



CHAPTER 39

Configuring Clientless SSL VPN

This chapter describes:

- [Getting Started, page 39-1](#)
- [Creating and Applying Clientless SSL VPN Policies for Accessing Resources, page 39-24](#)
- [Configuring Connection Profile Attributes for Clientless SSL VPN, page 39-25](#)
- [Configuring Group Policy and User Attributes for Clientless SSL VPN, page 39-26](#)
- [Configuring Browser Access to Client-Server Plug-ins, page 39-27](#)
- [Configuring Application Access, page 39-33](#)
- [Configuring File Access, page 39-51](#)
- [Using Clientless SSL VPN with PDAs, page 39-53](#)
- [Using E-Mail over Clientless SSL VPN, page 39-54](#)
- [Optimizing Clientless SSL VPN Performance, page 39-55](#)
- [Clientless SSL VPN End User Setup, page 39-61](#)
- [Capturing Data, page 39-89](#)

Getting Started



Note

When the security appliance is configured for clientless SSL VPN, you cannot enable security contexts (also called firewall multimode) or Active/Active stateful failover. Therefore, these features become unavailable.

Clientless SSL VPN lets users establish a secure, remote-access VPN tunnel to a security appliance using a web browser. Users do not need a software or hardware client.

Clientless SSL VPN provides secure and easy access to a broad range of web resources and web-enabled applications from almost any computer on the Internet. They include:

- Internal websites
- Web-enabled applications
- NT/Active Directory file shares
- E-mail proxies, including POP3S, IMAP4S, and SMTPS

- MS Outlook Web Access
- Application Access (that is, smart tunnel or port forwarding access to other TCP-based applications)

**Note**

The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. Neither the smart tunnel feature nor port forwarding supports MAPI. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.

Clientless SSL VPN uses Secure Sockets Layer Protocol and its successor, Transport Layer Security to provide the 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**

Browser-based VPN access does not save form-based authentication values to permanent local storage.

The network administrator provides access to resources by users of clientless SSL VPN sessions on a group basis. Users have no direct access to resources on the internal network.

The following sections address getting started with the configuration of clientless SSL VPN access:

- [Observing Clientless SSL VPN Security Precautions](#)
- [Understanding Clientless SSL VPN System Requirements](#)
- [Understanding Features Not Supported in Clientless SSL VPN](#)
- [Using SSL to Access the Central Site](#)
- [Authenticating with Digital Certificates](#)
- [Enabling Cookies on Browsers for Clientless SSL VPN](#)
- [Managing Passwords](#)
- [Using Single Sign-on with Clientless SSL VPN](#)

Observing Clientless SSL VPN Security Precautions

Clientless SSL VPN connections on the security appliance differ from remote access IPSec connections, particularly with respect to how they interact with SSL-enabled servers, and precautions to follow to reduce security risks.

In a clientless SSL VPN connection, the security appliance acts as a proxy between the end user web browser and target web servers. When a user connects to an SSL-enabled web server, the security appliance establishes a secure connection and validates the server SSL certificate. The end user browser never receives the presented certificate, so therefore cannot examine and validate the certificate.

The current implementation of clientless SSL VPN on the security appliance does not permit communication with sites that present expired certificates. Nor does the security appliance perform trusted CA certificate validation. Therefore, users cannot analyze the certificate an SSL-enabled web-server presents before using with the web-enabled service.

We recommend that you do the following to minimize risks posed by clientless SSL VPN access:

-
- Step 1** Configure a group policy for all users who need clientless SSL VPN access, and enable clientless SSL VPN only for that group policy.

- Step 2** Apply ACLs to permit access only to specific targets within the private network, permit access only to the private network, deny Internet access, or permit access only to reputable sites.
- Step 3** Disable URL entry on the *portal page*, the page that opens when the user establishes a browser-based connection, to help prevent user access confusion. To do so, go to the **Configuration > Remote Access VPN > Clientless SSL VPN Access > Dynamic Access Policies** window, select the applicable DAP, click the **Edit > Functions** tab, and click **Disable** next to **URL entry**.
- Step 4** Instruct users to enter external URLs in the native browser address field above the portal page or open a separate browser window to visit external sites.

**Caution**

Users must not use the portal page (including the address field inside the portal page if you do not follow Step 3), to visit external sites that display the https prefix (for example, online banking sites). The security appliance does not verify external https certificates.

Understanding Clientless SSL VPN System Requirements

Clientless SSL VPN supports access from the following OSs and browsers.

OSs	Browser and Java Versions	Feature Notes ¹
Windows Vista SP2 Vista SP1 with KB952876 or later.	Microsoft Internet Explorer 7 Firefox 2.0 or later.	Windows Vista does not support Windows Shares (CIFS) Web Folders. Additional requirements and limitations apply to smart tunnel and port forwarding .
Windows XP SP2 or later.	Microsoft Internet Explorer 7 and 6 Firefox 2.0 or later.	Windows XP SP2 or later requires Microsoft KB892211 hotfix to support Web Folders. Additional requirements and limitations apply to smart tunnel and port forwarding .
Windows 2000 SP4.	Microsoft Internet Explorer 7 and 6 Firefox 2.0 or later.	Windows Vista does not support Windows Shares (CIFS) Web Folders. Windows 2000 SP4 requires Microsoft KB892211 hotfix to support Web Folders. Additional requirements and limitations apply to smart tunnel and port forwarding .
Apple: Mac OS X 10.4 and 10.5	Safari 2.0 or later, or Firefox 2.0 or later.	Certificate authentication, including the DoD Common Access Card and SmartCard, works with the Safari keychain only. Web folders do not support Mac OS. Additional requirements and limitations apply to smart tunnel and port forwarding .
Linux	Firefox 2.0 or later.	Web folders and smart tunnel do not support Linux. Additional requirements apply to port forwarding .

1. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.

ActiveX pages require that you use the ActiveX Relay default setting (Enable) on the associated group policy. If you do so or assign a smart tunnel list to the policy, and the browser proxy exception list on the endpoint specifies a proxy, the user must add a “shutdown.webvpn.relay.” entry to that list.

Clientless SSL VPN access does not support Windows Shares (CIFS) Web Folders on Windows 7, Vista, Internet Explorer 8, Mac OS, and Linux. Windows XP SP2 requires a [Microsoft hotfix](#) to support Web Folders.

See the following sections for the platforms supported by the clientless applications named:

- [Port Forwarding Requirements and Restrictions, page 39-43](#)
- [Smart Tunnel Requirements, Restrictions, and Limitations, page 39-34](#)
- [Plug-in Requirements and Restrictions, page 39-28](#)

Understanding Features Not Supported in Clientless SSL VPN

The security appliance does not support the following features for clientless SSL VPN connections:

- DSA certificates; The ASA does support RSA certificates.
- Remote HTTPS certificates.
- Requirements of some domain-based security products. Because the adaptive security appliance encodes the URL, requests actually originate from the ASA, which in some cases do not satisfy the requirements of domain-based security products.
- Inspection features under the Modular Policy Framework, inspecting configuration control.
- Functionality the filter configuration commands provide, including the **vpn-filter** command.
- VPN connections from hosts with IPv6 addresses. Hosts must use IPv4 addresses to establish clientless SSL VPN or AnyConnect sessions. However, beginning with ASA 8.0(2), users can use these sessions to access internal IPv6-enabled resources.
- NAT, reducing the need for globally unique IP addresses.
- PAT, permitting multiple outbound sessions appear to originate from a single IP address.
- QoS, rate limiting using the **police** command and **priority-queue** command.
- Connection limits, checking either via the static or the Modular Policy Framework **set connection** command.
- The **established** command, allowing return connections from a lower security host to a higher security host if there is already an established connection from the higher level host to the lower level host.

Using SSL to Access the Central Site

Clientless SSL VPN uses SSL and its successor, TLS1 to provide a secure connection between remote users and specific, supported internal resources at a central site. This section includes the following topics:

- [Using HTTPS for Clientless SSL VPN Sessions](#)
- [Configuring Clientless SSL VPN and ASDM Ports](#)
- [Configuring Support for Proxy Servers](#)
- [Configuring SSL/TLS Encryption Protocols](#)

Using HTTPS for Clientless SSL VPN Sessions

Establishing clientless SSL VPN sessions requires the following:

- Enabling clientless SSL VPN sessions on the security appliance interface that users connect to.
- Using HTTPS to access the security appliance or load balancing cluster. In a web browser, users enter the security appliance IP address in the format *https:// address* where *address* is the IP address or DNS hostname of the security appliance interface.

To permit clientless SSL VPN sessions on an interface, perform the following steps:

-
- Step 1** In global configuration mode, enter the **webvpn** command to enter webvpn mode.
- Step 2** Enter the **enable** command with the name of the interface that you want to use for clientless SSL VPN sessions.

For example, to enable clientless SSL VPN sessions on the interface called outside, enter the following:

```
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
```

Configuring Clientless SSL VPN and ASDM Ports

Beginning with Version 8.0(2), the security appliance supports both clientless SSL VPN sessions and ASDM administrative sessions simultaneously on Port 443 of the outside interface. You do, however, have the option to configure these applications on different interfaces.

To change the SSL listening port for clientless SSL VPN, use the **port port_number** command in webvpn mode. The following example enables clientless SSL VPN on port 444 of the outside interface. HTTPS for ASDM is also configured on the outside interface and uses the default port (443). With this configuration, remote users initiating clientless SSL VPN sessions enter *https://<outside_ip>:444* in the browser.

```
hostname(config)# http server enable
hostname(config)# http 192.168.3.0 255.255.255.0 outside
hostname(config)# webvpn
hostname(config-webvpn)# port 444
hostname(config-webvpn)# enable outside
```

To change the listening port for ASDM, use the *port* argument of the **http server enable** command in privileged EXEC mode. The following example specifies that HTTPS ASDM sessions use port 444 on the outside interface. Clientless SSL VPN is also enabled on the outside interface and uses the default port (443). With this configuration, remote users initiate ASDM sessions by entering *https://<outside_ip>:444* in the browser.

```
hostname(config)# http server enable 444
hostname(config)# http 192.168.3.0 255.255.255.0 outside
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
```

Configuring Support for Proxy Servers

The security appliance can terminate HTTPS connections and forward HTTP and HTTPS requests to proxy servers. These servers act as intermediaries between users and the Internet. Requiring Internet access via a server that the organization controls provides another opportunity for filtering to assure secure Internet access and administrative control.

