



Configuring Tunnel Groups, Group Policies, and Users

This chapter describes how to configure VPN tunnel groups, group policies, and users. This chapter includes the following sections.

- [Overview of Tunnel Groups, Group Policies, and Users, page 25-1](#)
- [Configuring Tunnel Groups, page 25-4](#)
- [Group Policies, page 25-10](#)
- [Configuring Users, page 25-31](#)

In summary, you first configure tunnel groups to set the values for the connection. Then you configure group policies. These set values for users in the aggregate. Then you configure users, which can inherit values from groups and configure certain values on an individual user basis. This chapter describes how and why to configure these entities.

Overview of Tunnel Groups, Group Policies, and Users

Groups and users are core concepts in managing the security of virtual private networks (VPNs) and in configuring the security appliance. They specify attributes that determine user access to and use of the VPN. A *group* is a collection of users treated as a single entity. *Users* get their attributes from *group policies*. *Tunnel groups* identify the group policy for a specific connection. If you do not assign a particular group policy to a user, the default group policy for the connection applies.

Tunnel groups and group policies simplify system management. To streamline the configuration task, the security appliance provides a default LAN-to-LAN tunnel group (DefaultL2Lgroup), a default remote access tunnel group (DefaultRAGroup), and a default group policy (DfltGrpPolicy). The default tunnel groups and group policy provide settings that are likely to be common for many users. As you add users, you can specify that they “inherit” parameters from a group policy. Thus you can quickly configure VPN access for large numbers of users.

If you decide to grant identical rights to all VPN users, then you do not need to configure specific tunnel groups or group policies, but VPNs seldom work that way. For example, you might allow a finance group to access one part of a private network, a customer support group to access another part, and an MIS group to access other parts. In addition, you might allow specific users within MIS to access systems that other MIS users cannot access. Tunnel groups and group policies provide the flexibility to do so securely.

**Note**

The security appliance also includes the concept of object groups, which are a superset of network lists. Object groups let you define VPN access to ports as well as networks. Object groups relate to ACLs rather than to group policies and tunnel groups. For more information about using object groups, see [Chapter 13, “Identifying Traffic with Access Lists.”](#)

Tunnel Groups

A tunnel group consists of a set of records that contain tunnel connection policies. Tunnel groups contain a small number of attributes that pertain to creating the tunnel itself. Tunnel groups include a pointer to a group policy that defines user-oriented attributes.

The security appliance provides two default tunnel groups, one for LAN-to-LAN connections, and one for remote access connections. You can modify these default tunnel groups, but you cannot delete them. You can also create one or more tunnel groups specific to your environment. Tunnel groups are local to the security appliance and are not configurable on external servers.

Tunnel groups specify the following attributes:

- General parameters
- IPsec connection parameters

General Tunnel Group Parameters

The general parameters include the following:

- Tunnel group name—Both remote access and LAN-to-LAN clients select a tunnel group by its name, as follows:
 - For IPsec clients that use preshared keys to authenticate, the tunnel group name is the same as the group name that the IPsec client passes to the security appliance.
 - IPsec clients that use certificates to authenticate pass this name as part of the certificate, and the security appliance extracts the name from the certificate.

Tunnel group records contain tunnel connection policy information. These records identify the servers to which the tunnel user is authenticated, as well as the accounting servers to which connection information is sent. They also identify a default group policy for the connection, and they contain protocol-specific connection parameters.

- Connection type—Connection types include remote access IPsec, and LAN-to-LAN IPsec. A tunnel group can have only one connection type.
- Authentication, Authorization, and Accounting servers—These parameters identify the server groups or lists that the security appliance uses for the following purposes:
 - Authenticating users
 - Obtaining information about services users are authorized to access
 - Storing accounting records

A server group can consist of one or more servers.

- Default group policy for the connection—A group policy is a set of user-oriented attributes. The default group policy is the group policy whose attributes the security appliance uses as defaults when authenticating or authorizing a tunnel user.
- Client address assignment method—This method includes values for one or more DHCP servers or address pools that the security appliance assigns to clients.

IPSec Connection Parameters

IPSec parameters include the following:

- A client authentication method: preshared keys or certificates.
- ISAKMP keepalive settings. This feature lets the security appliance monitor the continued presence of a remote peer and report its own presence to that peer. If the peer becomes unresponsive, the security appliance removes the connection. Enabling IKE keepalives prevents hung connections when the IKE peer loses connectivity.

There are various forms of IKE keepalives. For this feature to work, both the security appliance and its remote peer must support a common form. This feature works with the following peers:

- Cisco VPN client (Release 3.0 and above)
- Cisco VPN 3000 Client (Release 2.x)
- Cisco VPN 3002 Hardware Client
- Cisco VPN 3000 Series Concentrators
- Cisco IOS software
- Cisco Secure PIX Firewall

Non-Cisco VPN clients do not support IKE keepalives.

If you are configuring a group of mixed peers, and some of those peers support IKE keepalives and others do not, enable IKE keepalives for the entire group. The feature does not affect the peers that do not support it.

If you disable IKE keepalives, connections with unresponsive peers remain active until they time out, so we recommend that you keep your idle timeout short. To change your idle timeout, see [“Configuring Group Policies” section on page 25-12](#).



Note

To reduce connectivity costs, disable IKE keepalives if this group includes any clients connecting via ISDN lines. ISDN connections normally disconnect if idle, but the IKE keepalives mechanism prevents connections from idling and therefore from disconnecting.

If you do disable IKE keepalives, the client disconnects only when either its IKE or IPSec keys expire. Failed traffic does not disconnect the tunnel with the Peer Timeout Profile values as it does when IKE keepalives are enabled.



Note

If you have a LAN-to-LAN configuration using IKE main mode, make sure that the two peers have the same IKE keepalives configuration. Both peers must have IKE keepalives enabled or both peers must have it disabled.

- Values for defining authorization usernames.

Configuring Tunnel Groups

The security appliance provides two default tunnel groups, one for remote access (DefaultRAGroup) and one for LAN-to-LAN (DefaultL2LGroup). You can modify these groups, but you cannot delete them. To see the current configured and default configuration of all your tunnel groups, including the default tunnel group, enter the **show running-config all tunnel-group** command.

You can configure a new tunnel group as either an IPsec Remote Access (ipsec-ra) tunnel or an IPsec LAN-to-LAN (ipsec-l2l) tunnel. The default is ipsec-ra. The subsequent parameters depend upon your choice of tunnel type.

Default Remote Access Tunnel Group Configuration

The contents of the default remote-access tunnel group are as follows:

```
tunnel-group DefaultRAGroup type ipsec-ra
tunnel-group DefaultRAGroup general-attributes
  no address-pool
  authentication-server-group LOCAL
  no authorization-server-group
  no accounting-server-group
  default-group-policy DfltGrpPolicy
  no dhcp-server
  no strip-realm
  no strip-group
tunnel-group DefaultRAGroup ipsec-attributes
  no pre-shared-key
  no authorization-required
  authorization-dn-attributes CN OU
  peer-id-validate req
  no radius-with-expiry
  no chain
  no trust-point
  isakmp keepalive threshold 300 retry 2
```

Configuring Remote-Access Tunnel Groups

To configure a remote-access tunnel group, follow the steps in this section. An IPsec Remote Access VPN tunnel group applies only to remote-access IPsec client connections.

Specify a Name and Type for the Remote-Access Tunnel Group

To assign a name and type for the tunnel group, enter the **tunnel-group** command to assign a name and type for the tunnel group.

```
hostname(config)# tunnel-group tunnel_group_name type tunnel_type
```

For a remote-access tunnel, the type is **ipsec-ra**; for example:

```
hostname(config)# tunnel-group TunnelGroup1 type ipsec-ra
```

Configure Remote-Access Tunnel Group General Attributes

To configure the tunnel group general attributes, specify the parameters in the following steps.

- Step 1** Enter the config-general mode by specifying the **tunnel-group** command with the general-attributes designator:

```
hostname(config)# tunnel-group tunnel_group_name general-attributes
```

This command enters config-general mode, in which you configure the tunnel-group general attributes.

- Step 2** Specify the name of the authentication-server group, if any, to use. If you want to use the LOCAL database for authentication if the specified server group fails, append the word LOCAL:

```
hostname(config-general)# authentication-server-group groupname [LOCAL]
```

You can also configure interface-specific authentication by including the name of an interface after the group name. The following command configures interface-specific authentication for the interface named “test” using the server “servergroup1” for authentication:

```
hostname(config-general)# authentication-server-group test servergroup1
```

- Step 3** Specify the name of the authorization-server group, if any, to use:

```
hostname(config-general)# authorization-server-group groupname
```

- Step 4** Specify the name of the accounting-server group, if any, to use:

```
hostname(config-general)# accounting-server-group groupname
```

- Step 5** Specify the name of the default group policy:

```
hostname(config-general)# default-group-policy policyname
```

The following example sets “DfltGrpPolicy” as the name of the group policy:

```
hostname(config)# default-group-policy DfltGrpPolicy
```

- Step 6** Specify the name or IP address of the DHCP server (up to 10 servers), and the names of the DHCP address pools (up to 6 pools). The defaults are no DHCP server and no address pool.

```
hostname(config-general)# dhcp-server server1 [...server10]
hostname(config-general)# address-pool [(interface name)] address_pool1 [...address_pool6]
```



Note The interface name must be enclosed in parentheses.

You configure address pools with the **ip local pool** command in global configuration mode.

- Step 7** Specify whether to strip the group or the realm from the username before passing it on to the AAA server. The default is not to strip either the group name or the realm.

```
hostname(config-general)# strip-group
hostname(config-general)# strip-realm
```

Enter the **strip-realm** command to remove the realm qualifier of the username during authentication. If you do so, authentication is based on the *username* alone. Otherwise, authentication is based on the full *username@realm* string. You must enable strip realm if your server is unable to parse delimiters. If you are using the Group Lookup feature and strip realm, do not use the @ character for the group delimiter.

- Step 8** Whether users must exist in the authorization database to connect.

```
hostname(config)# authorization-server-group groupname
```

Configure Remote-Access Tunnel Group IPsec Attributes

To configure the IPsec attributes, specify the following parameters:

- Step 1** Specify the IPsec-attributes designator:

```
hostname(config)# tunnel-group tunnel-group-name ipsec-attributes
```

For example, the following command designates that the config-ipsec mode commands that follow pertain to the tunnel group named “TG1”:

```
hostname(config)# tunnel-group TG1 ipsec-attributes
```

This command enters config-ipsec mode, in which you configure the tunnel-group IPsec attributes.

