load-balancing method?

A: No. If the remote access VPN session is directed to a device that has exhausted its IP address pools, the session does not establish. The load-balancing algorithm is based on load, and is computed as an integer percentage (number of active and maximum sessions) that each backup cluster member supplies.

Unique IP Address Pools

Q: To implement VPN load balancing, must the IP address pools for AnyConnect clients or IPsec clients on different ASAs be unique?

A: Yes. IP address pools must be unique for each device.

Using Load Balancing and Failover on the Same Device

Q: Can a single device use both load balancing and failover?

A: Yes. In this configuration, the client connects to the IP address of the cluster and is redirected to the least-loaded ASA in the cluster. If that device fails, the standby unit takes over immediately, and there is no impact to the VPN tunnel.

Load Balancing on Multiple Interfaces

Q: If we enable SSL VPN on multiple interfaces, is it possible to implement load balancing for both of the interfaces?

A: You can define only one interface to participate in the cluster as the public interface. The idea is to balance the CPU loads. Multiple interfaces converge on the same CPU, so the concept of load balancing on multiple interfaces has no meaning.

Maximum Simultaneous Sessions for Load Balancing Clusters

Q: Consider a deployment of two ASA 5525-Xs, each with a 100-user SSL VPN license. In a load-balancing cluster, does the maximum total number of users allow 200 simultaneous sessions, or only 100? If we add a third device later with a 100-user license, can we now support 300 simultaneous sessions?

A: With VPN load balancing, all devices are active, so the maximum number of sessions that your cluster can support is the total of the number of sessions for each of the devices in the cluster, in this case 300.

Viewing Load Balancing

The load-balancing cluster master receives a periodic message from each ASA in the cluster with the number of active AnyConnect and clientless sessions, as well as the maximum allowed sessions based on the configured or license limits. If an ASA in the cluster shows 100 percent full capacity, the cluster master cannot redirect more connections to it. Although the ASA may show as full, some users may be in inactive/wait-to-resume state, wasting the licenses. As a workaround, each ASA provides the total number of sessions minus the sessions in inactive state, instead of the total number of sessions. (Refer to the **-sessiondb summary** command in the command reference. In other words, the inactive sessions are not reported to the cluster master. Even if the ASA is full (with some inactive sessions), the cluster master still redirects connections to it if necessary. When the ASA receives the new connection, the session that has been inactive the longest is logged off, allowing new connections to take its license.

The following example shows 100 SSL sessions (active only) and a 2 percent SSL load. These numbers do not include the inactive sessions. In other words, inactive sessions do not count towards the load for load balancing.

```
hostname# load-balancing
  Status :      enabled
  Role   :      Master
  Failover :    Active
  Encryption : enabled
  Cluster IP : 192.168.1.100
  Peers  :      1
```

				Load %			
Sessions				IPsec	SSL	IPsec	SSL
Public IP	Role	Pri	Model				
192.168.1.9	Master	7	ASA-5540	4	2	216	100
192.168.1.19	Backup	9	ASA-5520	0	0	0	0