When configuring support for HTTP and HTTPS proxy services, you can assign preset credentials to send with each request for basic authentication. You can also specify URLs to exclude from HTTP and HTTPS requests.

You can specify a proxy autoconfiguration (PAC) file to download from an HTTP proxy server, however, you may not use proxy authentication when specifying the PAC file.

To configure the security appliance to use an external proxy server to handle HTTP and HTTPS requests, use the **http-proxy** and **https-proxy** commands in webvpn mode.

- **http-proxy** *host* [*port*] [**exclude** *url*] [**username** *username* {**password** *password*}]
- **https-proxy** *host* [*port*] [**exclude** *url*] [**username** *username* {**password** *password*}]
- **http-proxy** **pac** *url*

exclude—(Optional) Enter this keyword to exclude URLs from those that can be sent to the proxy server.

host—Enter the hostname or IP address for the external proxy server.

pac—Proxy autoconfiguration file to download to the browser. Once downloaded, the PAC file uses a JavaScript function to identify a proxy for each URL.

password—(Optional, and available only if you specify a *username*) Enter this keyword to accompany each proxy request with a password to provide basic, proxy authentication.

password—Enter the password to send to the proxy server with each HTTP or HTTPS request.

port—(Optional) Enter the port number used by the proxy server. The default HTTP port is 80. The default HTTPS port is 443. The security appliance uses each of these ports if you do not specify an alternative value. The range is 1-65535.

url—If you entered **exclude**, enter a URL or a comma-delimited list of several URLs to exclude from those that can be sent to the proxy server. The string does not have a character limit, but the entire command cannot exceed 512 characters. You can specify literal URLs or use the following wildcards:

- * to match any string, including slashes (/) and periods (.). You must accompany this wildcard with an alphanumeric string.
- ? to match any single character, including slashes and periods.
- [x-y] to match any single character in the range of x and y, where x represents one character and y represents another character in the ANSI character set.
- [!x-y] to match any single character that is not in the range.

If you entered **http-proxy pac**, follow it with **http://** and type the URL of the proxy autoconfiguration file. If you omit the **http://** portion, the CLI ignores the command.

username—(Optional) Enter this keyword to accompany each HTTP proxy request with a username for basic, proxy authentication. Only the **http-proxy host** command supports this keyword.

username—Enter the username the password to send to the proxy server with each HTTP or HTTPS request.

The security appliance clientless SSL VPN configuration supports only one **http-proxy** and one **https-proxy** command each. For example, if one instance of the **http-proxy** command is already present in the running configuration and you enter another, the CLI overwrites the previous instance.

The following example shows how to configure use of an HTTP proxy server with an IP address of 209.165.201.1 using the default port, send a username and password with each HTTP request:

```
hostname(config-webvpn)# http-proxy 209.165.201.1 user jsmith password mysecretdonttell
hostname(config-webvpn)
```

The following example shows the same command, except when the security appliance receives the specific URL www.example.com in an HTTP request, it resolves the request instead of passing it on to the proxy server:

```
hostname(config-webvpn)# http-proxy 209.165.201.1 exclude www.example.com username jsmith
password mysecretdonttell
hostname(config-webvpn)
```

The following example shows how to specify a URL to serve a proxy autoconfiguration file to the browser:

```
hostname(config-webvpn)# http-proxy pac http://www.example.com/pac
hostname(config-webvpn)
```

Configuring SSL/TLS Encryption Protocols

When you set SSL/TLS encryption protocols, be aware of the following:

- Make sure that the security appliance and the browser you use allow the same SSL/TLS encryption protocols.
- If you configure e-mail proxy, do not set the security appliance SSL version to TLSv1 Only. Microsoft Outlook and Microsoft Outlook Express do not support TLS.
- TCP Port Forwarding requires Sun Microsystems Java Runtime Environment (JRE) version 1.4.x and 1.5.x. Port forwarding does not work when a user of clientless SSL VPN connects with some SSL versions, as follows:

Negotiate SSLv3	Java downloads
Negotiate SSLv3/TLSv1	Java downloads
Negotiate TLSv1	Java does NOT download
TLSv1Only	Java does NOT download
SSLv3Only	Java does NOT download

Authenticating with Digital Certificates

SSL uses digital certificates for authentication. The security appliance creates a self-signed SSL server certificate when it boots; or you can install in the security appliance an SSL certificate that has been issued in a PKI context. For HTTPS, this certificate must then be installed on the client. You need to install the certificate from a given security appliance only once.

Restrictions for authenticating users with digital certificates include the following:

- Application Access does not work for users of clientless SSL VPN who authenticate using digital certificates. JRE does not have the ability to access the web browser keystore. Therefore JAVA cannot use a certificate that the browser uses to authenticate a user, so it cannot start.
- E-mail clients such as MS Outlook, MS Outlook Express, and Eudora lack the ability to access the certificate store.

For more information on authentication and authorization using digital certificates, see “[Using Certificates and User Login Credentials](#)” in the “[Configuring AAA Servers and the Local Database](#)” chapter.

Enabling Cookies on Browsers for Clientless SSL VPN

Browser cookies are required for the proper operation of clientless SSL VPN. When cookies are disabled on the web browser, the links from the web portal home page open a new window prompting the user to log in once more.

Managing Passwords

Optionally, you can configure the security appliance to warn end users when their passwords are about to expire. To do this, you specify the **password-management** command in tunnel-group general-attributes mode or enable the feature using ASDM at Configuration > Remote Access VPN > Clientless SSL VPN Access > Connection Profiles > Add or Edit > Advanced > General > Password Management.

The security appliance supports password management for the RADIUS and LDAP protocols. It supports the “password-expire-in-days” option for LDAP only.

You can configure password management for IPsec remote access and SSL VPN tunnel-groups.

When you configure password management, the security appliance notifies the remote user at login that the user’s current password is about to expire or has expired. The security appliance then offers the user the opportunity to change the password. If the current password has not yet expired, the user can still log in using that password.

This command is valid for AAA servers that support such notification. The security appliance ignores this command if RADIUS or LDAP authentication has not been configured.



Note

Some RADIUS servers that support MSCHAP currently do not support MSCHAPv2. This command requires MSCHAPv2 so please check with your vendor.

The security appliance, releases 7.1 and later, generally supports password management for the following connection types when authenticating with LDAP or with any RADIUS configuration that supports MS-CHAPv2:

- AnyConnect VPN Client
- IPsec VPN Client
- Clientless SSL VPN

Password management is *not* supported for any of these connection types for Kerberos/Active Directory (Windows password) or NT 4.0 Domain.

The RADIUS server (for example, Cisco ACS) could proxy the authentication request to another authentication server. However, from the security appliance perspective, it is talking only to a RADIUS server.

**Note**

For LDAP, the method to change a password is proprietary for the different LDAP servers on the market. Currently, the security appliance implements the proprietary password management logic only for Microsoft Active Directory and Sun LDAP servers.

Native LDAP requires an SSL connection. You must enable LDAP over SSL before attempting to do password management for LDAP. By default, LDAP uses port 636.

**Note**

If you are using an LDAP directory server for authentication, password management is supported with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server) and the Microsoft Active Directory.

Sun—The DN configured on the security appliance to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACI on the default password policy.

Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.

Note that this command does not change the number of days before the password expires, but rather, the number of days ahead of expiration that the security appliance starts warning the user that the password is about to expire.

If you do specify the **password-expire-in-days** keyword, you must also specify the number of days.

Specifying this command with the number of days set to 0 disables this command. The security appliance does not notify the user of the pending expiration, but the user can change the password after it expires.

The following example sets the days before password expiration to begin warning the user of the pending expiration to 90 for the connection profile “testgroup”:

```
hostname(config)# tunnel-group testgroup type webvpn
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-general)# password-management password-expire-in-days 90
```

Using Single Sign-on with Clientless SSL VPN

Single sign-on support lets users of clientless SSL VPN enter a username and password only once to access multiple protected services and web servers. In general, the SSO mechanism either starts as part of the AAA process or just after successful user authentication to a AAA server. The clientless SSL VPN server running on the security appliance acts as a proxy for the user to the authenticating server. When a user logs in, the clientless SSL VPN server sends an SSO authentication request, including username and password, to the authenticating server using HTTPS. If the server approves the authentication request, it returns an SSO authentication cookie to the clientless SSL VPN server. The security appliance keeps this cookie on behalf of the user and uses it to authenticate the user to secure websites within the domain protected by the SSO server.

This section describes the three SSO authentication methods supported by clientless SSL VPN: HTTP Basic and NTLMv1 (NT LAN Manager) authentication, the Computer Associates eTrust SiteMinder SSO server (formerly Netegrity SiteMinder), and Version 1.1 of Security Assertion Markup Language (SAML), the POST-type SSO server authentication.

This section includes:

- [Configuring SSO with HTTP Basic or NTLM Authentication](#)
- [Configuring SSO Authentication Using SiteMinder](#)
- [Configuring SSO Authentication Using SAML Browser Post Profile](#)
- [Configuring SSO with the HTTP Form Protocol](#)
- [Configuring SSO with Macro Substitution](#)

Configuring SSO with HTTP Basic or NTLM Authentication

This section describes single sign-on with HTTP Basic or NTLM authentication. You can configure the security appliance to implement SSO using either or both of these methods. The **auto-signon** command configures the security appliance to automatically pass clientless SSL VPN user login credentials (username and password) on to internal servers. You can enter multiple **auto-signon** commands. The security appliance processes them according to the input order (early commands take precedence). You specify the servers to receive the login credentials using either IP address and IP mask, or URI mask.

Use the **auto-signon** command in any of three modes: webvpn configuration, webvpn group-policy mode, or webvpn username mode. Username supersedes group, and group supersedes global. The mode you choose depends upon scope of authentication you want:

Mode	Scope
webvpn configuration	All clientless SSL VPN users globally
webvpn group-policy configuration	A subset of clientless SSL VPN users defined by a group policy
webvpn username configuration	An individual user of clientless SSL VPN

The following example commands present various possible combinations of modes and arguments.