- Step 2** Specify the attribute or attributes to use in deriving a name for an authorization query from a certificate. This attribute specifies what part of the subject DN field to use as the username for authorization:

```
hostname(config-ipsec)# authorization-dn-attributes {primary-attribute  
[secondary-attribute] | use-entire-name}
```

For example, the following command specifies the use of the “CN” attribute as the username for authorization:

```
hostname(config-ipsec)# authorization-dn-attributes CN
```

The authorization-dn-attributes are **C** (Country), **CN** (Common Name), **DNQ** (DN qualifier), **EA** (E-mail Address), **GENQ** (Generational qualifier), **GN** (Given Name), **I** (Initials), **L** (Locality), **N** (Name), **O** (Organization), **OU** (Organizational Unit), **SER** (Serial Number), **SN** (Surname), **SP** (State/Province), **T** (Title), and **UID** (User ID)

- Step 3** Specify whether to require a successful authorization before allowing a user to connect. The default is not to require authorization.

```
hostname(config-ipsec)# authorization-required
```

- Step 4** Specify the client-update parameters; that is, the client type and the acceptable revision levels for that client:

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

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** (VPN3002 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 four of these client update entries.

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

- Step 5** Specify the preshared key to support IKE connections based on preshared keys.

```
hostname(config-ipsec)# pre-shared-key xyzx
```

The preceding command specifies the preshared key *xyzx* to support IKE connections for an IPSec remote access tunnel group:

- Step 6** Specify whether to validate the identity of the peer using the peer's certificate:

```
hostname(config-ipsec)# peer-id-validate option
```

The available options are **req** (required), **cert** (if supported by certificate), and **nocheck** (do not check). The default is **req**.

- Step 7** Specify whether to enable sending of a certificate chain. The following command includes the root certificate and any subordinate CA certificates in the transmission:

```
hostname(config-ipsec)# chain
```

You can apply this attribute to all tunnel-group types.

- Step 8** Specify the name of a trustpoint that identifies the certificate to be sent to the IKE peer:

```
hostname(config-ipsec)# trust-point trust-point-name
```

The following command specifies "mytrustpoint" as the name of the certificate to be sent to the IKE peer:

```
hostname(config-ipsec)# trust-point mytrustpoint
```

You can apply this attribute to all tunnel-group types.

- Step 9** Specify whether to have the security appliance use MS-CHAPv2 to negotiate a password update with the user during authentication:

```
hostname(config-ipsec)# radius-with-expiry
```

The security appliance ignores this command if RADIUS authentication has not been configured.

- Step 10** ISAKMP keepalive threshold and the number of retries allowed.

```
hostname(config)# isakmp keepalive threshold <number> retry <number>
```

The **threshold** parameter specifies the number of seconds (10 through 3600) that the peer is allowed to idle before beginning keepalive monitoring. The **retry** parameter is the interval (2 through 10 seconds) between retries after a keepalive response has not been received. IKE keepalives are enabled by default. To disable IKE keepalives, enter the **no** form of the **isakmp** command:

For example, the following command sets the IKE keepalive threshold value to 15 seconds and sets the retry interval to 10 seconds:

```
hostname(config-ipsec)# isakmp keepalive threshold 15 retry 10
```

The default value for the **threshold** parameter is 300 for remote-access and 10 for LAN-to-LAN, and the default value for the **retry** parameter is 2.

Default LAN-to-LAN Tunnel Group Configuration

The contents of the default LAN-to-LAN tunnel group are as follows:

```
tunnel-group DefaultL2LGroup type ipsec-l2l
tunnel-group DefaultL2LGroup general-attributes
  no accounting-server-group
  default-group-policy DfltGrpPolicy
tunnel-group DefaultL2LGroup ipsec-attributes
  no pre-shared-key
  peer-id-validate req
  no chain
  no trust-point
  isakmp keepalive threshold 10 retry 2
```

LAN-to-LAN tunnel groups have fewer parameters than remote-access tunnel groups, and most of these are the same for both groups. For your convenience in configuring the connection, they are listed separately here.

Configuring LAN-to-LAN Tunnel Groups

An IPsec LAN-to-LAN VPN tunnel group applies only to LAN-to-LAN IPsec client connections. To configure a LAN-to-LAN tunnel group, follow the steps in this section.

Specify a Name and Type for the LAN-to-LAN Tunnel Group

To specify a name and a type for a tunnel group, enter the **tunnel-group** command, as follows:

```
hostname(config)# tunnel-group tunnel_group_name type tunnel_type
```

For a LAN-to-LAN tunnel, the type is **ipsec-l2l**.; for example:

```
hostname(config)# tunnel-group TunnelGroup1 type ipsec-l2l
```

Configure LAN-to-LAN Tunnel Group General Attributes

To configure the tunnel group general attributes, specify the parameters in the following steps:

-
- Step 1** Enter configuration-general mode by specifying the general-attributes designator:

```
hostname(config)# tunnel-group tunnel_group_tunnel-group-name general-attributes
hostname(config-general)#
```

The prompt changes to indicate that you are now in config-general mode, in which you configure the tunnel-group general attributes.

- Step 2** Specify the name of the accounting-server group, if any, to use:

```
hostname(config-general)# accounting-server-group groupname
```

For example, the following command specifies the use of the accounting-server group “acctgserv1”:

```
hostname(config-general)# accounting-server-group acctgserv1
```

Step 3 Specify the name of the default group policy:

```
hostname(config-general)# default-group-policy policyname
```

For example, the following command specifies that the name of the default group policy is “MyPolicy”:

```
hostname(config-general)# default-group-policy MyPolicy
```

Configure LAN-to-LAN IPsec Attributes

To configure the IPsec attributes, do the following steps:

Step 1 To enter config-ipsec mode, in which you configure the tunnel-group IPsec attributes, enter the tunnel-group command with the IPsec-attributes designator.

```
hostname(config)# tunnel-group tunnel-group-name ipsec-attributes
```

For example, the following command enters config-ipsec mode so you can configure the parameters for the tunnel group named “TG1”:

```
hostname(config)# tunnel-group TG1 ipsec-attributes
hostname(config-ipsec)#
```

The prompt changes to indicate that you are now in config-ipsec mode.

Step 2 Specify the preshared key to support IKE connections based on preshared keys.

```
hostname(config-ipsec)# pre-shared-key key
```

For example, the following command specifies the preshared key XYZX to support IKE connections for an IPsec remote access tunnel group:

```
hostname(config-ipsec)# pre-shared-key xyzx
```

Step 3 Specify whether to validate the identity of the peer using the peer’s certificate:

```
hostname(config-ipsec)# peer-id-validate option
```

The available options are **req** (required), **cert** (if supported by certificate), and **nocheck** (do not check). The default is **req**. For example, the following command sets the peer-id-validate option to **nocheck**:

```
hostname(config-ipsec)# peer-id-validate nocheck
```

Step 4 Specify whether to enable sending of a certificate chain. This action includes the root certificate and any subordinate CA certificates in the transmission:

```
hostname(config-ipsec)# chain
```

You can apply this attribute to all tunnel-group types.

Step 5 Specify the name of a trustpoint that identifies the certificate to be sent to the IKE peer:

```
hostname(config-ipsec)# trust-point trust-point-name
```

For example, the following command sets the trustpoint name to “mytrustpoint”:

```
hostname(config-ipsec)# trust-point mytrustpoint
```

You can apply this attribute to all tunnel-group types.

- Step 6** Specify the ISAKMP keepalive threshold and the number of retries allowed. The **threshold** parameter specifies the number of seconds (10 through 3600) that the peer is allowed to idle before beginning keepalive monitoring. The **retry** parameter is the interval (2 through 10 seconds) between retries after a keepalive response has not been received. IKE keepalives are enabled by default. To disable IKE keepalives, enter the **no** form of the **isakmp** command:

```
hostname(config)# isakmp keepalive threshold <number> retry <number>
```

For example, the following command sets the ISAKMP keepalive threshold to 15 seconds and sets the retry interval to 10 seconds.:

```
hostname(config-ipsec)# isakmp keepalive threshold 15 retry 10
```

The default value for the **threshold** parameter for LAN-to-LAN is 10, and the default value for the retry parameter is 2.

Group Policies

A group policy is a set of user-oriented attribute/value pairs for IPsec connections that are stored either internally (locally) on the device or externally on a RADIUS server. The tunnel group refers to a group policy that sets terms for user connections after the tunnel is established. Group policies let you apply whole sets of attributes to a user or a group of users, rather than having to specify each attribute individually for each user.

Enter the **group-policy** commands in global configuration mode to assign a group policy to users or to modify a group policy for specific users.

The security appliance includes a default group policy. You can modify this default group policy, but you cannot delete it. You can also create one or more group policies specific to your environment.

Group policies include the following attributes:

- Identity
- Defining servers
- Client firewall settings
- Tunneling protocols
- IPsec settings
- Hardware client settings
- Filters
- Client configuration settings
- WebVPN functions
- Connection settings

Default Group Policy

The security appliance supplies a default group policy. You can modify this default group policy, but you cannot delete it. A default group policy, named “DfltGrpPolicy”, always exists on the security appliance, but this default group policy does not take effect unless you configure the security appliance to use it. To view the default group policy, enter the following command:

```
hostname(config)# show running-config all group-policy DfltGrpPolicy
```

To configure the default group policy, enter the following command:

```
hostname(config)# group-policy DfltGrpPolicy internal
```



Note

The default group policy is internal. Despite the fact that the command syntax is `hostname(config)# group-policy DfltGrpPolicy {internal | external}`, you cannot change the type to external.

If you want to change any of the attributes of the group policy, use the `group-policy attributes` command to enter attributes mode, then specify the commands to change whatever attributes that you want to modify:

```
hostname(config)# group-policy DfltGrpPolicy attributes
```



Note

The attributes mode applies only for internal group policies.