Configuring VPN Session Limits

You can run as many IPsec and SSL VPN sessions as your platform and ASA license supports. To view the licensing information including maximum sessions for your ASA, enter the **show version** command in global configuration mode. The following example shows the command and the licensing information from the output of this command:

```
hostname(config)# show version

Cisco Adaptive Security Appliance Software Version 8.4(1)
Device Manager Version 6.4(1)

Compiled on Sun 02-Jan-11 03:45 by builders
System image file is "disk0:/cdisk.bin"
Config file at boot was "startup-config"
asa4 up 9 days 3 hours

Hardware:   ASA5510, 256 MB RAM, CPU Pentium 4 Celeron 1600 MHz
Internal ATA Compact Flash, 256MB
BIOS Flash M50FW080 @ 0xffff00000, 1024KB

Encryption hardware device : Cisco ASA-55x0 on-board accelerator (revision 0x0)
                          Boot microcode       : CN1000-MC-BOOT-2.00
                          SSL/IKE microcode    : CNLite-MC-SSLm-PLUS-2.03
                          IPsec microcode      : CNLite-MC-IPSECm-MAIN-2.06
                          Number of accelerators: 1

0: Ext: Ethernet0/0      : address is 001e.f75e.8b84, irq 9
1: Ext: Ethernet0/1      : address is 001e.f75e.8b85, irq 9
2: Ext: Ethernet0/2      : address is 001e.f75e.8b86, irq 9
3: Ext: Ethernet0/3      : address is 001e.f75e.8b87, irq 9
4: Ext: Management0/0    : address is 001e.f75e.8b83, irq 11
5: Int: Internal-Data0/0 : address is 0000.0001.0002, irq 11
6: Int: Internal-Control0/0 : address is 0000.0001.0001, irq 5

Licensed features for this platform:
Maximum Physical Interfaces      : Unlimited      perpetual
Maximum VLANs                   : 100           perpetual
Inside Hosts                     : Unlimited     perpetual
Failover                         : Active/Active perpetual
VPN-DES                          : Enabled       perpetual
VPN-3DES-AES                     : Enabled       perpetual
Security Contexts                : 2            perpetual
GTP/GPRS                        : Disabled     perpetual
AnyConnect Premium Peers        : 250          perpetual
AnyConnect Essentials           : Disabled     perpetual
Other VPN Peers                  : 250          perpetual
Total VPN Peers                  : 250          perpetual
Shared License                   : Disabled     perpetual
AnyConnect for Mobile           : Disabled     perpetual
AnyConnect for Cisco VPN Phone  : Disabled     perpetual
Advanced Endpoint Assessment     : Enabled       perpetual
UC Phone Proxy Sessions         : 2            perpetual
Total UC Proxy Sessions         : 2            perpetual
Botnet Traffic Filter           : Disabled     perpetual
Intercompany Media Engine       : Disabled     perpetual

This platform has an ASA 5510 Security Plus license.

hostname#
```

To limit AnyConnect VPN sessions (either IPsec/IKEv2 or SSL) to a lower value than the ASA allows, use the **vpn-sessiondb max-anyconnect-premium-or-essentials-limit** command in global configuration mode. To remove the session limit, use the **no** version of this command.

For example, if the ASA license allows 500 SSL VPN sessions, and you want to limit the number of AnyConnect VPN sessions to 250, enter the following command:

```
hostname(config)# vpn-sessiondb max-anyconnect-premium-or-essentials-limit 250
hostname(config)#
```

To remove the session limit, use the **no** version of this command.:

```
hostname(config)# no vpn-sessiondb max-anyconnect-premium-or-essentials-limit 250
hostname(config)#
```

To limit Cisco VPN client (IPsec IKEv1), Lan-to-Lan VPN, and clientless SSL VPN sessions to a lower value than the ASA allows, enter the **vpn-sessiondb max-other-vpn-limit** command in global configuration mode:

For example, if the ASA license allows 750 IPsec sessions, and you want to limit the number of IPsec sessions to 500, enter the following command:

```
hostname(config)# vpn-sessiondb max-other-vpn-limit 500
hostname(config)#
```

To remove the session limit, use the **no** version of this command:

```
hostname(config)# no vpn-sessiondb max-other-vpn-limit 500
hostname(config)#
```

Using an Identify Certificate When Negotiating

The ASA needs to use an identity certificate when negotiating the IKEv2 tunnel with AnyConnect clients. For `ikev2 remote-access trustpoint` configuration, use the following commands

```
crypto ikev2 remote-access trustpoint <name> [line<number>]
```

Using this command allows the AnyConnect client to support group selection for the end user. You can configure two trustpoints at the same time: two RSA, two ECDSA, or one of each. The ASA scans the configured trustpoint list and chooses the first one that the client supports. If ECDSA is preferred, you should configure that trustpoint before the RSA trustpoint.

The line number option specifies where in the line number you want the trustpoint inserted. Typically, this option is used to insert a trustpoint at the top without removing and re-adding the other line. If a line is not specified, the ASA adds the trustpoint at the end of the list.

If you try to add a trustpoint that already exists, you receive an error. If you use the `no crypto ikev2 remote-access trustpoint` command without specifying which trustpoint name to remove, all trustpoint configuration is removed.

Configuring the Pool of Cryptographic Cores

You can change the allocation of cryptographic cores on Symmetric Multi-Processing (SMP) platforms to give you better throughput performance for AnyConnect TLS/DTLS traffic. These changes can accelerate the SSL VPN datapath and provide customer-visible performance gains in AnyConnect, smart tunnels, and port forwarding. These steps describe configuring the pool of cryptographic cores in either single or multiple context mode:

**Note**

Multiple context mode only applies to IKEv2 and IKEv1 site to site but does not apply to AnyConnect, clientless SSL VPN, the legacy Cisco VPN client, the Apple native VPN client, the Microsoft native VPN client, or the cTCP for IKEv1 IPsec.

