



CHAPTER 62

Setting General VPN Parameters

The adaptive security appliance 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:

- [Configuring VPNs in Single, Routed Mode, page 62-1](#)
- [Configuring IPsec to Bypass ACLs, page 62-1](#)
- [Permitting Intra-Interface Traffic \(Hairpinning\), page 62-2](#)
- [Setting Maximum Active IPsec or SSL VPN Sessions, page 62-4](#)
- [Using Client Update to Ensure Acceptable IPsec Client Revision Levels, page 62-4](#)
- [Understanding Load Balancing, page 62-6](#)
- [Configuring Load Balancing, page 62-11](#)
- [Configuring VPN Session Limits, page 62-16](#)



Note

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 VPNs in Single, Routed Mode

VPNs work only in single, routed mode. VPN functionality is unavailable in configurations that include either security contexts, also referred to as multi-mode firewall, or Active/Active stateful failover.

The exception to this caveat is that you can configure and use one connection for administrative purposes to (not through) the adaptive security appliance in transparent mode.

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 adaptive security appliance and want to maximize the adaptive security appliance performance. Typically, you create an ACL that permits IPsec packets 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 adaptive security appliance.

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

The following example enables IPsec traffic through the adaptive security appliance 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" access-list, 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 adaptive security appliance 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.

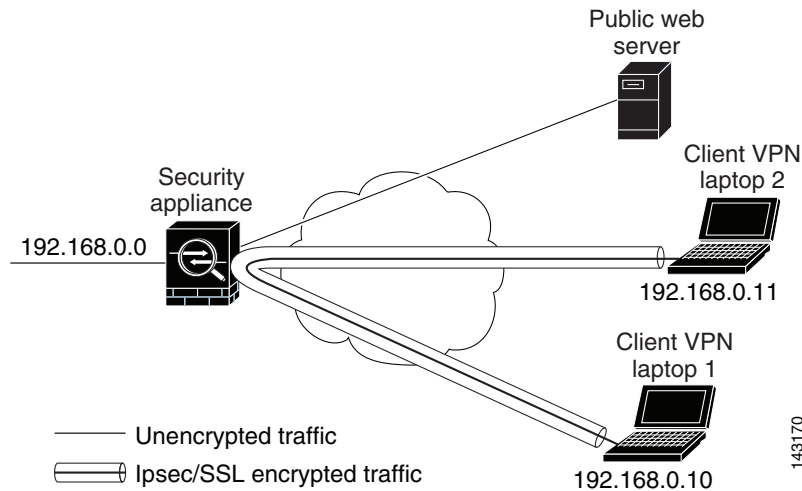
The **ssh** and **http** commands are of a higher priority than the ACLs. In other words, to deny ssh, telnet, or ICMP traffic to the box from the VPN session, use ssh, telnet and icmp commands, which denies the IP local pool should be added.

Permitting Intra-Interface Traffic (Hairpinning)

The adaptive security appliance 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 (adaptive security appliance).

In another application, this feature 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 62-1](#) shows VPN Client 1 sending secure IPsec traffic to VPN Client 2 while also sending unencrypted traffic to a public Web server.

Figure 62-1 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 adaptive security appliance interface, as discussed in the following section.

NAT Considerations for Intra-Interface Traffic

For the adaptive security appliance 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 adaptive security appliance 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 adaptive security appliance allows, enter the **vpn-sessiondb max-session-limit** command in global configuration mode.

- This command applies to all types of VPN sessions, including SSL VPN.
- This limit affects the calculated load percentage for VPN Load Balancing.

The syntax is **vpn-sessiondb max-session-limit** *{session-limit}*.

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

```
hostname (config)# vpn-sessiondb max-session-limit 450
hostname (config)#
```

The next example shows how to set both SSL VPN client and clientless max sessions limit:

```
hostname (config)# vpn-sessiondb max-webvpn-session-limit {session-limit}.
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 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 tells how to perform a client-update:

Step 1 In global configuration mode, enable client update by entering the 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 adaptive security appliance.

The syntax of the command to do this is:

```
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, it configures client update parameters for the IPsec remote-access tunnel-group "salesgrp". It 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 To define a set of client-update parameters for a particular ipsec-ra tunnel group, do the following. 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:

```
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** Optionally, you can 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, you would 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.

Understanding Load Balancing

If you have a remote-access configuration in which you are using two or more adaptive security appliances 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, thus distributing 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. It belongs to the current virtual cluster master; hence, it is 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 adaptive security appliance 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. Should the virtual cluster master itself fail, 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 of which is the virtual master, and the others 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 adaptive security appliances 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 adaptive security appliance supports two failover configurations, Active/Active failover and Active/Standby failover. VPN connections run only in Active/Standby, single routed mode. Active/Active failover requires multi-context mode, so does not support VPN connections.

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 adaptive security appliance 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 adaptive security appliance checks for the existence of this crypto license before enabling load balancing. If it does not detect an active 3DES or AES license, the adaptive security appliance 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 adaptive security appliance models ASA 5510 (with a Plus license) and Model 5520 and above. You can also include 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 adaptive security appliance (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
- 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 adaptive security appliance 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/SSL VPN tunnel to the device with the lowest load until it is 1% 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%.

1. The master device redirects tunnels to the first backup device (the one with the lowest inside IP address) until it reaches 1%.
2. The master device then redirects tunnels to the backup secondary device (the one with the highest inside IP address) until it also reaches 1%.
3. The master device redirects tunnels to itself only when the two backup devices both reach 1% load.
4. The cycle repeats when all three devices reach 1% load.

VPN Load-Balancing Cluster Configurations

A load-balancing cluster can consist of adaptive security appliances 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 adaptive security appliances, 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 adaptive security appliances 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 adaptive security appliances or same release adaptive security appliances and VPN 3000 concentrators or both can support only IPsec sessions. In such a configuration, however, the adaptive security appliances 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 represents a departure from the load balancing calculation for the ASA Release 7.0(x) software and the VPN 3000 concentrator, in that these platforms both use a weighting algorithm that, on some hardware platforms, calculates SSL VPN session load differently from IPsec session load.

The virtual master of the cluster assigns session requests to the members of the cluster. The adaptive security appliance 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 adaptive security appliance 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 adaptive security appliances 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 adaptive security appliances and VPN 3000 Concentrators. Some of the adaptive security appliance 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 adaptive security appliance 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, a adaptive security appliance running ASA Release 7.1(1) software is the initial cluster master; 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) adaptive security appliance. The following scenario illustrates this dilemma.

This scenario is similar to the previous one, in that the cluster consists of a mixture of adaptive security appliances and VPN 3000 Concentrators. Some of the adaptive security appliance 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 innately 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 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 adaptive security appliance 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 adaptive security appliance 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 start-up 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:

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

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

```
hostname(config-load-balancing)# nat 192.168.30.3
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 of the cluster to which this device belongs. This command specifies the single IP address that represents the entire virtual cluster. Choose an IP address that is within the public subnet address range shared by all the adaptive security appliances in the virtual cluster

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

For example, to set the cluster IP address to 192.168.10.10, enter the following command:

```
hostname(config-load-balancing)# cluster ip address 192.168.10.10
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 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** Optionally, 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 adaptive security appliances 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, select 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 to 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 adaptive security appliance can send a fully qualified domain name (FQDN), using reverse DNS lookup, of a cluster device (another adaptive security appliance 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 SSL VPN load Balancing 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 command - “dns domain-lookup inside” (or 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 adaptive security appliance consider IP address pool exhaustion as part of its VPN load balancing mechanism?

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/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 adaptive security appliances 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 adaptive security appliance 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 5520s, each with a 100-user SSL VPN license. In a load balancing cluster, does the maximum total number of users allow 200 simultaneous session, 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% 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 **show vpn-sessiondb summary** in the *Cisco ASA 5500 Series 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% SSL load. These numbers do not include the inactive sessions. In other words, inactive sessions do not count towards the load for load balancing.

```
nmeke-asa2# sh vpn load-balancing
  Status :      enabled
  Role   :      Master
  Failover :    Active
  Encryption :  enabled
  Cluster IP :  192.168.1.100
  Peers  :      1
```

				Load %			
Sessions							
Public IP	Role	Pri	Model	IPsec	SSL	IPsec	SSL
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 license for the adaptive security appliance supports. To view the licensing information for your adaptive security appliance, enter the **show version** command in global configuration mode. The following example shows the command and the licensing information excerpted from the output of this command:

```
hostname(config)# show version
```

```
Cisco Adaptive Security Appliance Software Version 7.1(0)182
Device Manager Version 5.1(0)128
```

```
Licensed features for this platform:
Maximum Physical Interfaces : Unlimited
Maximum VLANs              : 100
Inside Hosts                : Unlimited
Failover                    : Active/Active
VPN-DES                     : Enabled
VPN-3DES-AES                : Enabled
Security Contexts           : 10
GTP/GPRS                    : Enabled
VPN Peers                   : 750
WebVPN Peers                : 500
```

```
This platform has an ASA 5520 VPN Plus license.
```

To limit the maximum number of active IPsec VPN sessions to a lower value than the adaptive security appliance allows, enter the **vpn-sessiondb max-session-limit** command in global configuration mode. This limit affects the calculated load percentage for VPN Load Balancing.

```
hostname(config)# vpn-sessiondb max-session-limit number_of_sessions
hostname(config)#
```

For example, if the adaptive security appliance 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-session-limit 500
hostname(config)#
```

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

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

To limit SSL VPN sessions to a lower value than the adaptive security appliance allows, use the **vpn-sessiondb max-webvpn-session-limit** command in global configuration mode. To remove the session limit, use the **no** version of this command.

```
hostname(config)# vpn-sessiondb max-webvpn-session-limit number_of_sessions
hostname(config)#
```

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

```
hostname(config)# vpn-sessiondb max-webvpn-session-limit 250
hostname(config)#
```

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

```
hostname(config)# no vpn-sessiondb max-webvpn-session-limit
```



```
hostname(config)#
```

For a complete description of the features available with each license, see Appendix A, Feature Licenses and Specifications.