The default group policy that the security appliance provides, “DfltGrpPolicy”, is as follows:

```
group-policy DfltGrpPolicy internal
group-policy DfltGrpPolicy attributes
wins-server none
dns-server none
vpn-access-hours none
vpn-simultaneous-logins 3
vpn-idle-timeout 30
vpn-session-timeout none
vpn-filter none
vpn-tunnel-protocol IPSec
password-storage disable
ip-comp disable
re-xauth disable
group-lock none
pfs disable
banner none
ipsec-udp disable
ipsec-udp-port 10000
split-tunnel-policy tunnelall
split-tunnel-network-list none
default-domain none
split-dns none
secure-unit-authentication disable
user-authentication disable
user-authentication-idle-timeout 30
ip-phone-bypass disable
leap-bypass disable
nem disable
backup-servers keep-client-config
client-firewall none
client-access-rule none
```

```
webvpn
  functions url-entry
  no html-content-filter
  no homepage
  no filter
  no url-list
  no port-forward
  port-forward-name value Application Access
```

You can modify the default group policy, and you can also create one or more group policies specific to your environment.

Configuring Group Policies

A group policy can apply to either remote-access or LAN-to-LAN IPsec tunnels. In each case, if you do not explicitly define a parameter, the group takes the value from the default group policy. To configure a group policy, follow these steps:

Step 1 Specify a name and type (internal or external) for the group policy:

```
hostname(config)# group-policy group_policy_name type
```

For example, the following command specifies that the group policy is named “GroupPolicy1” and that its type is internal:

```
hostname(config)# group-policy GroupPolicy1 internal
```

The default type is **internal**.

You can initialize the attributes of an internal group policy to the values of a preexisting group policy by appending the keyword **from** and specifying the name of the existing policy:

```
hostname(config)# group-policy group_policy_name internal from group_policy_name
```

For an external group policy, you must identify the AAA server group that the security appliance can query for attributes and specify the password to use when retrieving attributes from the external AAA server group, as follows:

```
hostname(config)# group-policy name external server-group server_group password
server_password}
```



Note For an external group policy, RADIUS is the only supported AAA server type.

Step 2 Enter the group-policy attributes mode, using the **group-policy attributes** command in global configuration mode.

```
hostname(config)# group-policy name attributes
hostname(config-group-policy)#
```

The prompt changes to indicate the mode change. The group-policy-attributes mode lets you configure attribute-value pairs for a specified group policy. In group-policy-attributes mode, explicitly configure the attribute-value pairs that you do not want to inherit from the default group. The commands to do this are described in the following steps.

Step 3 Specify the primary and secondary WINS servers:

```
hostname(config-group-policy)# wins-server value {ip_address [ip_address] | none}
```

The first IP address specified is that of the primary WINS server. The second (optional) IP address is that of the secondary WINS server. Specifying the **none** keyword instead of an IP address sets WINS servers to a null value, which allows no WINS servers and prevents inheriting a value from a default or specified group policy.

Every time that you enter the **wins-server** command, you overwrite the existing setting. For example, if you configure WINS server x.x.x.x and then configure WINS server y.y.y.y, the second command overwrites the first, and y.y.y.y becomes the sole WINS server. The same is true for multiple servers. To add a WINS server rather than overwrite previously configured servers, include the IP addresses of all WINS servers when you enter this command.

The following example shows how to configure WINS servers with the IP addresses 10.10.10.15 and 10.10.10.30 for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# wins-server value 10.10.10.15 10.10.10.30
```

Step 4 Specify the primary and secondary DNS servers:

```
hostname(config-group-policy)# dns-server value {ip_address [ip_address] | none}
```

The first IP address specified is that of the primary DNS server. The second (optional) IP address is that of the secondary DNS server. Specifying the **none** keyword instead of an IP address sets DNS servers to a null value, which allows no DNS servers and prevents inheriting a value from a default or specified group policy.

Every time that you enter the **dns-server** command you overwrite the existing setting. For example, if you configure DNS server x.x.x.x and then configure DNS server y.y.y.y, the second command overwrites the first, and y.y.y.y becomes the sole DNS server. The same is true for multiple servers. To add a DNS server rather than overwrite previously configured servers, include the IP addresses of all DNS servers when you enter this command.

The following example shows how to configure DNS servers with the IP addresses 10.10.10.15, and 10.10.10.30 for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# dns-server value 10.10.10.15 10.10.10.30
```

Step 5 Set the VPN access hours. To do this, you associate a group policy with a configured time-range policy, using the **vpn-access-hours** command in group-policy configuration mode.

```
hostname(config-group-policy)# vpn-access-hours value {time-range | none}
```

A group policy can inherit a time-range value from a default or specified group policy. To prevent this inheritance, enter the **none** keyword instead of the name of a time-range in this command. This keyword sets VPN access hours to a null value, which allows no time-range policy.

The time-range variable is the name of a set of access hours defined in global configuration mode using the **time-range** command. The following example shows how to associate the group policy named “FirstGroup” with a time-range policy called “824”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-access-hours value 824
```

Step 6 Specify the number of simultaneous logins allowed for any user, using the **vpn-simultaneous-logins** command in group-policy configuration mode.

```
hostname(config-group-policy)# vpn-simultaneous-logins integer
```

The default value is 3. The range is an integer in the range 0 through 2147483647. A group policy can inherit this value from another group policy. Enter 0 to disable login and prevent user access. The following example shows how to allow a maximum of 4 simultaneous logins for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-simultaneous-logins 4
```



Note While there is no maximum limit to the number of simultaneous logins, allowing several could compromise security and affect performance.

- Step 7** Configure the user timeout period by entering the **vpn-idle-timeout** command in group-policy configuration mode or in username configuration mode:

```
hostname(config-group-policy)# vpn-idle-timeout {minutes | none}
```

The minimum time is 1 minute, and the maximum time is 35791394 minutes. The default is 30 minutes. If there is no communication activity on the connection in this period, the security appliance terminates the connection.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying a number of minutes with this command. The none keyword also permits an unlimited idle timeout period. It sets the idle timeout to a null value, thereby disallowing an idle timeout.

The following example shows how to set a VPN idle timeout of 15 minutes for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-idle-timeout 15
```

- Step 8** Configure a maximum amount of time for VPN connections, using the **vpn-session-timeout** command in group-policy configuration mode or in username configuration mode.

```
hostname(config-group-policy)# vpn-session-timeout {minutes | none}
```

The minimum time is 1 minute, and the maximum time is 35791394 minutes. There is no default value. At the end of this period of time, the security appliance terminates the connection.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying a number of minutes with this command. Specifying the **none** keyword permits an unlimited session timeout period and sets session timeout with a null value, which disallows a session timeout.

The following example shows how to set a VPN session timeout of 180 minutes for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-session-timeout 180
```

- Step 9** Specify the name of the ACL to use for VPN connections, using the **vpn-filter** command in group policy or username mode.

```
hostname(config-group-policy)# vpn-filter {value ACL name | none}
```

To remove the ACL, including a null value created by entering the **vpn-filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from another group policy.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying an ACL name. The **none** keyword indicates that there is no access list and sets a null value, thereby disallowing an access list.

You configure ACLs to permit or deny various types of traffic for this group policy. You then enter the **vpn-filter** command to apply those ACLs.

The following example shows how to set a filter that invokes an access list named “acl_vpn” for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-filter acl_vpn
```

Step 10 Specify the VPN tunnel type (IPSec or WebVPN) for this group policy.

```
hostname(config-group-policy)# vpn-tunnel-protocol {webvpn | IPSec}
```

The default is IPSec. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-group-policy)# no vpn-tunnel-protocol [webvpn | IPSec]
```

The parameter values for this command follow:

IPSec—Negotiates an IPSec tunnel between two peers (a remote access client or another secure gateway). Creates security associations that govern authentication, encryption, encapsulation, and key management.

webvpn—Provides VPN services to remote users via an HTTPS-enabled web browser, and does not require a client.

Enter this command to configure one or more tunneling modes. You must configure at least one tunneling mode for users to connect over a VPN tunnel.

The following example shows how to configure the IPSec tunneling mode for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-tunnel-protocol IPSec
```

Step 11 Specify whether to let users store their login passwords on the client system, using the **password-storage** command with the **enable** keyword in group-policy configuration mode. To disable password storage, use the **password-storage** command with the **disable** keyword.

```
hostname(config-group-policy)# password-storage {enable | disable}
```

For security reasons, password storage is disabled by default. Enable password storage only on systems that you know to be in secure sites.

To remove the password-storage attribute from the running configuration, enter the **no** form of this command:

```
hostname(config-group-policy)# no password-storage
```

Specifying the **no** form enables inheritance of a value for password-storage from another group policy.

This command does not apply to interactive hardware client authentication or individual user authentication for hardware clients.

The following example shows how to enable password storage for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# password-storage enable
```

Step 12 Specify whether to enable IP compression, which is disabled by default.

```
hostname(config-group-policy)# ip-comp {enable | disable}
```

To enable LZS IP compression, enter the **ip-comp** command with the **enable** keyword in group-policy configuration mode. To disable IP compression, enter the **ip-comp** command with the **disable** keyword.

To remove the **ip-comp** attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value from another group policy.

```
hostname(config-group-policy)# no ip-comp
```

Enabling data compression might speed up data transmission rates for remote dial-in users connecting with modems.



Caution

Data compression increases the memory requirement and CPU usage for each user session and consequently decreases the overall throughput of the security appliance. For this reason, we recommend that you enable data compression only for remote users connecting with a modem. Design a group policy specific to modem users, and enable compression only for them.

Step 13 Specify whether to require that users reauthenticate on IKE rekey by using the **re-xauth** command with the **enable** keyword in group-policy configuration mode. To disable user reauthentication on IKE rekey, enter the **disable** keyword.

```
hostname(config-group-policy)# re-xauth {enable | disable}
```

To remove the re-xauth attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for reauthentication on IKE rekey from another group policy.

```
hostname(config-group-policy)# no re-xauth
```

Reauthentication on IKE rekey is disabled by default. If you enable reauthentication on IKE rekey, the security appliance prompts the user to enter a username and password during initial Phase 1 IKE negotiation and also prompts for user authentication whenever an IKE rekey occurs. Reauthentication provides additional security.

If the configured rekey interval is very short, users might find the repeated authorization requests inconvenient. To avoid repeated authorization requests, disable reauthentication. To check the configured rekey interval, in monitoring mode, enter the **show crypto ipsec sa** command to view the security association lifetime in seconds and lifetime in kilobytes of data.



Note Reauthentication fails if there is no user at the other end of the connection.

Step 14 Specify whether to restrict remote users to access through the tunnel group only, using the **group-lock** command in group-policy configuration mode.

```
hostname(config-group-policy)# group-lock {value tunnel-grp-name | none}
hostname(config-group-policy)# no group-lock
```

The *tunnel-grp-name* variable specifies the name of an existing tunnel group that the security appliance requires for the user to connect. Group-lock restricts users by checking if the group configured in the VPN client is the same as the tunnel group to which the user is assigned. If it is not, the security appliance prevents the user from connecting. If you do not configure group-lock, the security appliance authenticates users without regard to the assigned group. Group locking is disabled by default.