All Users, IP Address Range, NTLM

To configure **auto-signon** for all users of clientless SSL VPN to servers with IP addresses ranging from 10.1.1.0 to 10.1.1.255 using NTLM authentication, for example, enter the following commands:

```
hostname(config)# webvpn
hostname(config-webvpn)# auto-signon allow ip 10.1.1.1 255.255.255.0 auth-type ntlm
```

All Users, URI Range, HTTP Basic

To configure **auto-signon** for all users of clientless SSL VPN, using basic HTTP authentication, to servers defined by the URI mask `https://*.example.com/*`, for example, enter the following commands:

```
hostname(config)# webvpn
hostname(config-webvpn)# auto-signon allow uri https://*.example.com/* auth-type basic
```

Group, URI Range, HTTP Basic and NTLM

To configure **auto-signon** for clientless SSL VPN sessions associated with the ExamplePolicy group policy, using either basic or NTLM authentication, to servers defined by the URI mask `https://*.example.com/*`, for example, enter the following commands:

```
hostname(config)# group-policy ExamplePolicy attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# auto-signon allow uri https://*.example.com/* auth-type all
```

Specific User, IP Address Range, HTTP Basic

To configure **auto-signon** for a user named Anyuser to servers with IP addresses ranging from 10.1.1.0 to 10.1.1.255 using HTTP Basic authentication, for example, enter the following commands:

```
hostname(config)# username Anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# auto-signon allow ip 10.1.1.1 255.255.255.0 auth-type basic
```

Configuring SSO Authentication Using SiteMinder

This section describes configuring the security appliance to support SSO with SiteMinder. You would typically choose to implement SSO with SiteMinder if your website security infrastructure already incorporates SiteMinder. With this method, SSO authentication is separate from AAA and happens once the AAA process completes. If you want to configure SSO for a user or group for clientless SSL VPN access, you must first configure a AAA server, such as a RADIUS or LDAP server. You can then set up SSO support for clientless SSL VPN. This section includes:

- [Task Overview: Configuring SSO with SiteMinder](#)
- [Detailed Tasks: Configuring SSO with SiteMinder](#)
- [Adding the Cisco Authentication Scheme to SiteMinder](#)

Task Overview: Configuring SSO with SiteMinder

This section presents an overview of the tasks necessary to configure SSO with SiteMinder SSO. These tasks are:

- Specifying the SSO server.
- Specifying the URL of the SSO server to which the security appliance makes SSO authentication requests.
- Specifying a secret key to secure the communication between the security appliance and the SSO server. This key is similar to a password: you create it, save it, and enter it on both the security appliance and the SiteMinder Policy Server using the Cisco Java plug-in authentication scheme.

Optionally, you can do the following configuration tasks in addition to the required tasks:

- Configuring the authentication request timeout.
- Configuring the number of authentication request retries.

After you complete these tasks, assign an SSO server to a user or group policy.

Detailed Tasks: Configuring SSO with SiteMinder

This section presents specific steps for configuring the security appliance to support SSO authentication with CA SiteMinder. To configure SSO with SiteMinder, perform the following steps:

-
- Step 1** In webvpn configuration mode, enter the **sso-server** command with the **type** option to create an SSO server. For example, to create an SSO server named Example of type siteminder, enter the following:
- ```
hostname(config)# webvpn
hostname(config-webvpn)# sso-server Example type siteminder
hostname(config-webvpn-sso-siteminder)#
```
- Step 2** Enter the **web-agent-url** command in webvpn-sso-siteminder configuration mode to specify the authentication URL of the SSO server. For example, to send authentication requests to the URL <http://www.Example.com/webvpn>, enter the following:
- ```
hostname(config-webvpn-sso-siteminder)# web-agent-url http://www.Example.com/webvpn
hostname(config-webvpn-sso-siteminder)#
```
- Step 3** Specify a secret key to secure the authentication communications between the security appliance and SiteMinder using the **policy-server-secret** command in webvpn-sso-siteminder configuration mode. You can create a key of any length using any regular or shifted alphanumeric character, but you must enter the same key on both the security appliance and the SSO server.
- For example, to create the secret key Atal8rD8!, enter the following:
- ```
hostname(config-webvpn-sso-siteminder)# policy-server-secret Atal8rD8!
hostname(config-webvpn-sso-siteminder)#
```
- Step 4** Optionally, you can configure the number of seconds before a failed SSO authentication attempt times out using the **request-timeout** command in webvpn-sso-siteminder configuration mode. The default number of seconds is 5 seconds and the possible range is 1 to 30 seconds. To change the number of seconds before a request times out to 8, for example, enter the following:
- ```
hostname(config-webvpn-sso-siteminder)# request-timeout 8
hostname(config-webvpn-sso-siteminder)#
```
- Step 5** Optionally, you can configure the number of times the security appliance retries a failed SSO authentication attempt before the authentication times-out using the **max-retry-attempts** command in webvpn-sso-siteminder configuration mode. The default is 3 retry attempts and the possible range is 1 to 5 attempts. To configure the number of retries to be 4, for example, enter the following:
- ```
hostname(config-webvpn-sso-siteminder)# max-retry-attempts 4
hostname(config-webvpn-sso-siteminder)#
```
- Step 6** After you configure the SSO server, you must specify SSO authentication for either a group or user. To specify SSO for a group, assign an SSO server to a group policy using the **sso-server value** command in group-policy-webvpn configuration mode. To specify SSO for a user, assign an SSO server to a user policy using the same command, **sso-server value**, but in username-webvpn configuration mode. For example, to assign the SSO server named Example to the user named Anyuser, enter the following:
- ```
hostname(config)# username Anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# sso-server value Example
hostname(config-username-webvpn)#
```
- Step 7** Finally, you can test the SSO server configuration using the **test sso-server** command in privileged EXEC mode. For example, to test the SSO server named Example using the username Anyuser, enter the following:
- ```
hostname# test sso-server Example username Anyuser
```

```
INFO: Attempting authentication request to sso-server Example for user Anyuser
INFO: STATUS: Success
hostname#
```

## Adding the Cisco Authentication Scheme to SiteMinder

In addition to configuring the security appliance for SSO with SiteMinder, you must also configure your CA SiteMinder Policy Server with the Cisco authentication scheme, a Java plug-in you download from the Cisco web site.



### Note

Configuring the SiteMinder Policy Server requires experience with SiteMinder. This section presents general tasks, not a complete procedure.

To configure the Cisco authentication scheme on your SiteMinder Policy Server, perform the following tasks:

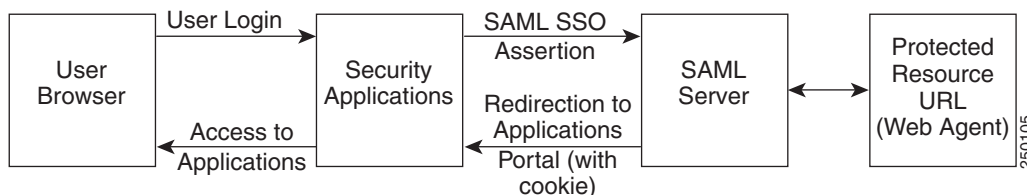
- Step 1** With the SiteMinder Administration utility, create a custom authentication scheme, being sure to use the following specific arguments:
- In the Library field, enter **smjavaapi**.
  - In the Secret field, enter the same secret configured on the security appliance.  
You configure the secret on the security appliance using the **policy-server-secret** command at the command line interface.
  - In the Parameter field, enter **CiscoAuthApi**.
- Step 2** Using your Cisco.com login, download the file **cisco\_vpn\_auth.jar** from <http://www.cisco.com/cisco/software/navigator.html> and copy it to the default library directory for the SiteMinder server. This .jar file is also available on the Cisco security appliance CD.

## Configuring SSO Authentication Using SAML Browser Post Profile

This section describes configuring the security appliance to support Security Assertion Markup Language (SAML), Version 1.1 POST profile Single Sign-On (SSO) for authorized users. SAML SSO is supported only for clientless SSL VPN sessions. This section includes:

- [Task Overview: Configuring SSO with SAML Post Profile](#)
- [Detailed Tasks: Configuring SSO with SAML Post Profile](#)
- [SSO Server Configuration](#)

After a session is initiated, the security appliance authenticates the user against a configured AAA method. Next, the security appliance (the asserting party) generates an assertion to the relying party, the consumer URL service provided by the SAML server. If the SAML exchange succeeds, the user is allowed access to the protected resource. [Figure 39-1](#) shows the communication flow:

**Figure 39-1 SAML Communication Flow****Note**

The SAML Browser Artifact method of exchanging assertions is not supported.

**Task Overview: Configuring SSO with SAML Post Profile**

This section presents an overview of the tasks necessary to configure SSO with SAML Browser Post Profile. These tasks are:

- Specify the SSO server with the **sso-server** command.
- Specify the URL of the SSO server for authentication requests (the **assertion-consumer-url** command)
- Specify the security appliance hostname as the component issuing the authentication request (the **issuer** command)
- Specify the trustpoint certificates use for signing SAML Post Profile assertions (the **trustpoint** command)

Optionally, in addition to these required tasks, you can do the following configuration tasks:

- Configure the authentication request timeout (the **request-timeout** command)
- Configure the number of authentication request retries (the **max-retry-attempts** command)

After completing the configuration tasks, you assign an SSO server to a user or group policy.

**Detailed Tasks: Configuring SSO with SAML Post Profile**

This section presents specific steps for configuring the security appliance to support SSO authentication with SAML Post Profile. To configure SSO with SAML-V1.1-POST, perform the following steps:

- Step 1** In webvpn configuration mode, enter the **sso-server** command with the **type** option to create an SSO server. For example, to create an SSO server named Sample of type SAML-V1.1-POST, enter the following:

```
hostname(config)# webvpn
hostname(config-webvpn)# sso-server sample type SAML-V1.1-post
hostname(config-webvpn-sso-saml)#
```

**Note**

The security appliance currently supports only the Browser Post Profile type of SAML SSO Server.