Limitations

- Cryptographic core rebalancing is available on the following platforms:
 - 5585-X
 - 5545-X
 - 5555-X
 - ASASM

Detailed Steps

	Command	Purpose
Step 1	<pre>asa1(config)# crypto engine ? asa1(config)# crypto engine accelerator-bias ?</pre>	Specifies how to allocate crypto accelerator processors: <ul style="list-style-type: none"> • balanced - Equally distribute crypto hardware resources • ipsec - Allocate crypto hardware resources to favor IPsec/Encrypted Voice (SRTP) • ssl - Allocate crypto hardware resources to favor SSL

Viewing Active VPN Sessions

Viewing Active AnyConnect Sessions by IP Address Type

To view active AnyConnect sessions using the command line interface, enter the **show vpn-sessiondb anyconnect filter p-ipversion** or **show vpn-sessiondb anyconnect filter a-ipversion** command in privileged EXEC mode.

Command	Purpose
<code>show vpn-sessiondb anyconnect filter p-ipversion {v4 v6}</code>	This command shows active AnyConnect sessions filtered by the endpoint's public IPv4 or IPv6 address. The public address is the address assigned to the endpoint by the enterprise.
<code>show vpn-sessiondb anyconnect filter a-ipversion {v4 v6}</code>	This command shows active AnyConnect sessions filtered by the endpoint's assigned IPv4 or IPv6 address. The assigned address is the address assigned to the AnyConnect Secure Mobility Client by the ASA.

Examples

Example 3-1 Output from `show vpn-sessiondb anyconnect filter p-ipversion [v4 / v6]` command

```
hostname(config)# show vpn-sessiondb anyconnect filter p-ipversion v4

Session Type: AnyConnect

Username       : user1                Index       : 40
Assigned IP    : 192.168.17.10        Public IP    : 198.51.100.1
Protocol       : AnyConnect-Parent SSL-Tunnel
License        : AnyConnect Premium
Encryption     : AnyConnect-Parent: (1)none  SSL-Tunnel: (1)RC4
Hashing        : AnyConnect-Parent: (1)none  SSL-Tunnel: (1)SHA1
Bytes Tx       : 10570                Bytes Rx     : 8085
Group Policy   : GroupPolicy_SSLACCLIENT
Tunnel Group   : SSLACCLIENT
Login Time     : 15:17:12 UTC Mon Oct 22 2012
Duration       : 0h:00m:09s
Inactivity     : 0h:00m:00s
NAC Result     : Unknown
VLAN Mapping   : N/A                    VLAN         : none
```

Example 3-2 Output from `show vpn-sessiondb anyconnect filter a-ipversion [v4 / v6]` command

```
hostname(config)# show vpn-sessiondb anyconnect filter a-ipversion v6

Session Type: AnyConnect

Username       : user1                Index       : 45
Assigned IP    : 192.168.17.10
Public IP      : 2001:DB8:8:1:90eb:3fe5:9eea:fb29
Assigned IPv6  : 2001:DB8:9:1::24
Protocol       : AnyConnect-Parent SSL-Tunnel
License        : AnyConnect Premium
Encryption     : AnyConnect-Parent: (1)none  SSL-Tunnel: (1)RC4
Hashing        : AnyConnect-Parent: (1)none  SSL-Tunnel: (1)SHA1
Bytes Tx       : 10662                Bytes Rx     : 17248
Group Policy   : GroupPolicy_SSL_IPv6      Tunnel Group : SSL_IPv6
Login Time     : 17:42:42 UTC Mon Oct 22 2012
Duration       : 0h:00m:33s
Inactivity     : 0h:00m:00s
NAC Result     : Unknown
VLAN Mapping   : N/A                    VLAN         : none
```

Viewing Active Clientless SSL VPN Sessions by IP Address Type

To view active clientless SSL VPN sessions using the command line interface, enter the **show vpn-sessiondb webvpn filter ipversion** command in privileged EXEC mode.

Command	Purpose
<code>show vpn-sessiondb webvpn filter ipversion {v4 v6}</code>	This command shows active clientless SSL VPN sessions filtered by the endpoint's public IPv4 or IPv6 address. The public address is the address assigned to the endpoint by the enterprise.