To remove the **group-lock** attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from another group policy.

To disable group-lock, enter the **group-lock** command with the **none** keyword. The none keyword sets group-lock to a null value, thereby allowing no group-lock restriction. It also prevents inheriting a group-lock value from a default or specified group policy

- Step 15** Specify whether to enable perfect forward secrecy by using the **pfs** command with the **enable** keyword in group-policy configuration mode.

```
hostname(config-group-policy)# pfs {enable | disable}
```

In IPSec negotiations, PFS ensures that each new cryptographic key is unrelated to any previous key. PFS is disabled by default.

To disable PFS, enter the **disable** keyword.

To remove the PFS attribute from the running configuration, enter the **no** form of this command. A group policy can inherit a value for PFS from another group policy. To prevent inheriting a value, enter the **no** form of this command.

```
hostname(config-group-policy)# no pfs
```

- Step 16** Specify the banner, or welcome message, if any, that you want to display. The default is no banner. The message that you specify is displayed on remote clients when they connect. To specify a banner, enter the **banner** command in group-policy configuration mode. The banner text can be up to 510 characters long. Enter the “\n” sequence to insert a carriage return.



Note A carriage-return/line-feed included in the banner counts as two characters.

To delete a banner, enter the **no** form of this command. Be aware that using the **no** version of the command deletes all banners for the group policy.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying a value for the banner string, as follows:

```
hostname(config-group-policy)# banner {value banner_string | none}
```

The following example shows how to create a banner for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# banner value Welcome to Cisco Systems 7.0.
```

- Step 17** Specify whether to enable IPSec over UDP. To use IPSec over UDP, you must also configure the **ipsec-udp-port** command, as follows:

```
hostname(config-group-policy)# ipsec-udp {enable | disable}
hostname(config-group-policy)# no ipsec-udp
```

IPSec over UDP, sometimes called IPSec through NAT, lets a Cisco VPN client or hardware client connect via UDP to a security appliance that is running NAT. It is disabled by default. To enable IPSec over UDP, configure the **ipsec-udp** command with the **enable** keyword in group-policy configuration mode. To disable IPSec over UDP, enter the **disable** keyword. To remove the IPSec over UDP attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for IPSec over UDP from another group policy.

The Cisco VPN client must also be configured to use IPSec over UDP (it is configured to use it by default). The VPN 3002 requires no configuration to use IPSec over UDP.

IPSec over UDP is proprietary; it applies only to remote-access connections, and it requires mode configuration. The security appliance exchanges configuration parameters with the client while negotiating SAs. Using IPSec over UDP may slightly degrade system performance.

The following example shows how to set IPsec over UDP for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# ipsec-udp enable
```

If you enabled IPsec over UDP, you must also configure the **ipsec-udp-port** command in group-policy configuration mode. This command sets a UDP port number for IPsec over UDP. In IPsec negotiations, the security appliance listens on the configured port and forwards UDP traffic for that port even if other filter rules drop UDP traffic. The port numbers can range from 4001 through 49151. The default port value is 10000.

To disable the UDP port, enter the **no** form of this command. This enables inheritance of a value for the IPsec over UDP port from another group policy.

```
hostname(config-group-policy)# ipsec-udp-port port
```

The following example shows how to set an IPsec UDP port to port 4025 for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# ipsec-udp-port 4025
```

Step 18 Set the rules for tunneling traffic by specifying the split-tunneling policy.

```
hostname(config-group-policy)# split-tunnel-policy {tunnelall | tunnelspecified |
excludespecified}
hostname(config-group-policy)# no split-tunnel-policy
```

The default is to tunnel all traffic. To set a split tunneling policy, enter the **split-tunnel-policy** command in group-policy configuration mode. To remove the **split-tunnel-policy** attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for split tunneling from another group policy.

Split tunneling lets a remote-access IPsec client conditionally direct packets over an IPsec tunnel in encrypted form or to a network interface in clear text form. With split tunneling enabled, packets not bound for destinations on the other side of the IPsec tunnel do not have to be encrypted, sent across the tunnel, decrypted, and then routed to a final destination. This command applies this split tunneling policy to a specific network.

The **excludespecified** keyword defines a list of networks to which traffic goes in the clear. This feature is useful for remote users who want to access devices on their local network, such as printers, while they are connected to the corporate network through a tunnel. This option applies only to the Cisco VPN client.

The **tunnelall** keyword specifies that no traffic goes in the clear or to any other destination than the security appliance. This, in effect, disables split tunneling. Remote users reach internet networks through the corporate network and do not have access to local networks. This is the default option.

The **tunnelspecified** keyword tunnels all traffic from or to the specified networks. This option enables split tunneling. It lets you create a network list of addresses to tunnel. Data to all other addresses travels in the clear and is routed by the remote user’s Internet service provider.



Note Split tunneling is primarily a traffic management feature, not a security feature. For optimum security, we recommend that you do not enable split tunneling.

The following example shows how to set a split tunneling policy of tunneling only specified networks for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-tunnel-policy tunnelspecified
```

- Step 19** Create a network list for split tunneling using the **split-tunnel-network-list** command in group-policy configuration mode.

```
hostname(config-group-policy)# split-tunnel-network-list {value access-list_name | none}
hostname(config-group-policy)# no split-tunnel-network-list value [access-list_name]
```

Split tunneling network lists distinguish networks that require traffic to travel across the tunnel from those that do not require tunneling. The security appliance makes split tunneling decisions on the basis of a network list, which is an ACL that consists of a list of addresses on the private network. Only standard-type ACLs are allowed.

The **value** *access-list name* parameter identifies an access list that enumerates the networks to tunnel or not tunnel.

The **none** keyword indicates that there is no network list for split tunneling; the security appliance tunnels all traffic. Specifying the **none** keyword sets a split tunneling network list with a null value, thereby disallowing split tunneling. It also prevents inheriting a default split tunneling network list from a default or specified group policy.

To delete a network list, enter the **no** form of this command. To delete all split tunneling network lists, enter the **no split-tunnel-network-list** command without arguments. This command deletes all configured network lists, including a null list if you created one by entering the **none** keyword.

When there are no split tunneling network lists, users inherit any network lists that exist in the default or specified group policy. To prevent users from inheriting such network lists, enter the **split-tunnel-network-list none** command.

The following example shows how to set a network list called “FirstList” for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-tunnel-network-list FirstList
```

- Step 20** Specify the default domain name. To set a default domain name for users of the group policy, enter the **default-domain** command in group-policy configuration mode. To delete a domain name, enter the **no** form of this command.

```
hostname(config-group-policy)# default-domain {value domain-name | none}
hostname(config-group-policy)# no default-domain [domain-name]
```

The security appliance passes the default domain name to the IPsec client to append to DNS queries that omit the domain field. This domain name applies only to tunneled packets. When there are no default domain names, users inherit the default domain name in the default group policy.

The **value** *domain-name* parameter identifies the default domain name for the group. To specify that there is no default domain name, enter the **none** keyword. This command sets a default domain name with a null value, which disallows a default domain name and prevents inheriting a default domain name from a default or specified group policy.

To delete all default domain names, enter the **no default-domain** command without arguments. This command deletes all configured default domain names, including a null list if you created one by entering the **default-domain** command with the **none** keyword. The **no** form allows inheriting a domain name.

The following example shows how to set a default domain name of “FirstDomain” for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# default-domain value FirstDomain
```

- Step 21** Enter a list of domains to be resolved through the split tunnel. Enter the **split-dns** command in group-policy configuration mode. To delete a list, enter the **no** form of this command.

**Note**

The AnyConnect VPN client and the SSL VPN Client do not support split DNS.

When there are no split tunneling domain lists, users inherit any that exist in the default group policy. To prevent users from inheriting such split tunneling domain lists, enter the **split-dns** command with the **none** keyword.

To delete all split tunneling domain lists, enter the **no split-dns** command without arguments. This deletes all configured split tunneling domain lists, including a null list created by issuing the **split-dns** command with the **none** keyword.

The parameter **value domain-name** provides a domain name that the security appliance resolves through the split tunnel. The **none** keyword indicates that there is no split DNS list. It also sets a split DNS list with a null value, thereby disallowing a split DNS list, and prevents inheriting a split DNS list from a default or specified group policy.

```
hostname(config-group-policy)# split-dns {value domain-name1 [domain-name2...
domain-nameN] | none}
hostname(config-group-policy)# no split-dns [domain-name domain-name2 domain-nameN]
```

Enter a single space to separate each entry in the list of domains. There is no limit on the number of entries, but the entire string can be no longer than 255 characters. You can use only alphanumeric characters, hyphens (-), and periods (.).

The following example shows how to configure the domains Domain1, Domain2, Domain3, and Domain4 to be resolved through split tunneling for the group policy named "FirstGroup":

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-dns value Domain1 Domain2 Domain3 Domain4
```

- Step 22** Specify whether to enable secure unit authentication by entering the **secure-unit-authentication** command with the **enable** keyword in group-policy configuration mode.

```
hostname(config-group-policy)# secure-unit-authentication {enable | disable}
hostname(config-group-policy)# no secure-unit-authentication
```

Secure unit authentication provides additional security by requiring VPN hardware clients to authenticate with a username and password each time that the client initiates a tunnel. With this feature enabled, the hardware client does not have a saved username and password. Secure unit authentication is disabled by default.

To disable secure unit authentication, enter the **disable** keyword. To remove the secure unit authentication attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for secure unit authentication from another group policy.

**Note**

With this feature enabled, to bring up a VPN tunnel, a user must be present to enter the username and password.

Secure unit authentication requires that you have an authentication server group configured for the tunnel group the hardware client(s) use.

If you require secure unit authentication on the primary security appliance, be sure to configure it on any backup servers as well.

The following example shows how to enable secure unit authentication for the group policy named "FirstGroup":

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# secure-unit-authentication enable
```

- Step 23** Specify whether to enable user authentication by entering the **user-authentication** command with the **enable** keyword in group-policy configuration mode.

```
hostname(config-group-policy)# user-authentication {enable | disable}
hostname(config-group-policy)# no user-authentication
```

To disable user authentication, enter the **disable** keyword. To remove the user authentication attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for user authentication from another group policy.