- Step 2** Enter the **assertion-consumer-url** command in webvpn-sso-saml configuration mode to specify the authentication URL of the SSO server. For example, to send authentication requests to the URL <http://www.Example.com/webvpn>, enter the following:

```
hostname(config-webvpn-sso-saml)# assertion-consumer-url http://www.sample.com/webvpn
hostname(config-webvpn-sso-saml)#
```

- Step 3** Specify a unique string that identifies the security appliance itself when it generates assertions. Typically, this issuer name is the hostname for the security appliance as follows:

```
hostname(config-webvpn-sso-saml) # issuer myasa
hostname(config-webvpn-sso-saml) #
```

- Step 4** Specify the identification certificate for signing the assertion with the **trust-point** command. An example follows:

```
hostname(config) # tunnel-group 209.165.200.225 type IPSec_L2L
hostname(config) # tunnel-group 209.165.200.225 ipsec-attributes
hostname(config-tunnel-ipsec) # trust-point mytrustpoint
```

Optionally, you can configure the number of seconds before a failed SSO authentication attempt times out using the **request-timeout** command in webvpn-sso-saml configuration mode. The default number of seconds is 5 seconds and the possible range is 1 to 30 seconds. To change the number of seconds before a request times out to 8, for example, enter the following:

```
hostname(config-webvpn-sso-saml) # request-timeout 8
hostname(config-webvpn-sso-saml) #
```

- Step 5** Optionally, you can configure the number of times the security appliance retries a failed SSO authentication attempt before the authentication times-out using the **max-retry-attempts** command in webvpn-sso-saml configuration mode. The default is 3 retry attempts and the possible range is 1 to 5 attempts. To configure the number of retries to be 4, for example, enter the following:

```
hostname(config-webvpn-sso-saml) # max-retry-attempts 4
hostname(config-webvpn-sso-saml) #
```

- Step 6** After you configure the SSO server, you must specify SSO authentication for either a group or user. To specify SSO for a group, assign an SSO server to a group policy using the **sso-server value** command in group-policy-webvpn configuration mode. To specify SSO for a user, assign an SSO server to a user policy using the same command, **sso-server value**, but in username-webvpn configuration mode. For example, to assign the SSO server named Example to the user named Anyuser, enter the following:

```
hostname(config) # username Anyuser attributes
hostname(config-username) # webvpn
hostname(config-username-webvpn) # sso-server value sample
hostname(config-username-webvpn) #
```

- Step 7** Finally, you can test the SSO server configuration using the **test sso-server** command in privileged EXEC mode. For example, to test the SSO server, Example using the username Anyuser, enter:

```
hostname# test sso-server Example username Anyuser
INFO: Attempting authentication request to sso-server sample for user Anyuser
INFO: STATUS: Success
```

---

## SSO Server Configuration

Use the SAML server documentation provided by the server software vendor to configure the SAML server in Relying Party mode. The following steps list the specific parameters required to configure the SAML Server for Browser Post Profile:

- 
- Step 1** Configure the SAML server parameters to represent the asserting party (the security appliance):
- Recipient consumer url (same as the assertion consumer url configured on the ASA)
  - Issuer ID, a string, usually the hostname of appliance
  - Profile type -Browser Post Profile
- Step 2** Configure certificates.
- Step 3** Specify that asserting party assertions must be signed.
- Step 4** Select how the SAML server identifies the user:
- Subject Name Type is DN
  - Subject Name format is uid=<user>
- 

## Configuring SSO with the HTTP Form Protocol

**Note**

To configure SSO with the HTTP protocol correctly, you must have a thorough working knowledge of authentication and HTTP protocol exchanges.

This section describes using the HTTP Form protocol for SSO. HTTP Form protocol enables SSO authentication that can also qualify as a AAA method. It provides a secure method for exchanging authentication information between users of clientless SSL VPN and authenticating web servers. As a common protocol, it is applicable with web servers and web-based SSO products if the following conditions are met:

- The web form must not have a dynamic parameter that is relevant for authentication (such as parameters set by Javascript or unique for each request).
- The authentication cookie must be set for successful request and not set for unauthorized logons. In this case, ASA cannot distinguish between successful and failed authentication.

You can use it in conjunction with other AAA servers such as RADIUS or LDAP servers.

**Note**

It is important to remember that HTTP Form authentication can be used in conjunction with RADIUS or LDAP authorization, but *not* with RADIUS or LDAP authentication.

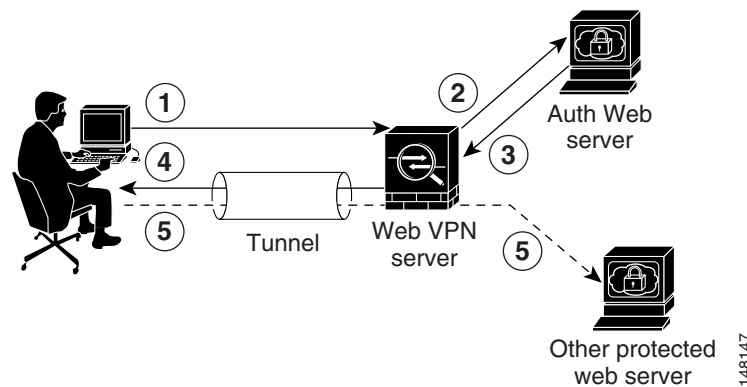
The security appliance again serves as a proxy for users of clientless SSL VPN to an authenticating web server but, in this case, it uses HTTP Form protocol and the POST method for requests. You must configure the security appliance to send and receive form data. [Figure 39-2](#) illustrates the following SSO authentication steps:

1. A user of clientless SSL VPN first enters a username and password to log into the clientless SSL VPN server on the security appliance.



2. The clientless SSL VPN server acts as a proxy for the user and forwards the form data (username and password) to an authenticating web server using a POST authentication request.
3. If the authenticating web server approves the user data, it returns an authentication cookie to the clientless SSL VPN server where it is stored on behalf of the user.
4. The clientless SSL VPN server establishes a tunnel to the user.
5. The user can now access other websites within the protected SSO environment without reentering a username and password.

**Figure 39-2 SSO Authentication Using HTTP Forms**



While you would expect to configure form parameters that let the security appliance include POST data such as the username and password, you initially might not be aware of additional hidden parameters that the web server requires. Some authentication applications expect hidden data which is neither visible to nor entered by the user. You can, however, discover hidden parameters the authenticating web server expects by making a direct authentication request to the web server from your browser without the security appliance in the middle acting as a proxy. Analyzing the web server response using an HTTP header analyzer reveals hidden parameters in a format similar to the following:

```
<param name>=<URL encoded value>&<param name>=<URL encoded>
```

Some hidden parameters are mandatory and some are optional. If the web server requires data for a hidden parameter, it rejects any authentication POST request that omits that data. Because a header analyzer does not tell you if a hidden parameter is mandatory or not, we recommend that you include all hidden parameters until you determine which are mandatory.

This section describes:

- [Gathering HTTP Form Data](#)
- [Task Overview: Configuring SSO with HTTP Form Protocol](#)
- [Detailed Tasks: Configuring SSO with HTTP Form Protocol](#)

## Gathering HTTP Form Data

This section presents the steps for discovering and gathering necessary HTTP Form data. If you do not know what parameters the authenticating web server requires, you can gather parameter data by analyzing an authentication exchange using the following steps:

**Note**

These steps require a browser and an HTTP header analyzer.

- Step 1** Start your browser and HTTP header analyzer, and connect directly to the web server login page without going through the security appliance.
- Step 2** After the web server login page has loaded in your browser, examine the login sequence to determine if a cookie is being set during the exchange. If the web server has loaded a cookie with the login page, configure this login page URL as the *start-URL*.
- Step 3** Enter the username and password to log in to the web server, and press Enter. This action generates the authentication POST request that you examine using the HTTP header analyzer.

An example POST request—with host HTTP header and body—follows:

```
POST
/emco/myemco/authc/forms/MCOlogin.fcc?TYPE=33554433&REALMOID=06-000430e1-7443-125c-ac05
-83846dc90034&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=SM5FZmjnk3DRNwNjk2KcqVCFbIr
NT9%2bJ0H0KPshFtg6rB1UV2PxxHqLw%3d%3d&TARGET=https%3A%2F%2Fwww.example.com%2Femco%2Fmye
mco%2FHTTP/1.1

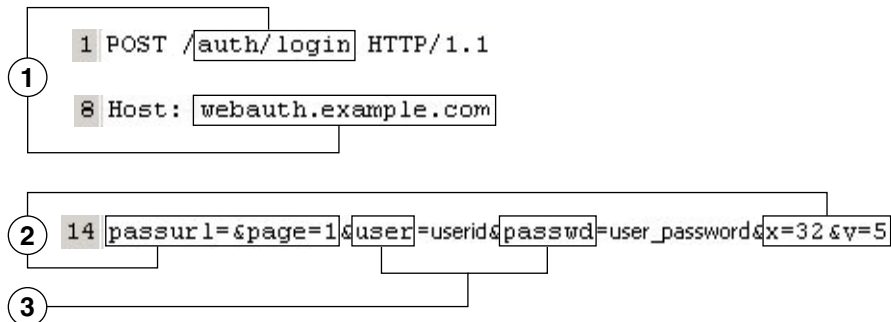
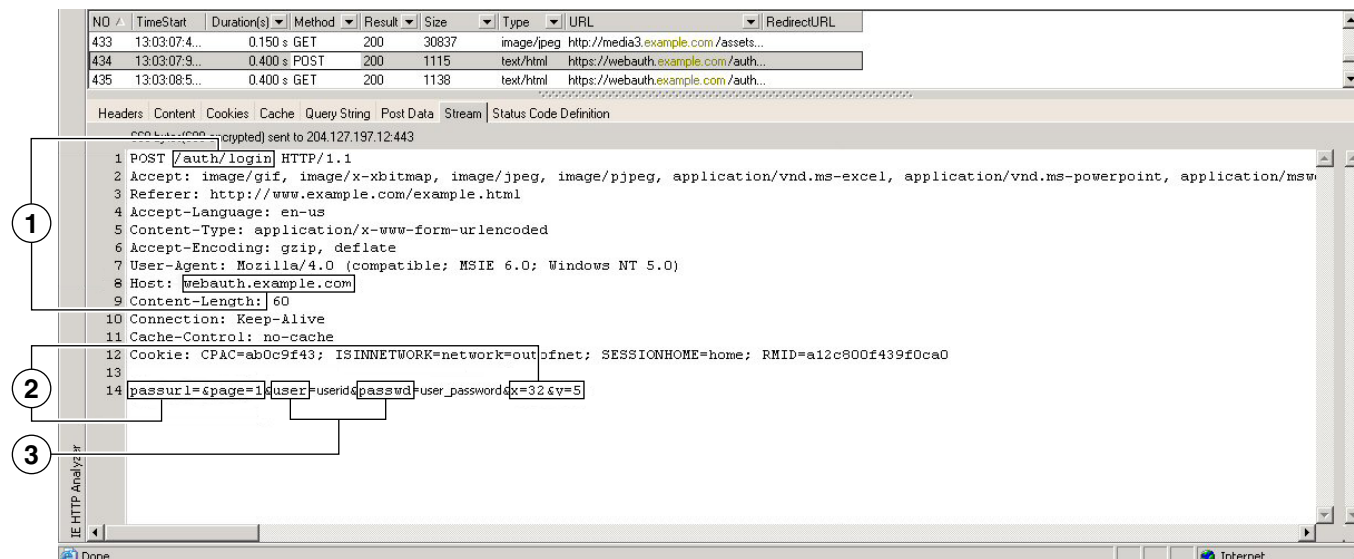
Host: www.example.com

(BODY)

SMENC=ISO-8859-1&SMLOCALE=US-EN&USERID=Anyuser&USER_PASSWORD=XXXXXX&target=https%3A%2F%
2Fwww.example.com%2Femco%2Fmyemco%2F&smauthreason=0
```