Examples

Example 3-3 Output from show vpn-sessiondb webvpn filter ipversion [v4 / v6] command

```
hostname# sh vpn-sessiondb webvpn filter ipversion v4

Session Type: WebVPN

Username      : user1                Index      : 63
Public IP    : 171.16.17.6
Protocol     : Clientless
License      : AnyConnect Premium
Encryption   : Clientless: (1)RC4   Hashing    : Clientless: (1)SHA1
Bytes Tx     : 62454                Bytes Rx   : 13082
Group Policy : SSLv6                Tunnel Group : SSL_IPv6
Login Time   : 18:07:48 UTC Mon Oct 22 2012
Duration     : 0h:00m:16s
Inactivity   : 0h:00m:00s
NAC Result   : Unknown
VLAN Mapping : N/A                VLAN       : none
```

Viewing Active Lan to Lan VPN Sessions by IP Address Type

To view active clientless SSL VPN sessions using the command line interface, enter the **show vpn-sessiondb l2l filter ipversion** command in privileged EXEC mode.

Command	Purpose
<code>show vpn-sessiondb l2l filter ipversion {v4 v6}</code>	This command shows active lan to lan VPN sessions filtered by the connection's public IPv4 or IPv6 address. The public address is the address assigned to the endpoint by the enterprise.

Configuring ISE Policy Enforcement

The Cisco Identity Services Engine (ISE) is a security policy management and control platform. It automates and simplifies access control and security compliance for wired, wireless, and VPN connectivity. Cisco ISE is primarily used to provide secure access and guest access, support BYOD initiatives, and enforce usage policies in conjunction with Cisco TrustSec.

The ISE Change of Authorization (CoA) feature provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is established. When a policy changes for a user or user group in AAA, CoA packets can be sent directly to the ASA from the ISE to reinitialize authentication and apply the new policy. An Inline Posture Enforcement Point (IPEP) is no longer required to apply access control lists (ACLs) for each VPN session established with the ASA.

ISE policy enforcement is supported on the following VPN clients:

- IPsec
- AnyConnect
- L2TP/IPsec

The system flow is as follows:

1. An end user requests a VPN connection.
2. The ASA authenticates the user to the ISE and receives a user ACL that provides limited access to the network.
3. An accounting start message is sent to the ISE to register the session.
4. Posture assessment occurs directly between the NAC agent and the ISE. This process is transparent to the ASA.
5. The ISE sends a policy update to the ASA via a CoA “policy push.” This identifies a new user ACL that provides increased network access privileges.

**Note**

Additional policy evaluations may occur during the lifetime of the connection, transparent to the ASA, via subsequent CoA updates.

Configuring RADIUS Server Groups

If you want to use an external RADIUS server for authentication, authorization, or accounting, you must first create at least one RADIUS server group per AAA protocol and add one or more servers to each group. You identify AAA server groups by name.

To add a RADIUS server group, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	<pre>aaa-server server_tag protocol radius</pre> <p>Example:</p> <pre>hostname(config)# aaa-server servergroup1 protocol radius hostname(config-aaa-server-group)#</pre>	<p>Identifies the server group name and the protocol.</p> <p>When you enter the aaa-server protocol command, you enter aaa-server group configuration mode.</p>
Step 2	<pre>merge-dacl {before-avpair after-avpair}</pre> <p>Example:</p> <pre>hostname(config)# aaa-server servergroup1 protocol radius hostname(config-aaa-server-group)# merge-dacl before-avpair</pre>	<p>Merges a downloadable ACL with the ACL received in the Cisco AV pair from a RADIUS packet. The default setting is no merge dacl, which specifies that downloadable ACLs will not be merged with Cisco AV pair ACLs. If both an AV pair and a downloadable ACL are received, the AV pair has priority and is used.</p> <p>The before-avpair option specifies that the downloadable ACL entries should be placed before the Cisco AV pair entries.</p> <p>The after-avpair option specifies that the downloadable ACL entries should be placed after the Cisco AV pair entries. This option applies only to VPN connections. For VPN users, ACLs can be in the form of Cisco AV pair ACLs, downloadable ACLs, and an ACL that is configured on the ASA. This option determines whether or not the downloadable ACL and the AV pair ACL are merged, and does not apply to any ACLs configured on the ASA.</p>