User authentication is disabled by default. When enabled, user authentication requires that individual users behind a hardware client authenticate to gain access to the network across the tunnel. Individual users authenticate according to the order of authentication servers that you configure.

If you require user authentication on the primary security appliance, be sure to configure it on any backup servers as well.

The following example shows how to enable user authentication for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# user-authentication enable
```

- Step 24** Set an idle timeout for individual users behind hardware clients, using the **user-authentication-idle-timeout** command in group-policy configuration mode.

```
hostname(config-group-policy)# user-authentication-idle-timeout {minutes | none}
hostname(config-group-policy)# no user-authentication-idle-timeout
```

The minutes parameter specifies the number of minutes in the idle timeout period. The minimum is 1 minute, the default is 30 minutes, and the maximum is 35791394 minutes.

To delete the idle timeout value, enter the **no** form of this command. This option allows inheritance of an idle timeout value from another group policy.

To prevent inheriting an idle timeout value, enter the **user-authentication-idle-timeout** command with the **none** keyword. This command sets the idle timeout with a null value, which disallows an idle timeout and prevents inheriting an user authentication idle timeout value from a default or specified group policy.

If there is no communication activity by a user behind a hardware client in the idle timeout period, the security appliance terminates the client’s access.



Note The **user-authentication-idle-timeout** command terminates only the client’s access through the VPN tunnel, not the VPN tunnel itself.

The following example shows how to set an idle timeout value of 45 minutes for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# user-authentication-idle-timeout 45
```

- Step 25** To enable IP Phone Bypass, enter the **ip-phone-bypass** command with the **enable** keyword in group-policy configuration mode. IP Phone Bypass lets IP phones behind hardware clients connect without undergoing user authentication processes. IP Phone Bypass is disabled by default. If enabled, secure unit authentication remains in effect.

To disable IP Phone Bypass, enter the **disable** keyword. To remove the IP phone Bypass attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for IP Phone Bypass from another group policy.

```
hostname(config-group-policy)# ip-phone-bypass {enable | disable}
hostname(config-group-policy)# no ip-phone-bypass
```

- Step 26** Specify whether to enable LEAP Bypass. To enable LEAP Bypass, enter the **leap-bypass** command with the **enable** keyword in group-policy configuration mode. To disable LEAP Bypass, enter the **disable** keyword. To remove the LEAP Bypass attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for LEAP Bypass from another group policy.

```
hostname(config-group-policy)# leap-bypass {enable | disable}
hostname(config-group-policy)# no leap-bypass
```

When LEAP Bypass is enabled, LEAP packets from wireless devices behind a VPN hardware client travel across a VPN tunnel prior to user authentication. This action lets workstations using Cisco wireless access point devices establish LEAP authentication and then authenticate again per user authentication. LEAP Bypass is disabled by default.



Note IEEE 802.1X is a standard for authentication on wired and wireless networks. It provides wireless LANs with strong mutual authentication between clients and authentication servers, which can provide dynamic per-user, per session wireless encryption privacy (WEP) keys, removing administrative burdens and security issues that are present with static WEP keys.

Cisco Systems has developed an 802.1X wireless authentication type called Cisco LEAP. LEAP (Lightweight Extensible Authentication Protocol) implements mutual authentication between a wireless client on one side of a connection and a RADIUS server on the other side. The credentials used for authentication, including a password, are always encrypted before they are transmitted over the wireless medium.

Cisco LEAP authenticates wireless clients to RADIUS servers. It does not include RADIUS accounting services.

This feature does not work as intended if you enable interactive hardware client authentication.



Caution

There might be security risks to your network in allowing any unauthenticated traffic to traverse the tunnel.

The following example shows how to set LEAP Bypass for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# leap-bypass enable
```

- Step 27** Enable network extension mode for hardware clients by entering the **nem** command with the **enable** keyword in group-policy configuration mode.

```
hostname(config-group-policy)# nem {enable | disable}
hostname(config-group-policy)# no nem
```

Network Extension mode lets hardware clients present a single, routable network to the remote private network over the VPN tunnel. IPSec encapsulates all traffic from the private network behind the hardware client to networks behind the security appliance. PAT does not apply. Therefore, devices

behind the security appliance have direct access to devices on the private network behind the hardware client over the tunnel, and only over the tunnel, and vice versa. The hardware client must initiate the tunnel, but after the tunnel is up, either side can initiate data exchange.

To disable NEM, enter the **disable** keyword. To remove the NEM attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from another group policy.

The following example shows how to set NEM for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# nem enable
```

Step 28 Configure backup servers if you plan on using them. IPsec backup servers let a VPN client connect to the central site when the primary security appliance is unavailable. To configure backup servers, enter the **backup-servers** command in group-policy configuration mode.

```
hostname(config-group-policy)# backup-servers {server1 server2... server10 |
clear-client-config | keep-client-config}
```

When you configure backup servers, the security appliance pushes the server list to the client as the IPsec tunnel is established. Backup servers do not exist until you configure them, either on the client or on the primary security appliance.

To remove a backup server, enter the **no** form of this command. To remove the backup-servers attribute from the running configuration and enable inheritance of a value for backup-servers from another group policy, enter the **no** form of this command without arguments.

```
hostname(config-group-policy)# no backup-servers [server1 server2... server10 |
clear-client-config | keep-client-config]
```

The **clear-client-config** keyword specifies that the client uses no backup servers. The security appliance pushes a null server list.

The **keep-client-config** keyword specifies that the security appliance sends no backup server information to the client. The client uses its own backup server list, if configured. This is the default.

The *server1 server 2... server10* parameter list is a space-delimited, priority-ordered list of servers for the VPN client to use when the primary security appliance is unavailable. This list identifies servers by IP address or hostname. The list can be 500 characters long, but it can contain only 10 entries.

Configure backup servers either on the client or on the primary security appliance. If you configure backup servers on the security appliance, it pushes the backup server policy to the clients in the group, replacing the backup server list on the client if one is configured.



Note If you are using hostnames, it is wise to have backup DNS and WINS servers on a separate network from that of the primary DNS and WINS servers. Otherwise, if clients behind a hardware client obtain DNS and WINS information from the hardware client via DHCP, and the connection to the primary server is lost, and the backup servers have different DNS and WINS information, clients cannot be updated until the DHCP lease expires. In addition, if you use hostnames and the DNS server is unavailable, significant delays can occur.

The following example shows how to configure backup servers with IP addresses 10.10.10.1 and 192.168.10.14, for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# backup-servers 10.10.10.1 192.168.10.14
```

Step 29 Set personal firewall policies that the security appliance pushes to the VPN client during IKE tunnel negotiation by using the **client-firewall** command in group-policy configuration mode. To delete a firewall policy, enter the **no** form of this command.

To delete all firewall policies, enter the **no client-firewall** command without arguments. This command deletes all configured firewall policies, including a null policy if you created one by entering the **client-firewall** command with the **none** keyword.

When there are no firewall policies, users inherit any that exist in the default or other group policy. To prevent users from inheriting such firewall policies, enter the **client-firewall** command with the **none** keyword.

Enter the following commands to set the appropriate client firewall parameters. [Table 25-1](#), following this set of commands, explains the syntax elements of these commands:

```
hostname(config-group-policy)# client-firewall none

hostname(config-group-policy)# client-firewall opt | req custom vendor-id num product-id num policy AYT | {CPP acl-in ACL acl-out ACL} [description string]

hostname(config-group-policy)# client-firewall opt | req zonelabs-zonealarm policy AYT | {CPP acl-in ACL acl-out ACL}

hostname(config-group-policy)# client-firewall opt | req zonelabs-zonealarmpro policy AYT | {CPP acl-in ACL acl-out ACL}

client-firewall opt | req zonelabs-zonealarmpro policy AYT | {CPP acl-in ACL acl-out ACL}

hostname(config-group-policy)# client-firewall opt | req cisco-integrated acl-in ACL acl-out ACL

hostname(config-group-policy)# client-firewall opt | req sygate-personal

hostname(config-group-policy)# client-firewall opt | req sygate-personal-pro

hostname(config-group-policy)# client-firewall opt | req sygate-security-agent

hostname(config-group-policy)# client-firewall opt | req networkkice-blackkice

hostname(config-group-policy)# client-firewall opt | req cisco-security-agent
```

Table 25-1 *client-firewall Command Parameters*

Parameter	Description
acl-in <ACL>	Provides the policy the client uses for inbound traffic.
acl-out <ACL>	Provides the policy the client uses for outbound traffic.
AYT	Specifies that the client PC firewall application controls the firewall policy. The security appliance checks to make sure that the firewall is running. It asks, “Are You There?” If there is no response, the security appliance tears down the tunnel.
cisco-integrated	Specifies Cisco Integrated firewall type.
cisco-security-agent	Specifies Cisco Intrusion Prevention Security Agent firewall type.
CPP	Specifies Policy Pushed as source of the VPN client firewall policy.
custom	Specifies Custom firewall type.
description <string>	Describes the firewall.
networkkice-blackkice	Specifies Network ICE Black ICE firewall type.

Table 25-1 *client-firewall Command Parameters (continued)*

none	Indicates that there is no client firewall policy. Sets a firewall policy with a null value, thereby disallowing a firewall policy. Prevents inheriting a firewall policy from a default or specified group policy.
opt	Indicates an optional firewall type.
product-id	Identifies the firewall product.
req	Indicates a required firewall type.
sygate-personal	Specifies Sygate Personal firewall type.
sygate-personal-pro	Specifies Sygate Personal Pro firewall type.
sygate-security-agent	Specifies Sygate Security Agent firewall type.
vendor-id	Identifies the firewall vendor.
zonelabs-zonealarm	Specifies Zone Labs Zone Alarm firewall type.
zonelabs-zonealarmorpro policy	Specifies Zone Labs Zone Alarm or Pro firewall type.
zonelabs-zonealarmpro policy	Specifies Zone Labs Zone Alarm Pro firewall type.

The following example shows how to set a client firewall policy that requires Cisco Intrusion Prevention Security Agent for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# client-firewall req cisco-security-agent
```

Step 30 Configure rules that limit the remote access client types and versions that can connect via IPSec through the security appliance by using the **client-access-rule** command in group-policy configuration mode. To delete a rule, enter the **no** form of this command. This command is equivalent to the following command:

```
hostname(config-group-policy)# client-access-rule 1 deny type "Cisco VPN Client" version 4.0
```

To delete all rules, enter the **no client-access-rule command** without arguments. This deletes all configured rules, including a null rule if you created one by issuing the **client-access-rule** command with the **none** keyword.