- Step 4** Examine the POST request and copy the protocol, host, and the complete URL to configure the action-uri parameter.
- Step 5** Examine the POST request body and copy the following:
- Username parameter. In the preceding example, this parameter is USERID, not the value anyuser.
  - Password parameter. In the preceding example, this parameter is USER\_PASSWORD.
  - Hidden parameter. This parameter is everything in the POST body except the username and password parameters. In the preceding example, the hidden parameter is:  
SMENC=ISO-8859-1&SMLOCALE=US-EN&target=https%3A%2F%2Fwww.example.com%2Femco%2Fmyemco%2F&smauthreason=0

Figure 39-3 highlights the action URI, hidden, username and password parameters within sample output from an HTTP analyzer. This is only an example; output varies widely across different websites.

**Figure 39-3** Action-uri, hidden, username and password parameters

1	Action URI parameter
2	Hidden parameters
3	Username and password parameters

**Step 6** If you successfully log in to the web server, examine the server response with the HTTP header analyzer to locate the name of the session cookie set by the server in your browser. This is the **auth-cookie-name** parameter.

In the following server response header, the name of the session cookie is SMSESSION. You just need the name, not the value.

249533

Set-Cookie:

```
SMSESSION=yN4Yp5hHVNDgs4FT8dn7+Rwev41hsE49XlKc+ltwie0ggnjbhkTkUnR8XWP3hvDH6PZP
bHIHtWLDKtA8ngDB/lbYTjIxrbdX8WPWwag3CxCva3adOxHFR8yjD55GevK3ZF4ujgU1lh06fta0dSS
OSepWvnsCb7IFxCw+MGiw0o88uHa2t4l+SillqfJvcpuXfiIA006D/gtDF400w5YKHEl2KhDevv+yQ
zxwfEz2cl7Ef5iMr8LgGcDK7qvMcvrqUqx68JQOK2+RSwtHQ15bCZmsDU5vQVCvSQWC8OMHNGwpS25
3XwRLvd/h6S/tM0k98QMv+i3N8oOdj1V7flBqech7+kVrU01F6oFzr0zM1kMyLr5Hh1VDh7B0k9wp0
dUFZiAzaf43jupD5f6CEkuLeudYWlxgNzsR8eqtPK6t1gFJyOn0s7QdNQ7q9knsPJsekRAH9hrLBhW
BLTU/3B1QS94wEGD2YTuiW36TiP14hYwO1CAYRj2/bY3+1YzVu7EmzMq+UefYxh4cF2gYD8RZL2Rwm
P9JV5148I3XBFPNUw/3V5jf7nRuLr/CdfK3008+Pa3V6/nNhokErSgyxjzMD88DVzM41LxxaUDhbcm
koHT9ImzBvKzJX0J+o7FoUDFOxEdIq1AN4GNqk49cpi2sXDbIarALp6B13+tbB4M1HGH+0CPscZXqo
i/kon9YmGauHyRs+0m6wthdlAmCnv1JCDfDoXtn8DpabgiW6VDTrv13SGPyQtUv7Wdahug5SxbUzjY
2JxQnrUtWb977NCzYu2sOtN+dsEReWJ6ueyJBbMzKyzUB4L3i5uSYN50B4PCv1w5KdRKA5p3N0Nfq6
RM6dfipMEJw0Ny1sZ7ohz3fbvQ/YZ7lw/k7ods/8VbaR15ivkE8dSCzuf/AInHtCzuQ6wApzEp9CUo
G8/dapWriHjNoi41lJOgCst33wEhxFxcWy2UWxs4EZSjsI5GyBnefSQTPVfma5dc/emWor9vWr0HnT
QaHP5rg5dTnqunkDEdMIhfbeP3F90cZeJvZihM6igiS6P/CEJAjE; Domain=.example.com; Path=
/
```

Figure 39-4 shows an example of authorization cookies in HTTP analyzer output. This is only an example; output varies widely across different websites.

**Figure 39-4** Authorization cookies in sample HTTP analyzer output

NO	TimeStart	Duration(s)	Method	Result	Size	Type	URL	RedirectURL
36	16:11:03:1...	0.009 s	GET	304	226	text/html	http://www.example.com/	
37	16:11:03:1...	0.080 s	GET	200	335	application/javascript	http://www.example.com/js/global.js	

Request Headers	Value	Response Headers	Value
(Request-Line)	GET /auth/login HTTP/1.1	(Status-Line)	HTTP/1.1 200 OK
Accept	image/gif, image/x-bitmap, image/jpeg, image/pipe, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-flash, */*	Server	Netscape-Enterprise/6.0
Accept-Language	en-us	Date	Thu, 15 Dec 2005 21:11:08 GMT
Accept-Encoding	gzip, deflate	Content-length	136
User-Agent	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)	Content-type	text/html
Host	webauth.example.com	Set-cookie	AUTH=; path=/; domain=.example.com; expires=Wed, 01-Jan-1997 00:00:00 GMT;
Connection	Keep-Alive	Set-cookie	SAUTH=; path=/; domain=.example.com; expires=Wed, 01-Jan-1997 00:00:00 GMT; secure
Cookie	ISINNETWDRK=network=outofnet; RMD=a12c800i4390ca0; RMDf=011Emxm0204fRIQ104Uq; CPAC=d2dba143; SESSIONHOME=home; SAUTH=wkB9g1HKNAhNK7hmDZ56xeTutAuTHZ+E AUTH=sCZ0SDWig6pXc00dhu0oHheTutAuTHZRfud;	Set-cookie	AUTH=Tz8la/yAH+8GBnRMB7yShP/LRkCzmDttBfzDrc4kx4Eh2DEpi+etoFyEF4CITRLHN/cJ86BYCoikI path=/; domain=.example.com
		Connection	close

1 AUTH=; path=/; domain=.example.com; expires=Wed, 01-Jan-1997 00:00:00 GMT;  
SAUTH=; path=/; domain=.example.com; expires=Wed, 01-Jan-1997 00:00:00 GMT; secure

## 1 Authorization cookies

**Step 7** In some cases, the server may set the same cookie regardless of whether the authentication was successful or not, and such a cookie is unacceptable for SSO purposes. To confirm that the cookies are different, repeat [Step 1](#) through [Step 6](#) using invalid login credentials and then compare the “failure” cookie with the “success” cookie.

You now have the necessary parameter data to configure the security appliance for SSO with HTTP Form protocol.

### Task Overview: Configuring SSO with HTTP Form Protocol

This section presents an overview of configuring SSO with the HTTP Form protocol. To enable SSO using HTTP Forms, perform the following tasks:

- Configure the uniform resource identifier on the authenticating web server to receive and process the form data (**action-uri**).
- Configure the username parameter (**user-parameter**).
- Configure the user password parameter (**password-parameter**).

You might also need to do the following tasks depending upon the requirements of authenticating web server:

- Configure a starting URL if the authenticating web server requires a pre-login cookie exchange (**start-url**).
- Configure any hidden authentication parameters required by the authenticating web server (**hidden-parameter**).
- Configure the name of an authentication cookie set by the authenticating web server (**auth-cookie-name**).

### Detailed Tasks: Configuring SSO with HTTP Form Protocol

This section presents the detailed tasks required to configure SSO with the HTTP Form protocol. Perform the following steps to configure the security appliance to use HTTP Form protocol for SSO:

- Step 1** If the authenticating web server requires it, enter the **start-url** command in aaa-server-host configuration mode to specify the URL from which to retrieve a pre-login cookie from the authenticating web server. For example, to specify the authenticating web server URL `http://example.com/east/Area.do?Page-Grp1` in the `testgrp1` server group with an IP address of 10.0.0.2, enter the following:

```
hostname(config)# aaa-server testgrp1 protocol http-form
hostname(config)aaa-server testgrp1 host 10.0.0.2
hostname(config-aaa-server-host)# start-url http://example.com/east/Area.do?Page-Grp1
hostname(config-aaa-server-host)#
```

- Step 2** To specify a URI for an authentication program on the authenticating web server, enter the **action-uri** command in aaa-server- host configuration mode. A URI can be entered on multiple, sequential lines. The maximum number of characters per line is 255. The maximum number of characters for a complete URI is 2048. An example action URI follows:

```
http://www.example.com/auth/index.html/appdir/authc/forms/MCologin.fcc?TYPE=33554433&REALMOID=06-000a1311-a828-1185-ab41-8333b16a0008&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=SM5FZmjnk3DRNwNjk2KcqVCFbIrNT9%2bJ0H0KPshFtg6rB1UV2PxkHqLw%3d%3d&TARGET=https%3A%2F%2Fauth.example.com
```

To specify this action URI, enter the following commands:

```
hostname(config-aaa-server-host)# action-uri http://www.example.com/auth/index.htm
hostname(config-aaa-server-host)# action-uri 1/appdir/authc/forms/MCologin.fcc?TYP
hostname(config-aaa-server-host)# action-uri 554433&REALMOID=06-000a1311-a828-1185
hostname(config-aaa-server-host)# action-uri -ab41-8333b16a0008&GUID=&SMAUTHREASON
hostname(config-aaa-server-host)# action-uri =0&METHOD=GET&SMAGENTNAME=SM5FZmjnk
hostname(config-aaa-server-host)# action-uri 3DRNwNjk2KcqVCFbIrNT9%2bJ0H0KPshFtg6r
```

```
hostname(config-aaa-server-host) # action-uri B1UV2PxxHqLw%3d%3d&TARGET=https%3A%2F
hostname(config-aaa-server-host) # action-uri %2Fauth.example.com
hostname(config-aaa-server-host) #
```



**Note** You must include the hostname and protocol in the action URI. In the preceding example, these appear at the start of the URI in `http://www.example.com`.