	Command	Purpose
Step 3	<p><code>max-failed-attempts number</code></p> <p>Example: <pre>hostname(config-aaa-server-group)# max-failed-attempts 2</pre></p>	<p>Specifies the maximum number of requests sent to a RADIUS server in the group before trying the next server. The <i>number</i> argument can range from 1 and 5. The default is 3.</p> <p>If you configured a fallback method using the local database (for management access only), and all the servers in the group fail to respond, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default), so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the reactivation-mode command in the next step.</p> <p>If you do not have a fallback method, the ASA continues to retry the servers in the group.</p>
Step 4	<p><code>reactivation-mode {depletion [deadtime minutes] timed}</code></p> <p>Example: <pre>hostname(config-aaa-server-group)# reactivation-mode deadtime 20</pre></p>	<p>Specifies the method (reactivation policy) by which failed servers in a group are reactivated.</p> <p>The depletion keyword reactivates failed servers only after all of the servers in the group are inactive.</p> <p>The deadtime minutes keyword-argument pair specifies the amount of time in minutes, between 0 and 1440, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes.</p> <p>The timed keyword reactivates failed servers after 30 seconds of down time.</p>
Step 5	<p><code>accounting-mode simultaneous</code></p> <p>Example: <pre>hostname(config-aaa-server-group)# accounting-mode simultaneous</pre></p>	<p>Sends accounting messages to all servers in the group.</p> <p>To restore the default of sending messages only to the active server, enter the accounting-mode single command.</p>
Step 6	<p><code>aaa-server server_group [interface_name] host server_ip</code></p> <p>Example: <pre>hostname(config)# aaa-server servergroup1 outside host 10.10.1.1</pre></p>	<p>Identifies the server and the AAA server group to which it belongs.</p> <p>When you enter the aaa-server host command, you enter aaa-server host configuration mode.</p>

	Command	Purpose
Step 7	<p><code>dynamic-authorization {port port-number}</code></p> <p>Example: <pre>(config-aaa-server-group)# dynamic-authorization port 1700</pre></p>	<p>Enables the RADIUS Dynamic Authorization (CoA) services for the AAA server group.</p> <p>Once defined, the corresponding RADIUS server group will be registered for CoA notification and the ASA will listen to the port for the CoA policy updates from ISE.</p> <p>The valid range of the CoA listening <i>port-number</i> is 1 to 65535.</p> <p>If the port number or interface specified in the ‘no’ form of this command does not match a line in the current configuration, an error message will be displayed.</p>
Step 8	<p><code>authorize-only</code></p> <p>Example: <pre>(config-aaa-server-group)# authorize-only</pre></p>	<p>Enables authorize-only mode for the RADIUS server group. This indicates that when this server group is used for authorization, the RADIUS Access Request message will be built as an “Authorize Only” request as opposed to the configured password methods that are available now. The Authorize-Only request includes a Service-Type attribute with value Authorize-Only (17) and message authenticator within the Access-Request.</p> <p>The support of the authorize-only mode eliminates the need of including the RADIUS common password in the Access-Request. Thus, it does not require the configuration of common password using the <code>radius-common-pw</code> CLI in the <code>aaa-server-host</code> mode.</p> <p> Note The authorize-only mode is configured for the server-group while the common password is host-specific. Thus, once authorize-only mode is configured, the common password configured for individual AAA server would be ignored.</p>
Step 9	<p><code>without-csd {anyconnect}</code></p> <p>Example: <pre>(config-tunnel-webvpn)# without-csd anyconnect</pre></p>	<p>Switches off hostscan processing for connections that are made to a specific tunnel-group. This setting currently applies to clientless and L3 connections. This command has been modified to allow this setting to be applied to AnyConnect connections only.</p>

	Command	Purpose
Step 10	<pre>interim-accounting-update {periodic interval}</pre> <p>Example:</p> <pre>(config-aaa-server-group)# interim-accounting-update periodic 12</pre>	<p>Enables the generation of RADIUS interim-accounting-update messages. Currently these messages are only generated when a VPN tunnel connection is added to a clientless VPN session. When this happens the accounting update is generated in order to inform the RADIUS server of the newly assigned IP address. Keywords have been added to this command to enable it to be configured to allow the current capabilities or to allow the generation of periodic interim accounting updates for all sessions that are configured to send accounting messages to the indicated server group.</p> <p><i>periodic</i> - This optional keyword enables the periodic generation and transmission of accounting records for every VPN session that is configured to send accounting records to the server group in question.</p> <p><i>interval</i> - This is a numeric value that represents the length, in hours, of the interval between periodic accounting updates. The valid range is 1 to 120 and the default value is 24.</p>

Example Configuration

The following example shows how to add one RADIUS group with a single server:

```
hostname(config)# aaa-server AuthOutbound protocol radius
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.3
hostname(config-aaa-server-host)# key RadUauthKey
hostname(config-aaa-server-host)# exit
```