By default, there are no access rules. When there are no client access rules, users inherit any rules that exist in the default group policy.

To prevent users from inheriting client access rules, enter the **client-access-rule** command with the **none** keyword. The result of this command is that all client types and versions can connect.

```
hostname(config-group-policy)# client-access rule priority {permit | deny} type type version {version | none}
```

```
hostname(config-group-policy)# no client-access rule [priority {permit | deny} type type version version]
```

Table 25-2 explains the meaning of the keywords and parameters in these commands.

Table 25-2 *client-access rule Command Parameters*

Parameter	Description
deny	Denies connections for devices of a particular type and/or version.
none	Allows no client access rules. Sets client-access-rule to a null value, thereby allowing no restriction. Prevents inheriting a value from a default or specified group policy.
permit	Permits connections for devices of a particular type and/or version.
priority	Determines the priority of the rule. The rule with the lowest integer has the highest priority. Therefore, the rule with the lowest integer that matches a client type and/or version is the rule that applies. If a lower priority rule contradicts, the security appliance ignores it.
type <i>type</i>	Identifies device types via free-form strings, for example VPN 3002. A string must match exactly its appearance in the show vpn-sessiondb remote display, except that you can enter the * character as a wildcard.
version <i>version</i>	Identifies the device version via free-form strings, for example 7.0. A string must match exactly its appearance in the show vpn-sessiondb remote display, except that you can enter the * character as a wildcard.

Construct rules according to these guidelines:

- If you do not define any rules, the security appliance permits all connection types.
- When a client matches none of the rules, the security appliance denies the connection. If you define a deny rule, you must also define at least one permit rule, or the security appliance denies all connections.
- For both software and hardware clients, type and version must match exactly their appearance in the **show vpn-sessiondb remote** display.
- The * character is a wildcard, which you can enter multiple times in each rule. For example, **client-access rule 3 deny type * version 3.*** creates a priority 3 client access rule that denies all client types running release versions 3.x software.
- You can construct a maximum of 25 rules per group policy.
- There is a limit of 255 characters for an entire set of rules.
- You can enter n/a for clients that do not send client type and/or version.

The following example shows how to create client access rules for the group policy named “FirstGroup”. These rules permit Cisco VPN clients running software version 4.x, while denying all Windows NT clients:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# client-access-rule 1 deny type WinNT version *
hostname(config-group-policy)# client-access-rule 2 permit "Cisco VPN Client" version 4.*
```



Note The “type” field is a free-form string that allows any value, but that value must match the fixed value that the client sends to the security appliance at connect time.

Step 31 Customize a WebVPN configuration for specific users or group policies. Enter `webvpn` mode by using the `webvpn` command in group-policy configuration mode. Webvpn commands for group policies define access to files, MAPI proxy, URLs and TCP applications over WebVPN. They also identify ACLs and types of traffic to filter. WebVPN is disabled by default.

To remove all commands entered in `webvpn` mode, enter the `no` form of this command. These `webvpn` commands apply to the username or group policy from which you configure them.

```
hostname(config-group-policy)# webvpn
hostname(config-group-policy)# no webvpn
```

You do not need to configure WebVPN to use e-mail proxies.

WebVPN lets users establish a secure, remote-access VPN tunnel to the security appliance using a web browser. There is no need for either a software or hardware client. WebVPN provides easy access to a broad range of web resources and web-enabled applications from almost any computer that can reach HTTPS Internet sites. WebVPN uses SSL and its successor, TLS1 to provide a secure connection between remote users and specific, supported internal resources that you configure at a central site. The security appliance recognizes connections that need to be proxied, and the HTTP server interacts with the authentication subsystem to authenticate users.



Note

The `webvpn` mode that you enter from global configuration mode lets you configure global settings for WebVPN. The `webvpn` mode described in this section, which you enter from group-policy mode, lets you customize a WebVPN configuration for specific group policies.

In `webvpn` mode, you can customize the following parameters, each of which is described in the subsequent steps:

- `functions url-entry`
- `html-content-filter`
- `homepage`
- `filter`
- `url-list`
- `port-forward`
- `port-forward-name` value Application Access

The following example shows how to enter `webvpn` mode for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)#
```

Step 32 Configure the WebVPN functions that you want to enable. To configure file access and file browsing, HTTP Proxy, MAPI Proxy, and URL entry over WebVPN for this group policy, enter the `functions` command in `webvpn` mode.

```
hostname(config-username-webvpn)# functions {file-access | file-browsing | file-entry |
http-proxy | url-entry | mapi | none}

hostname(config-username-webvpn)# no functions [file-access | file-browsing | file-entry |
http-proxy | url-entry | mapi]
```

To remove a configured function, enter the `no` form of this command. These functions are disabled by default.

To remove all configured functions, including a null value created by issuing the **functions none** command, enter the **no** form of this command without arguments. The **no** option allows inheritance of a value from another group policy. To prevent inheriting function values, enter the **functions none** command.

The following table describes the meaning of the keywords used in this command.

file-access	Enables or disables file access. When enabled, the WebVPN home page lists file servers in the server list. You must enable file access to enable file browsing and/or file entry.
file-browsing	Enables or disables browsing for file servers and shares. You must enable file browsing to allow user entry of a file server.
file-entry	Enables or disables user ability to enter names of file servers.
http-proxy	Enables or disables the forwarding of an HTTP applet proxy to the client. The proxy is useful for technologies that interfere with proper mangling, such as Java, ActiveX, and Flash. It bypasses mangling while ensuring the continued use of the security appliance. The forwarded proxy modifies the browser's old proxy configuration automatically and redirects all HTTP and HTTPS requests to the new proxy configuration. It supports virtually all client side technologies, including HTML, CSS, JavaScript, VBScript, ActiveX, and Java. The only browser it supports is Microsoft Internet Explorer.
mapi	Enables or disables Microsoft Outlook/Exchange port forwarding.
none	Sets a null value for all WebVPN functions . Prevents inheriting functions from a default or specified group policy
url-entry	Enables or disables user entry of URLs. When enabled, the security appliance still restricts URLs with any configured URL or network ACLs. When URL entry is disabled, the security appliance restricts WebVPN users to the URLs on the home page.

The following example shows how to configure file access, file browsing, and MAPI Proxy for the group policy named "FirstGroup":

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# functions file-access file-browsing MAPI
```

- Step 33** Specify whether to filter Java, ActiveX, images, scripts, and cookies for WebVPN sessions for this group policy by using the **html-content-filter** command in webvpn mode. To remove a content filter, enter the **no** form of this command. To remove all content filters, including a null value created by issuing the **html-content-filter** command with the **none** keyword, enter the **no** form of this command without arguments. The **no** option allows inheritance of a value from another group policy. To prevent inheriting an html content filter, enter the **html-content-filter** command with the **none** keyword. HTML filtering is disabled by default.

Using the command a second time overrides the previous setting.

```
hostname(config-username-webvpn)# html-content-filter {java | images | scripts | cookies | none}

hostname(config-username-webvpn)# no html-content-filter [java | images | scripts | cookies | none]
```

The following table describes the meaning of the keywords used in this command.

cookies	Removes cookies from images, providing limited ad filtering and privacy.
images	Removes references to images (removes tags).
java	Removes references to Java and ActiveX (removes <EMBED>, <APPLET>, and <OBJECT> tags).
none	Indicates that there is no filtering. Sets a null value, thereby disallowing filtering. Prevents inheriting filtering values.
scripts	Removes references to scripting (removes <SCRIPT> tags).

The following example shows how to set filtering of JAVA and ActiveX, cookies, and images for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# html-content-filter java cookies images
```

- Step 34** Specify a URL for the web page that displays upon login for this WebVPN group policy by using the **homepage** command in webvpn mode. To remove a configured home page, including a null value created by issuing the **homepage none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from another group policy. To prevent inheriting a home page, enter the **homepage none** command.

The **none** keyword indicates that there is no WebVPN home page. It sets a null value, thereby disallowing a home page and prevents inheriting an home page.

The *url-string* variable following the keyword **value** provides a URL for the home page. The string must begin with either `http://` or `https://`.

There is no default home page.

```
hostname(config-username-webvpn)# homepage {value url-string | none}
hostname(config-username-webvpn)# no homepage
```

- Step 35** Specify the name of the access list to use for WebVPN connections for this group policy or username by using the **filter** command in webvpn mode. To remove the access list, including a null value created by issuing the **filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from another group policy. To prevent inheriting filter values, enter the **filter value none** command.

WebVPN access lists do not apply until you enter the **filter** command to specify them.

You configure ACLs to permit or deny various types of traffic for this group policy. You then enter the **filter** command to apply those ACLs for WebVPN traffic.

```
hostname(config-username-webvpn)# filter {value ACLname | none}
hostname(config-username-webvpn)# no filter
```

The **none** keyword indicates that there is no **webvpntype** access list. It sets a null value, thereby disallowing an access list and prevents inheriting an access list from another group policy.

The *ACLname* string following the keyword **value** provides the name of the previously configured access list.



Note

WebVPN does not use ACLs defined in the **vpn-filter** command.

The following example shows how to set a filter that invokes an access list named *acl_in* for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# filter acl_in
```

- Step 36** To apply a list of WebVPN servers and URLs to a particular group policy, enter the **url-list** command in webvpn mode, which you enter from group-policy or username mode. To remove a list, including a null value created by using the **url-list** command with the **none** keyword, enter the **no** form of this command. The **no** option allows inheritance of a value from another group policy. To prevent inheriting a url list, enter the **url-list** command with the **none** keyword.

```
hostname(config-username-webvpn)# url-list {listname displayname url | none}
hostname(config-username-webvpn)# no url-list
```

The following table describes the meaning of the keywords and variables used in this command.

displayname	Specifies a name for the URL. This name appears on the WebVPN end user interface.
listname	Identifies a name by which to group URLs.
none	Indicates that there is no list of URLs. Sets a null value, thereby disallowing a URL list. Prevents inheriting URL list values.
url	Specifies a URL that WebVPN users can access.

There is no default URL list.

Using the command a second time overrides the previous setting.

Before you can enter the **url-list** command in webvpn mode to identify a URL list that you want to display on the WebVPN home page for a group policy, you must create the list. Enter the **url-list** command in global configuration mode to create one or more lists.

The following example shows how to set a URL list called “FirstGroupURLs” for the group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# url-list value FirstGroupURLs
```

- Step 37** Enable WebVPN application access for this group policy by using the **port-forward** command in webvpn mode. To remove the port forwarding attribute from the configuration, including a null value created by issuing the **port-forward none** command, enter the **no** form of this command. The **no** option allows inheritance of a list from another group policy. To prevent inheriting a port forwarding list, enter the **port-forward** command with the **none** keyword.

```
hostname(config-username-webvpn)# port-forward {value listname | none}
hostname(config-username-webvpn)# no port-forward
```

The **none** keyword indicates that there is no filtering. It sets a null value, thereby disallowing a filtering, and prevents inheriting filtering values.