- Step 3** To configure a username parameter for the HTTP POST request, enter the **user-parameter** command in `aaa-server-host` configuration mode. For example, the following command configures the username parameter `userid`:

```
hostname(config-aaa-server-host) # user-parameter userid
hostname(config-aaa-server-host) #
```

- Step 4** To configure a user password parameter for the HTTP POST request, use the **password-parameter** command in `aaa-server-host` configuration mode. For example, the following command configures a user password parameter named `user_password`:

```
hostname(config-aaa-server-host) # password-parameter user_password
hostname(config-aaa-server-host) #
```

- Step 5** To specify hidden parameters for exchange with the authenticating web server, use the **hidden-parameter** command in `aaa-server-host` configuration mode. An example hidden parameter excerpted from a POST request follows:

```
SMENC=ISO-8859-1&SMLOCALE=US-EN&target=https%3A%2F%2Fwww.example.com%2Femco%2Fappdir%2FAreaRoot.do%3FEMCOPageCode%3DENG&smauthreason=0
```

This hidden parameter includes four form entries and their values, separated by `&`. The four entries and their values are:

- SMENC with a value of ISO-8859-1
- SMLOCALE with a value of US-EN
- target with a value of `https%3A%2F%2Fwww.example.com%2Femco%2Fappdir%2FAreaRoot.do%3FEMCOPageCode%3DENG`
- smauthreason with a value of 0

To specify this hidden parameter, enter the following commands:

```
hostname(config) # aaa-server testgrp1 host example.com
hostname(config-aaa-server-host) # hidden-parameter SMENC=ISO-8859-1&SMLOCALE=US-EN&targe
hostname(config-aaa-server-host) # hidden-parameter t=https%3A%2F%2Fwww.example.com%2Femc
hostname(config-aaa-server-host) # hidden-parameter o%2Fappdir%2FAreaRoot.do%3FEMCOPageCo
hostname(config-aaa-server-host) # hidden-parameter de%3DENG&smauthreason=0
hostname(config-aaa-server-host) #
```

- Step 6** To specify the name for the authentication cookie, enter the **auth-cookie-name** command in `aaa-server-host` configuration mode. This command is optional. The following example specifies the authentication cookie name of `SsoAuthCookie`:

```
hostname(config-aaa-server-host) # auth-cookie-name SsoAuthCookie
hostname(config-aaa-server-host) #
```

- Step 7** To configure a tunnel group to use the SSO server configured in these steps, use the **authentication-server-group** command from `tunnel-group general-attributes` mode. The following example configures the tunnel-group named `/testgroup/` to use the SSO server(s) named `/testgrp1/`.

```
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-tunnel-general)# authentication-server-group testgrp1
```

## Configuring SSO with Macro Substitution

This section describes using macro substitution for SSO. Configuring SSO with macro substitution allows for you to inject certain variables into bookmarks to substitute for dynamic values.



### Note

Smart tunnel bookmarks support auto-signon but not variable substitution. For example, a SharePoint bookmark configured for smart tunnel uses the same username and password credentials to log into the application as the credentials used to log into clientless SSL VPN. You can use variable substitutions and auto signon simultaneously or separately.

The following variables (or macros) allow for substitutions in bookmarks and forms-based HTTP POST operations:

- CSCO\_WEBVPN\_USERNAME — user login ID
- CSCO\_WEBVPN\_PASSWORD — user login password
- CSCO\_WEBVPN\_INTERNAL\_PASSWORD — user internal (or domain) password. This cached credential is not authenticated against a AAA server. When you enter this value, the security appliance uses it as the password for auto signon, instead of the password/primary password value.



### Note

You cannot use any of these three variables in GET-based http(s) bookmarks. Only POST-based http(s) and cifs bookmarks can use these variables.

- CSCO\_WEBVPN\_CONNECTION\_PROFILE —user login group drop-down (connection profile alias)
- CSCO\_WEBVPN\_MACRO1 — set with the RADIUS-LDAP Vendor Specific Attribute (VSA). If you are mapping from LDAP with an ldap-attribute-map command, use the WebVPN-Macro-Substitution-Value1 Cisco attribute for this macro. See the Active Directory ldap-attribute-mapping examples at [http://www.cisco.com/en/US/docs/security/asa/asa82/configuration/guide/ref\\_extserver.html#wp1572118](http://www.cisco.com/en/US/docs/security/asa/asa82/configuration/guide/ref_extserver.html#wp1572118).

The CSCO\_WEBVPN\_MACRO1 macro substitution with RADIUS is performed by VSA#223 (see the following table).

**Table 39-1 VSA#223**

WebVPN-Macro-Value1	Y	223	String	Single	Unbounded
WebVPN-Macro-Value2	Y	224	String	Single	Unbounded

A value such as [www.cisco.com/email](http://www.cisco.com/email) dynamically populates a bookmark on the Clientless SSL VPN portal, such as [https://CSCO\\_WEBVPN\\_MACRO1](https://CSCO_WEBVPN_MACRO1) or [https://CSCO\\_WEBVPN\\_MACRO2](https://CSCO_WEBVPN_MACRO2) for the particular DAP or group policy.

- **CSCO\_WEBVPN\_MACRO2** —Set with RADIUS-LDAP Vendor Specific Attribute (VSA). If you are mapping from LDAP with an `ldap-attribute-map` command, use the `WebVPN-Macro-Substitution-Value2` Cisco attribute for this macro. See the Active Directory `ldap-attribute-mapping` examples at [http://www.cisco.com/en/US/docs/security/asa/asa82/configuration/guide/ref\\_extserver.html#wp1572118](http://www.cisco.com/en/US/docs/security/asa/asa82/configuration/guide/ref_extserver.html#wp1572118).

The **CSCO\_WEBVPN\_MACRO2** macro substitution with RADIUS is performed by VSA#224 (see the previous table).

Each time clientless SSL VPN recognizes one of these six strings in an end-user request (in the form of a bookmark or Post Form), it replaces the string with the user-specified value and then passes the request to a remote server.

## Authenticating with Digital Certificates

Clientless SSL VPN users that authenticate using digital certificates do not use global authentication and authorization settings. Instead, they use an authorization server to authenticate once the certificate validation occurs. For more information on authentication and authorization using digital certificates, see “[Using Certificates and User Login Credentials](#)” in the “[Configuring AAA Servers and the Local Database](#)” chapter.

## Creating and Applying Clientless SSL VPN Policies for Accessing Resources

Creating and applying policies for clientless SSL VPN that govern access to resources at the central site includes the following task:

- [Assigning Users to Group Policies](#)

Chapter 32, “[Configuring Connection Profiles, Group Policies, and Users](#)” includes step-by-step instructions for all of these tasks.

## Assigning Users to Group Policies

Assigning users to group policies simplifies the configuration by letting you apply policies to many users. You can use an internal authentication server or a RADIUS server to assign users to group policies. See [Chapter 32, “Configuring Connection Profiles, Group Policies, and Users”](#) for a thorough explanation of ways to simplify configuration with group policies.

## Using the Security Appliance Authentication Server

You can configure users to authenticate to the security appliance internal authentication server, and assign these users to a group policy on the security appliance.

## Using a RADIUS Server

Using a RADIUS server to authenticate users, assign users to group policies by following these steps:



- Step 1** Authenticate the user with RADIUS and use the Class attribute to assign that user to a particular group policy.
- Step 2** Set the class attribute to the group policy name in the format `OU=group_name`
- For example, to assign a user of clientless SSL VPN to the `SSL_VPN` group, set the RADIUS Class Attribute to a value of `OU=SSL_VPN`; (Do not omit the semicolon.)

## Configuring Connection Profile Attributes for Clientless SSL VPN

Table 39-2 provides a list of connection profile attributes that are specific to clientless SSL VPN. In addition to these attributes, you configure general connection profile attributes common to all VPN connections. For step-by-step information on configuring connection profiles, see “Configuring Connection Profiles for Clientless SSL VPN Sessions” in Chapter 32, “Configuring Connection Profiles, Group Policies, and Users.”



### Note

In earlier releases, “connection profiles” were known as “tunnel groups.” You configure a connection profile with tunnel-group commands. This chapter often uses these terms interchangeably.

**Table 39-2** Connection Profile Attributes for Clientless SSL VPN

Command	Function
<b>authentication</b>	Sets the authentication method.
<b>customization</b>	Identifies the name of a previously defined customization to apply.
<b>nbns-server</b>	Identifies the name of the NetBIOS Name Service server (nbns-server) to use for CIFS name resolution.
<b>group-alias</b>	Specifies the alternate names by which the server can refer to a connection profile
<b>group-url</b>	Identifies one or more group URLs. If you configure this attribute, users coming in on a specified URL need not select a group at login
<b>dns-group</b>	Identifies the DNS server group that specifies the DNS server name, domain name, name server, number of retries, and timeout values
<b>hic-fail-group-policy</b>	Specifies a VPN feature policy if you use the Cisco Secure Desktop Manager to set the Group-Based Policy attribute to “Use Failure Group-Policy” or “Use Success Group-Policy, if criteria match.”
<b>override-svc-download</b>	Overrides downloading the group-policy or username attributes configured for downloading the AnyConnect VPN client to the remote user.
<b>radius-reject-message</b>	Enables the display of the RADIUS reject message on the login screen when authentication is rejected.