The following example shows how to configure an ISE server object for authorization-only, dynamic authorization (CoA) updates, and hourly periodic accounting:

```
hostname(config)# aaa-server ise protocol radius
hostname(config-aaa-server-group)# authorize-only
hostname(config-aaa-server-group)# interim-accounting-update periodic 1
hostname(config-aaa-server-group)# dynamic-authorization
hostname(config-aaa-server-group)# exit
hostname(config-aaa-server-group)# authorize-only
hostname(config)# aaa-server ise (inside) host 10.1.1.3
hostname(config-aaa-server-host)# key sharedsecret
hostname(config-aaa-server-host)# exit
```

The following example shows how to configure a tunnel group for password authentication with ISE:

```
hostname(config)# tunnel-group aaa-coa general-attributes
hostname(config-tunnel-general)# address-pool vpn
hostname(config-tunnel-general)# authentication-server-group ise
hostname(config-tunnel-general)# accounting-server-group ise
hostname(config-tunnel-general)# exit
```

The following example shows how to configure a tunnel group for local certificate validation and authorization with ISE:

```
hostname(config)# tunnel-group aaa-coa general-attributes
```

```
hostname (config-tunnel-general) # address-pool vpn
hostname (config-tunnel-general) # authentication certificate
hostname (config-tunnel-general) # authorization-server-group ise
hostname (config-tunnel-general) # accounting-server-group ise
hostname (config-tunnel-general) # exit
```

For further details on how to enable CoA, see the “*Configuring RADIUS Servers for AAA*” chapter in the “*Cisco ASA Series General Operations CLI Configuration Guide*.”

Command Summary

Command	Purpose
<pre>(config-aaa-server-group)# dynamic-authorization {port port-number}</pre>	<p>Enables the RADIUS Dynamic Authorization (CoA) services for the AAA server group.</p> <p>Once defined, the corresponding RADIUS server group will be registered for CoA notification and the ASA will listen to the port for the CoA policy updates from ISE.</p> <p>The valid range of the CoA listening <i>port-number</i> is 1 to 65535.</p> <p>If the port number or interface specified in the ‘no’ form of this command does not match a line in the current configuration, an error message will be displayed.</p>
<pre>(config-aaa-server-group)# authorize-only</pre>	<p>Enables authorize-only mode for the RADIUS server group. This indicates that when this server group is used for authorization, the RADIUS Access Request message will be built as an “Authorize Only” request as opposed to the configured password methods that are available now. The Authorize-Only request includes a Service-Type attribute with value Authorize-Only (17) and message authenticator within the Access-Request.</p> <p>The support of the authorize-only mode eliminates the need of including the RADIUS common password in the Access-Request. Thus, it does not require the configuration of common password using the radius-common-pw CLI in the aaa-server-host mode.</p> <p> Note The authorize-only mode is configured for the server-group while the common password is host-specific. Thus, once authorize-only mode is configured, the common password configured for individual AAA server would be ignored.</p>

Command	Purpose
<code>(config-tunnel-webvpn)# without-csd {anyconnect}</code>	Switches off hostscan processing for connections that are made to a specific tunnel-group. This setting currently applies to clientless and L3 connections. This command has been modified to allow this setting to be applied to AnyConnect connections only.
<code>(config-aaa-server-group)# interim-accounting-update {periodic interval}</code>	<p>Enables the generation of RADIUS interim-accounting-update messages. Currently these messages are only generated when a VPN tunnel connection is added to a clientless VPN session. When this happens the accounting update is generated in order to inform the RADIUS server of the newly assigned IP address. Keywords have been added to this command to enable it to be configured to allow the current capabilities or to allow the generation of periodic interim accounting updates for all sessions that are configured to send accounting messages to the indicated server group.</p> <p><i>periodic</i> - This optional keyword enables the periodic generation and transmission of accounting records for every VPN session that is configured to send accounting records to the server group in question.</p> <p><i>interval</i> - This is a numeric value that represents the length, in hours, of the interval between periodic accounting updates. The valid range is 1 to 120 and the default value is 24.</p>

Troubleshooting

The following commands can be used for debugging.

To trace CoA activity:

```
debug radius dynamic-authorization
```

To trace redirect URL functionality:

```
debug aaa url-redirect
```

To view NP classification rules corresponding to URL redirect functionality:

```
show asp table classify domain url-redirect
```