Port forwarding is disabled by default.

The *listname* string following the keyword **value** identifies the list of applications WebVPN users can access. Enter the port-forward command in configuration mode to define the list.

Using the command a second time overrides the previous setting.

Before you can enter the **port-forward** command in webvpn mode to enable application access, you must define a list of applications that you want users to be able to use in a WebVPN connection. Enter the **port-forward** command in global configuration mode to define this list.

The following example shows how to set a portforwarding list called *ports1* for the internal group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup internal attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# port-forward value ports1
```

- Step 38** Configure the display name that identifies TCP port forwarding to end users for a particular user or group policy by using the **port-forward-name** command in webvpn mode. To delete the display name, including a null value created by using the **port-forward-name none** command, enter the **no** form of the command. The **no** option restores the default name, “Application Access.” To prevent a display name, enter the **port-forward none** command.

```
hostname(config-username-webvpn)# port-forward-name {value name | none}
hostname(config-username-webvpn)# no port-forward-name
```

The following example shows how to set the name, “Remote Access TCP Applications,” for the internal group policy named “FirstGroup”:

```
hostname(config)# group-policy FirstGroup internal attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# port-forward-name value Remote Access TCP Applications
```

Configuring Users

By default, users inherit all user attributes from the assigned group policy. The security appliance also lets you assign individual attributes at the user level, overriding values in the group policy that applies to that user. For example, you can specify a group policy giving all users access during business hours, but give a specific user 24-hour access.

Viewing the Username Configuration

To display the configuration for all usernames, including default values inherited from the group policy, enter the **all** keyword with the **show running-config username** command, as follows:

```
hostname# show running-config all username
```

If you omit the **all** keyword, only explicitly configured values appear in this list. In this example, the usernames are “testuser” and “oliverw”. The configuration for all configured users, including the inherited values is as follows:

```
username testuser password 12RsxXQnphyr/I9Z encrypted privilege 15
username testuser attributes
  vpn-group-policy testing
  vpn-access-hours value averylongtime
  vpn-simultaneous-logins 4
  vpn-idle-timeout 30
  vpn-session-timeout none
  vpn-filter value tunneled
  no vpn-framed-ip-address
  group-lock value test
webvpn
  no functions
  html-content-filter java images scripts cookies
  no homepage
```

```

no filter
no url-list
no port-forward
no port-forward-name

username oliverw password vt/qqEzfgfrXXya4 encrypted privilege 2
username oliverw attributes
no vpn-group-policy
vpn-tunnel-protocol webvpn
no vpn-framed-ip-address
webvpn
functions url-entry file-access file-entry file-browsing
no html-content-filter
no homepage
no filter
no url-list
no port-forward
no port-forward-name
username cisco password 3USUcOPFUimCO4Jk encrypted privilege 15
username newuser nopassword privilege 15

```

Configuring Specific Users

To configure specific users, you assign a password (or no password) and attributes to a user using the **username** command, which enters username mode. Any attributes that you do not specify are inherited from the group policy.

The internal user authentication database consists of the users entered with the **username** command. The **login** command uses this database for authentication. To add a user to the security appliance database, enter the **username** command in global configuration mode. To remove a user, use the **no** version of this command with the username you want to remove. To remove all usernames, use the **clear configure username** command without appending a username.

Setting a User Password and Privilege Level

Use the **username** command to assign a password and a privilege level for a user. You can, instead, enter the **nopassword** keyword to specify that this user does not require a password. If you do specify a password, you can specify whether that password is stored in an encrypted form.

The optional privilege keyword lets you set a privilege level for this user. Privilege levels range from 0 (the lowest) through 15. System administrators generally have the highest privilege level. The default level is 2.

```
hostname(config)# username name {nopassword | password password [encrypted]} [privilege
priv_level]}
```

```
hostname(config)# no username [name]
```

The following table describes the meaning of the keywords and variables used in this command.

encrypted	Indicates that the password is encrypted.
<i>name</i>	Provides the name of the user.
nopassword	Indicates that this user needs no password.
password <i>password</i>	Indicates that this user has a password, and provides the password.
privilege <i>priv_level</i>	Sets a privilege level for this user. The range is from 0 to 15, with lower numbers having less ability to use commands and administer the security appliance. The default privilege level is 2. The typical privilege level for a system administrator is 15.

By default, VPN users that you add with this command have no attributes or group policy association. You must configure all values explicitly.

The following example shows how to configure a user named “anyuser” with a n encrypted password of pw_12345678 and a privilege level of 12:

```
hostname(config)# username anyuser password pw_12345678 encrypted privilege 12
```

Configuring User Attributes

After configuring the user’s password (if any) and privilege level, you set the other attributes. These can be in any order. To remove any attribute-value pair, enter the **no** form of the command.

Step 1 Enter username mode by entering the **username** command with the **attributes** keyword:

```
hostname(config)# username name attributes
hostname(config-username)#
```

The prompt changes to indicate the new mode. You can now configure the attributes.

Step 2 Specify the name of the group policy from which this user inherits attributes. By default, VPN users have no group policy association.

```
hostname(config-username)# vpn-group-policy group-policy-name
hostname(config-username)# no vpn-group-policy group-policy-name
```

Using this command lets users inherit attributes that you have not configured at the username level.

You can override the value of an attribute in a group policy for a particular user by configuring it in username mode, if that attribute is available in username mode.

The following example shows how to configure a user named “anyuser” to use attributes from the group policy named “FirstGroup”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-group-policy FirstGroup
```

Step 3 Associate the hours that this user is allowed to access the system by specifying the name of a configured time-range policy:

To remove the attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a time-range value from another group policy. To prevent inheriting a value, enter the **vpn-access-hours none** command. The default is unrestricted access.

```
hostname(config-username)# vpn-access-hours value {time-range | none}
hostname(config-username)# vpn-access-hours value none
```

The following example shows how to associate the user named “anyuser” with a time-range policy called 824:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-access-hours 824
```

- Step 4** Specify the maximum number of simultaneous logins allowed for this user. The range is 0 through 2147483647. The default is 3 simultaneous logins. To remove the attribute from the running configuration, enter the **no** form of this command. Enter 0 to disable login and prevent user access.

```
hostname(config-username)# vpn-simultaneous-logins integer
hostname(config-username)# no vpn-simultaneous-logins
```

The following example shows how to allow a maximum of 4 simultaneous logins for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-simultaneous-logins 4
```

- Step 5** Specify the idle timeout period in minutes, or enter **none** to disable the idle timeout. If there is no communication activity on the connection in this period, the security appliance terminates the connection.

The range is 1 through 35791394 minutes. The default is 30 minutes. To allow an unlimited timeout period, and thus prevent inheriting a timeout value, enter the **vpn-idle-timeout** command with the **none** keyword. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-username)# vpn-idle-timeout {minutes | none}
hostname(config-username)# no vpn-idle-timeout
```

The following example shows how to set a VPN idle timeout of 15 minutes for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-idle-timeout 30
```

- Step 6** Specify the maximum user connection time in minutes, or enter **none** to allow unlimited connection time and prevent inheriting a value for this attribute. At the end of this period of time, the security appliance terminates the connection.

The range is 1 through 35791394 minutes. There is no default timeout. To allow an unlimited timeout period, and thus prevent inheriting a timeout value, enter the **vpn-session-timeout** command with the **none** keyword. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-username)# vpn-session-timeout {minutes | none}
hostname(config-username)# no vpn-session-timeout
```

The following example shows how to set a VPN session timeout of 180 minutes for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-session-timeout 180
```

- Step 7** Specify the name of a previously-configured, user-specific ACL to use as a filter for VPN connections. To disallow an access list and prevent inheriting an access list from the group policy, enter the **vpn-filter** command with the **none** keyword. To remove the ACL, including a null value created by issuing the **vpn-filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. There are no default behaviors or values for this command.

You configure ACLs to permit or deny various types of traffic for this user. You then use the **vpn-filter** command to apply those ACLs.

```
hostname(config-username)# vpn-filter {value ACL name | none}
hostname(config-username)# no vpn-filter
```



Note WebVPN does not use the ACL defined in the **vpn-filter** command.

The following example shows how to set a filter that invokes an access list named “acl_vpn” for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-filter value acl_vpn
```

Step 8 Specify the IP address and netmask to assign to a particular user. To remove the IP address, enter the **no** form of this command.

```
hostname(config-username)# vpn-framed-ip-address {ip_address}
hostname(config-username)# no vpn-framed-ip-address
```

The following example shows how to set an IP address of 10.92.166.7 for a user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-framed-ip-address 10.92.166.7
```

Step 9 Specify the network mask to use with the IP address specified in the previous step. If you used the **no vpn-framed-ip-address** command, do not specify a network mask. To remove the subnet mask, enter the **no** form of this command. There is no default behavior or value.

```
hostname(config-username)# vpn-framed-ip-netmask {netmask}
hostname(config-username)# no vpn-framed-ip-netmask
```

The following example shows how to set a subnet mask of 255.255.255.254 for a user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-framed-ip-netmask 255.255.255.254
```

Step 10 Specify the VPN tunnel types (IPSec or WebVPN) that this user can use. The default is taken from the default group policy, the default for which is IPSec. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-username)# vpn-tunnel-protocol {webvpn | IPSec}
hostname(config-username)# no vpn-tunnel-protocol [webvpn | IPSec]
```

The parameter values for this command are as follows:

- **IPSec**—Negotiates an IPSec tunnel between two peers (a remote access client or another secure gateway). Creates security associations that govern authentication, encryption, encapsulation, and key management.
- **webvpn**—Provides VPN services to remote users via an HTTPS-enabled web browser, and does not require a client

Enter this command to configure one or more tunneling modes. You must configure at least one tunneling mode for users to connect over a VPN tunnel.