# Configuring Group Policy and User Attributes for Clientless SSL VPN

Table 39-3 provides a list of group policy and user attributes for clientless SSL VPN. For step-by-step instructions on configuring group policy and user attributes, see “Configuring Group Policies” and “Configuring Attributes for Specific Users” in Chapter 32, “Configuring Connection Profiles, Group Policies, and Users.”

**Table 39-3**      *Group Policy and User Attributes for Clientless SSL VPN*

Command	Function
<b>activex-relay</b>	Lets a user who has established a clientless SSL VPN session use the browser to launch Microsoft Office applications. The applications use the session to download and upload Microsoft Office documents. The ActiveX relay remains in force until the clientless SSL VPN session closes.
<b>auto-signon</b>	Sets values for auto signon, which requires only that the user enter username and password credentials only once for a clientless SSL VPN connection.
<b>customization</b>	Assigns a customization object to a group-policy or user.
<b>deny-message</b>	Specifies the message delivered to a remote user who logs into clientless SSL VPN successfully, but has no VPN privileges.
file-browsing	Enables CIFS file browsing for file servers and shares. Browsing requires NBNS (Master Browser or WINS)
file-entry	Allows users to enter file server names to access.
<b>filter</b>	Sets the name of the webtype access list.
hidden-shares	Controls the visibility of hidden shares for CIFS files.
<b>homepage</b>	Sets the URL of the web page that displays upon login.
<b>html-content-filter</b>	Configures the content and objects to filter from the HTML for this group policy.
<b>http-comp</b>	Configures compression.
http-proxy	Configures the security appliance to use an external proxy server to handle HTTP requests.
<b>keep-alive-ignore</b>	Sets the maximum object size to ignore for updating the session timer.
<b>port-forward</b>	Applies a list of clientless SSL VPN TCP ports to forward. The user interface displays the applications on this list.
<b>post-max-size</b>	Sets the maximum object size to post.
smart-tunnel	Configures a list of programs to use smart tunnel.
<b>sso-server</b>	Sets the name of the SSO server.
storage-objects	Configures storage objects for the data stored between sessions.
<b>svc</b>	Configures SSL VPN Client attributes.
unix-auth-gid	Sets the UNIX group ID.
unix-auth-uid	Sets the UNIX user ID.
upload-max-size	Sets the maximum object size to upload.
url-entry	Controls the ability of the user to enter any HTTP/HTTP URL.

**Table 39-3** Group Policy and User Attributes for Clientless SSL VPN

Command	Function
url-list	Applies a list of servers and URLs that the clientless SSL VPN portal page displays for end user access.
user-storage	Configures a location for storing user data between sessions.

## Configuring Browser Access to Client-Server Plug-ins

The following sections describe the integration of browser plug-ins for clientless SSL VPN browser access:

- [Introduction to Browser Plug-Ins, page 39-27](#)
- [Plug-in Requirements and Restrictions, page 39-28](#)
- [Preparing the Security Appliance for a Plug-in, page 39-28](#)
- [Installing Plug-ins Redistributed By Cisco, page 39-29](#)
- [Providing Access to Third-Party Plug-ins, page 39-31](#)
- [Viewing the Plug-ins Installed on the Security Appliance, page 39-32](#)

## Introduction to Browser Plug-Ins

A browser plug-in is a separate program that a web browser invokes to perform a dedicated function, such as connect a client to a server within the browser window. The security appliance lets you import plug-ins for download to remote browsers in clientless SSL VPN sessions. Of course, Cisco tests the plug-ins it redistributes, and in some cases, tests the connectivity of plug-ins we cannot redistribute. However, we do not recommend importing plug-ins that support streaming media at this time.



### Note

Per the GNU General Public License (GPL), Cisco redistributes plug-ins without having made any changes to them. Per the GPL, Cisco cannot directly enhance these plug-ins.

The security appliance does the following when you install a plug-in onto the flash device:

- (Cisco-distributed plug-ins only) Unpacks the jar file specified in the *URL*.
- Writes the file to the `cisco-config/97/plugin` directory on the security appliance file system.
- Populates the drop-down menu next to the URL attributes in ASDM.
- Enables the plug-in for all future clientless SSL VPN sessions, and adds a main menu option and an option to the drop-down menu next to the Address field of the portal page.

[Table 39-4](#) shows the changes to the main menu and address field of the portal page when you add the plug-ins described in the following sections.

**Table 39-4** Effects of Plug-ins on the Clientless SSL VPN Portal Page

Plug-in	Main Menu Option Added to Portal Page	Address Field Option Added to Portal Page
ica	Citrix Client	citrix://
rdp	Terminal Servers	rdp://

**Table 39-4** Effects of Plug-ins on the Clientless SSL VPN Portal Page

Plug-in	Main Menu Option Added to Portal Page	Address Field Option Added to Portal Page
rdp2	Terminal Servers Vista	rdp2://
ssh,telnet	SSH	ssh://
	Telnet	telnet://
vnc	VNC Client	vnc://

When the user in a clientless SSL VPN session clicks the associated menu option on the portal page, the portal page displays a window to the interface and displays a help pane. The user can select the protocol displayed in the drop-down menu and enter the URL in the Address field to establish a connection.

**Note**

Some Java plug-ins may report a status of connected or online even when a session to the destination service is not set up. The open-source plug-in reports the status, not the security appliance.

## Plug-in Requirements and Restrictions

Clientless SSL VPN must be enabled on the security appliance to provide remote access to the plug-ins. The plug-ins do not work if the security appliance configures the clientless session to use a proxy server.

**Note**

The remote desktop protocol plug-in does not support load balancing with a session broker. Because of the way the protocol handles the redirect from the session broker, the connection fails. If a session broker is not used, the plug-in works.

The plug-ins support single sign-on (SSO). They use the *same* credentials entered to open the clientless SSL VPN session. Because the plug-ins do not support macro substitution, you do not have the options to perform SSO on different fields such as the internal domain password or on an attribute on a Radius or LDAP server.

To configure SSO support for a plug-in, you install the plug-in, add a bookmark entry to display a link to the server, and specify SSO support when adding the bookmark.

The minimum access rights required for remote use belong to the guest privilege mode.

A stateful failover does not retain sessions established using plug-ins. Users must reconnect following a failover.

## Preparing the Security Appliance for a Plug-in

Before installing a plug-in, prepare the security appliance as follows:

- Step 1** Make sure clientless SSL VPN (“webvpn”) is enabled on a security appliance interface. To do so, enter the **show running-config** command.
- Step 2** Install an SSL certificate onto the security appliance interface to which remote users use a fully-qualified domain name (FQDN) to connect.

**Note**

Do not specify an IP address as the common name (CN) for the SSL certificate. The remote user attempts to use the FQDN to communicate with the security appliance. The remote PC must be able to use DNS or an entry in the System32\drivers\etc\hosts file to resolve the FQDN.

Go to the section that identifies the type of plug-in you want to provide for clientless SSL VPN access.

- [Installing Plug-ins Redistributed By Cisco, page 39-29](#)
- [Providing Access to Third-Party Plug-ins, page 39-31](#)

## Installing Plug-ins Redistributed By Cisco

Cisco redistributes the following open-source, Java-based components to be accessed as plug-ins for web browsers in clientless SSL VPN sessions:

**Table 39-5** *Plug-ins Redistributed by Cisco*

Cisco Download Link	Protocol	Description	Source of Redistributed Plug-in
<a href="#">rdp2-plugin.090211.jar</a>	RDP2	Accesses Microsoft Terminal Services hosted by Windows Vista and Windows 2003 R2. Supports Remote Desktop ActiveX Control. <b>Note:</b> You can import the RDP and RDP2 plug-ins to make both of them available to clientless users.	Cisco redistributes this plug-in without any changes to it per the GNU General Public License. The original source of the redistributed plug-in is <a href="http://properjavardp.sourceforge.net/">http://properjavardp.sourceforge.net/</a>
<a href="#">rdp-plugin.080506.jar</a>	RDP	Accesses Microsoft Terminal Services hosted by Windows 2003 R1. Supports Remote Desktop ActiveX Control.	Cisco redistributes this plug-in without any changes to it per the GNU General Public License. The source of the redistributed plug-in is <a href="http://properjavardp.sourceforge.net/">http://properjavardp.sourceforge.net/</a>
<a href="#">ssh-plugin.080430.jar</a>	SSH	The Secure Shell-Telnet plug-in lets the remote user establish a Secure Shell or Telnet connection to a remote computer.	Cisco redistributes this plug-in without any changes to it per the GNU General Public License. The web site containing the source of the redistributed plug-in is <a href="http://javassh.org/">http://javassh.org/</a>
<a href="#">vnc-plugin.080130.jar</a>	VNC	The Virtual Network Computing plug-in lets the remote user use a monitor, keyboard, and mouse to view and control a computer with remote desktop sharing turned on. This version changes the default color of the text, and contains updated French and Japanese help files.	Cisco redistributes this plug-in without any changes to it per the GNU General Public License. The web site containing the source of the redistributed plug-in is <a href="http://www.tightvnc.com/">http://www.tightvnc.com/</a>

These plug-ins are available on the [Cisco Adaptive Security Appliance Software Download](#) site.

Before installing a plug-in:

- Make sure clientless SSL VPN (“webvpn”) is enabled on an interface on the security appliance. To do so, enter the **show running-config** command.

- Create a temporary directory named “plugins” on a local TFTP or FTP server (for example, with the hostname “local\_tftp\_server”), and download the plug-ins from the Cisco web site to the “plugins” directory.

To provide clientless SSL VPN browser access to a plug-in redistributed by Cisco, install the plug-in onto the flash device of the security appliance by entering the following command in privileged EXEC mode.

**import webvpn plug-in protocol *protocol URL***

*protocol* is one of the following values:

- **rdp** to provide plug-in access to Remote Desktop Protocol services. Then specify the path to the rdp-plugin.080130.jar file in the *URL* field.
- **ssh,telnet** to provide plug-in access to *both* Secure Shell and Telnet services. Then specify the path to the ssh-plugin.jar file in the *URL* field.



#### Caution

Do *not* enter this command once for SSH and once for Telnet. When typing the **ssh,telnet** string, do *not* insert a space. Use the **revert webvpn plug-in protocol** command to remove any **import webvpn plug-in protocol** commands that deviate from these requirements.