The following example shows how to configure WebVPN and IPSec tunneling modes for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-tunnel-protocol webvpn
hostname(config-username)# vpn-tunnel-protocol IPSec
```

- Step 11** Configure the **group-lock** attribute with the **value** keyword to restrict remote users to access only through the specified, preexisting tunnel group. To remove the **group-lock** attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from the group policy. To disable group-lock, and to prevent inheriting a group-lock value from a default or specified group policy, enter the **group-lock** command with the **none** keyword.

Group-lock restricts users by checking whether the group configured in the VPN client is the same as the tunnel group to which the user is assigned. If it is not, the security appliance prevents the user from connecting. If you do not configure group-lock, the security appliance authenticates users without regard to the assigned group.

```
hostname(config-username)# group-lock {value tunnel-grp-name | none}
hostname(config-username)# no group-lock
```

The following example shows how to set group lock for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# group-lock value tunnel group name
```

- Step 12** Specify whether to let users store their login passwords on the client system. Password storage is disabled by default. Enable password storage only on systems that you know to be in secure sites. To disable password storage, enter the **password-storage** command with the **disable** keyword. To remove the password-storage attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for password-storage from the group policy.

```
hostname(config-username)# password-storage {enable | disable}
hostname(config-username)# no password-storage
```

This command has no bearing on interactive hardware client authentication or individual user authentication for hardware clients.

The following example shows how to enable password storage for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# password-storage enable
```

- Step 13** Customize a WebVPN configuration for specific users. Enter webvpn mode by using the **webvpn** command in username configuration mode. The **webvpn** commands for usernames define access to files, MAPI proxy, URLs and TCP applications over WebVPN. They also identify ACLs and types of traffic to filter. WebVPN is disabled by default.

To remove all commands entered in webvpn mode, use the **no** form of this command. These **webvpn** commands apply to the username from which you configure them.

```
hostname(config-username)# webvpn
hostname(config-username)# no webvpn
```

You do not need to configure WebVPN to use e-mail proxies.

WebVPN lets users establish a secure, remote-access VPN tunnel to the security appliance using a web browser. There is no need for either a software or hardware client. WebVPN provides easy access to a broad range of web resources and web-enabled applications from almost any computer that can reach HTTPS Internet sites. WebVPN uses SSL and its successor, TLS1 to provide a secure connection between remote users and specific, supported internal resources that you configure at a central site. The security appliance recognizes connections that need to be proxied, and the HTTP server interacts with the authentication subsystem to authenticate users.

**Note**

The `webvpn` mode that you enter from global configuration mode lets you configure global settings for WebVPN. The `webvpn` mode described in this section, which you enter from username mode, lets you customize a WebVPN configuration for specific users.

In `webvpn` mode, you can customize the following parameters, each of which is described in the subsequent steps:

- `functions url-entry`
- `html-content-filter`
- `homepage`
- `filter`
- `url-list`
- `port-forward`
- `port-forward-name` value Application Access

The following example shows how to enter `webvpn` mode for the username “anyuser” attributes:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)#
```

- Step 14** Configure the WebVPN functions you want to enable. To configure file access and file browsing, HTTP Proxy, MAPI Proxy, and URL entry over WebVPN for this user, enter the **functions** command in `webvpn` mode. To remove a configured function, enter the **no** form of this command. These functions are disabled by default.

To remove all configured functions, including a null value created by issuing the **functions none** command, enter the **no** form of this command without arguments. The **no** option allows inheritance of a value from another group policy. To prevent inheriting function values, enter the **functions none** command.

```
hostname(config-username-webvpn)# functions {file-access | file-browsing | file-entry |
http-proxy | url-entry | mapi | none}

hostname(config-username-webvpn)# no functions [file-access | file-browsing | file-entry |
http-proxy | url-entry | mapi]
```

The keywords used in this command are as follows:

- **file-access**—Enables or disables file access. When enabled, the WebVPN home page lists file servers in the server list. You must enable file access to enable file browsing and/or file entry.
- **file-browsing**—Enables or disables browsing for file servers and shares. You must enable file browsing to allow user entry of a file server.
- **file-entry**—Enables or disables user ability to enter names of file servers.
- **http-proxy**—Enables or disables the forwarding of an HTTP applet proxy to the client. The proxy is useful for technologies that interfere with proper mangling, such as Java, ActiveX, and Flash. It bypasses mangling while ensuring the continued use of the security appliance. The forwarded proxy modifies the browser’s old proxy configuration automatically and redirects all HTTP and HTTPS requests to the new proxy configuration. It supports virtually all client side technologies, including HTML, CSS, JavaScript, VBScript, ActiveX, and Java. The only browser it supports is Microsoft Internet Explorer.

- **mapi**—Enables or disables Microsoft Outlook/Exchange port forwarding.
- **none**—Sets a null value for all WebVPN **functions**. Prevents inheriting functions from a default or specified group policy
- **url-entry**—Enables or disables user entry of URLs. When enabled, the security appliance still restricts URLs with any configured URL or network ACLs. When URL entry is disabled, the security appliance restricts WebVPN users to the URLs on the home page.

The following example shows how to configure file access, file browsing, HTTP Proxy, and MAPI Proxy for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# functions file-access file-browsing MAPI
```

- Step 15** To filter Java, ActiveX, images, scripts, and cookies for WebVPN sessions for this user, enter the **html-content-filter** command in webvpn mode. To remove a content filter, enter the **no** form of this command. To remove all content filters, including a null value created by issuing the **html-content-filter none** command, enter the **no** form of this command without arguments. The **no** option allows inheritance of a value from the group policy. To prevent inheriting an HTML content filter, enter the **html-content-filter none** command. HTML filtering is disabled by default.

Using the command a second time overrides the previous setting.

```
hostname(config-username-webvpn)# html-content-filter {java | images | scripts | cookies | none}

hostname(config-username-webvpn)# no html-content-filter [java | images | scripts | cookies | none]
```

The keywords used in this command are as follows:

- **cookies**—Removes cookies from images, providing limited ad filtering and privacy.
- **images**—Removes references to images (removes tags).
- **java**—Removes references to Java and ActiveX (removes <EMBED>, <APPLET>, and <OBJECT> tags).
- **none**—Indicates that there is no filtering. Sets a null value, thereby disallowing filtering. Prevents inheriting filtering values.
- **scripts**—Removes references to scripting (removes <SCRIPT> tags).

The following example shows how to set filtering of JAVA and ActiveX, cookies, and images for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# html-content-filter java cookies images
```

- Step 16** To specify a URL for the web page that displays upon login for this WebVPN user, enter the **homepage** command in webvpn mode. To remove a configured home page, including a null value created by issuing the **homepage none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. To prevent inheriting a home page, enter the **homepage none** command.

The **none** keyword indicates that there is no WebVPN home page. It sets a null value, thereby disallowing a home page and prevents inheriting a home page.

The *url-string* variable following the keyword **value** provides a URL for the home page. The string must begin with either http:// or https://.

There is no default home page.

```
hostname(config-username-webvpn)# homepage {value url-string | none}
hostname(config-username-webvpn)# no homepage
```

The following example shows how to specify `www.example.com` as the home page for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# homepage value www.example.com
```

- Step 17** To specify the name of the access list to use for WebVPN connections for this user, enter the **filter** command in `webvpn` mode. To remove the access list, including a null value created by issuing the **filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. To prevent inheriting filter values, enter the **filter value none** command.

WebVPN access lists do not apply until you enter the **filter** command to specify them.

You configure ACLs to permit or deny various types of traffic for this user. You then enter the **filter** command to apply those ACLs for WebVPN traffic.

```
hostname(config-username-webvpn)# filter {value ACLname | none}
hostname(config-username-webvpn)# no filter
```

The **none** keyword indicates that there is no **webvpntype** access list. It sets a null value, thereby disallowing an access list and prevents inheriting an access list from another group policy.

The *ACLname* string following the keyword **value** provides the name of the previously configured access list.



Note WebVPN does not use ACLs defined in the **vpn-filter** command.

The following example shows how to set a filter that invokes an access list named `acl_in` for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# filter acl_in
```

- Step 18** To apply a list of WebVPN servers and URLs to a particular user, enter the **url-list** command in `webvpn` mode. To remove a list, including a null value created by using the **url-list none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. To prevent inheriting a url list, enter the **url-list none** command.

```
hostname(config-username-webvpn)# url-list {listname displayname url | none}
hostname(config-username-webvpn)# no url-list
```

The keywords and variables used in this command are as follows:

- *displayname*—Specifies a name for the URL. This name appears on the WebVPN end user interface.
- *listname*—Identifies a name by which to group URLs.
- **none**—Indicates that there is no list of URLs. Sets a null value, thereby disallowing a URL list. Prevents inheriting URL list values.
- *url*—Specifies a URL that WebVPN users can access.

There is no default URL list.

Using the command a second time overrides the previous setting.

Before you can enter the **url-list** command in webvpn mode to identify a URL list that you want to display on the WebVPN home page for a user, you must create the list. Enter the **url-list** command in global configuration mode to create one or more lists.

The following example shows how to set a URL list called “AnyuserURLs” for the user named “anyuser”:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# url-list value AnyuserURLs
```

- Step 19** To enable WebVPN application access for this user, enter the **port-forward** command in webvpn mode. To remove the port forwarding attribute from the configuration, including a null value created by issuing the **port-forward none** command, enter the **no** form of this command. The **no** option allows inheritance of a list from the group policy. To disallow filtering and prevent inheriting a port forwarding list, enter the **port-forward** command with the **none** keyword.

Port forwarding is disabled by default.

```
hostname(config-username-webvpn)# port-forward {value listname | none}
hostname(config-username-webvpn)# no port-forward
```

The *listname* string following the keyword **value** identifies the list of applications WebVPN users can access. Enter the **port-forward** command in configuration mode to define the list.

Using the command a second time overrides the previous setting.

Before you can enter the **port-forward** command in webvpn mode to enable application access, you must define a list of applications that you want users to be able to use in a WebVPN connection. Enter the **port-forward** command in global configuration mode to define this list.

The following example shows how to configure a portforwarding list called “ports1”:

```
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# port-forward value ports1
```

- Step 20** Configure the display name that identifies TCP port forwarding to end users for a particular user by using the **port-forward-name** command in webvpn mode. To delete the display name, including a null value created by using the **port-forward-name none** command, enter the **no** form of the command. The **no** option restores the default name, “Application Access.” To prevent a display name, enter the **port-forward none** command.

```
hostname(config-username-webvpn)# port-forward-name {value name | none}
hostname(config-username-webvpn)# no port-forward-name
```

The following example shows how to configure the port-forward name “test”:

```
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# port-forward-name value test
```