- **vnc** to provide plug-in access to Virtual Network Computing services. Then specify the path to the vnc-plugin.080130.jar file in the *URL* field.

*URL* is the remote path to the source of the plug-in. Enter the host name or address of the TFTP or FTP server and the path to the plug-in.

The following example command adds clientless SSL VPN support for RDP:

```
hostname# import webvpn plug-in protocol rdp
tftp://local_tftp_server/plugins/rdp-plugin.080130.jar
Accessing
tftp://local_tftp_server/plugins/rdp-plugin.080130.jar...!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!
Writing file disk0:/cisco_config/97/plugin/rdp...
!!
329994 bytes copied in 5.350 secs (65998 bytes/sec)
```

The following example command adds clientless SSL VPN support for SSH and Telnet:

```
hostname# import webvpn plug-in protocol ssh,telnet
tftp://local_tftp_server/plugins/ssh-plugin.jar
Accessing
tftp://local_tftp_server/plugins/ssh-plugin.jar...!!
Writing file disk0:/cisco_config/97/plugin/ssh...
!!
238510 bytes copied in 3.650 secs (79503 bytes/sec)
```

The following example command adds clientless SSL VPN support for VNC:

```
hostname# import webvpn plug-in protocol vnc
tftp://local_tftp_server/plugins/vnc-plugin.080130.jar
Accessing tftp://local_tftp_server/plugins/vnc-plugin.080130.jar...!!!!!!!!!!!!!!!!!!!!
Writing file disk0:/cisco_config/97/plugin/vnc...
!!!!!!!!!!!!!!!!!!!!
58147 bytes copied in 2.40 secs (29073 bytes/sec)
```

**Note**

The security appliance does not retain the **import webvpn plug-in protocol** command in the configuration. Instead, it loads the contents of the `cisco-config/97/plugin` directory automatically. A secondary security appliance obtains the plug-ins from the primary security appliance.

After you import a plug-in, type the corresponding protocol and resource location in the address bar of the SSL VPN home page to access it. For example:

```
rdp://10.1.1.1
vnc://10.1.1.1
ssh://10.1.1.1
telnet://10.1.1.1
```

To disable and remove clientless SSL VPN support for a Java-based client application, as well as to remove it from the flash drive of the security appliance, use the following command:

**revert webvpn plug-in protocol** *protocol*

The following example command removes RDP:

```
hostname# revert webvpn plug-in protocol rdp
```

## Providing Access to Third-Party Plug-ins

The open framework of the security appliance lets you add plug-ins to support third-party Java client/server applications.

**Caution**

Cisco does not provide direct support for or recommend any particular plug-ins that are not redistributed by Cisco. As a provider of clientless SSL VPN services, you are responsible for reviewing and complying with any license agreements required for the use of plug-ins.

### Example: Providing Access to a Citrix Java Presentation Server

The open framework of the security appliance lets you add plug-ins to support third-party Java client/server applications. As an example of how to provide clientless SSL VPN browser access to third-party plug-ins, this section describes how to add clientless SSL VPN support for the Citrix Presentation Server Client.

**Caution**

Cisco does not provide direct support for or recommend any particular plug-ins that are not redistributed by Cisco. As a provider of clientless SSL VPN services, you are responsible for reviewing and complying with any license agreements required for the use of plug-ins.

With a Citrix plug-in installed on the security appliance, clientless SSL VPN users can use a connection to the security appliance to access Citrix MetaFrame services.

A stateful failover does not retain sessions established using the Citrix plug-in. Citrix users must reauthenticate after failover.

To provide access to the Citrix plug-in, follow the procedures in the following sections.

- [Preparing the Citrix MetaFrame Server for Clientless SSL VPN Access](#)
- [Creating and Installing the Citrix Plug-in](#)

## Preparing the Citrix MetaFrame Server for Clientless SSL VPN Access

The security appliance performs the connectivity functions of the Citrix secure gateway when the Citrix client connects to the Citrix MetaFrame Server. Therefore, you must configure the Citrix Web Interface software to operate in a mode that does not use the (Citrix) “secure gateway.” Otherwise, the Citrix client cannot connect to the Citrix MetaFrame Server.




### Note

If you are not already providing support for a plug-in, you must follow the instructions in the “[Preparing the Security Appliance for a Plug-in](#)” section on page 39-28 before using this section.

## Creating and Installing the Citrix Plug-in

To create and install the Citrix plug-in, perform the following steps:

- 
- Step 1** Download the ica-plugin.zip file from the Cisco Software Download web site, <http://www.cisco.com/cisco/software/navigator.html>.  
This file contains files that Cisco customized for use with the Citrix plug-in.
- Step 2** Download the Citrix Java client from the Citrix site.
- Step 3** Extract the following files from the Citrix Java client, then add them to the ica-plugin.zip file:
- JICA-configN.jar
  - JICAEngN.jar
- You can use WinZip to perform this step.
- Step 4** Ensure the EULA included with the Citrix Java client grants you the rights and permissions to deploy the client on your web servers.
- Step 5** Open a CLI session with the security appliance and install the plug-in by entering the following command in privileged EXEC mode:
- ```
import webvpn plug-in protocol ica URL
```
- URL* is the host name or IP address and path to the ica-plugin.zip file.
-
-  **Note** After you import the plug-in, remote users can choose **ica** and enter *host/?DesiredColor=4&DesiredHRes=1024&DesiredVRes=768* into the Address field of the portal page to access Citrix services. We recommend that you add a bookmark to make it easy for users to connect. Adding a bookmark is required if you want to provide SSO support for Citrix sessions.
-
- Step 6** Establish an SSL VPN clientless session and click the bookmark or enter the URL for the citrix server.
Use the [Client for Java Administrator's Guide](#) as needed.
-

Viewing the Plug-ins Installed on the Security Appliance

Enter the following command in privileged EXEC mode to list the Java-based client applications available to users of clientless SSL VPN:

show import webvpn plug-in

For example:

```
hostname# show import webvpn plug-in
ssh
rdp
vnc
ica
```

Configuring Application Access

The following sections describe how to enable smart tunnel access and port forwarding on clientless SSL VPN sessions, specify the applications to be provided with such access, and provide notes on using it:

- [Configuring Smart Tunnel Access](#)
- [Configuring Port Forwarding](#)
- [Application Access User Notes](#)

Configuring Smart Tunnel Access

The following sections describe smart tunnels and how to configure them:

- [About Smart Tunnels](#)
- [Why Smart Tunnels?](#)
- [Smart Tunnel Requirements, Restrictions, and Limitations](#)
- [Adding Applications to Be Eligible for Smart Tunnel Access](#)
- [Assigning a Smart Tunnel List](#)
- [Configuring Smart Tunnel Auto Sign-on](#)
- [Automating Smart Tunnel Access](#)
- [Enabling and Disabling Smart Tunnel Access](#)

About Smart Tunnels

A smart tunnel is a connection between a TCP-based application and a private site, using a clientless (browser-based) SSL VPN session with the security appliance as the pathway, and the security appliance as a proxy server. You can identify applications to which you want to grant smart tunnel access, and specify the local path to each application. For applications running on Microsoft Windows, you can also require a match of the SHA-1 hash of the checksum as a condition for granting smart tunnel access.

Lotus SameTime and Microsoft Outlook Express are examples of applications to which you might want to grant smart tunnel access.

Configuring smart tunnels requires one of the following procedures, depending on whether the application is a client or is a web-enabled application:

- Create one or more smart tunnel lists of the client applications, then assign the list to the group policies or local user policies for whom you want to provide smart tunnel access.

- Create one or more bookmark list entries that specify the URLs of the web-enabled applications eligible for smart tunnel access, then assign the list to the DAPs, group policies, or local user policies for whom you want to provide smart tunnel access.

You can also list web-enabled applications for which to automate the submission of login credentials in smart tunnel connections over clientless SSL VPN sessions.

Why Smart Tunnels?

Smart tunnel access lets a client TCP-based application use a browser-based VPN connection to connect to a service. It offers the following advantages to users, compared to plug-ins and the legacy technology, port forwarding:

- Smart tunnel offers better performance than plug-ins.
- Unlike port forwarding, smart tunnel simplifies the user experience by not requiring the user connection of the local application to the local port.
- Unlike port forwarding, smart tunnel does not require users to have administrator privileges.

The advantage of a plug-in is that it does not require the client application to be installed on the remote computer.

Smart Tunnel Requirements, Restrictions, and Limitations

The following sections categorize the smart tunnel requirements and limitations.

General Requirements and Limitations

Smart tunnel has the following general requirements and limitations:

- The remote host originating the smart tunnel must be running a 32-bit version of Microsoft Windows Vista, Windows XP, or Windows 2000; or Mac OS 10.4 or 10.5.
- Smart tunnel auto sign-on supports only Microsoft Internet Explorer on Windows.
- The browser must be enabled with Java, Microsoft ActiveX, or both.
- Smart tunnel supports only proxies placed between computers running Microsoft Windows and the security appliance. Smart tunnel uses the Internet Explorer configuration (that is, the one intended for system-wide use in Windows). If the remote computer requires a proxy server to reach the security appliance, the URL of the terminating end of the connection must be in the list of URLs excluded from proxy services. If the proxy configuration specifies that traffic destined for the ASA goes through a proxy, all smart tunnel traffic goes through the proxy.

In an HTTP-based remote access scenario, sometimes a subnet does not provide user access to the VPN gateway. In this case, a proxy placed in front of the ASA to route traffic between the web and the end user's location provides web access. However, only VPN users can configure proxies placed in front of the ASA. When doing so, they must make sure these proxies support the CONNECT method. For proxies that require authentication, smart tunnel supports only the basic digest authentication type.

- When smart tunnel starts, the security appliance by default passes all browser traffic through the VPN session if the browser process is the same. The security appliance also does this if a tunnel-all policy applies. If the user starts another instance of the browser process, it passes all traffic through the VPN session. If the browser process is the same and the security appliance does not provide access to a URL, the user cannot open it. As a workaround, assign a tunnel policy that is not tunnel-all.

- A stateful failover does not retain smart tunnel connections. Users must reconnect following a failover.

Windows Requirements and Limitations

The following requirements and limitations apply to Windows only:

- Only Winsock 2, TCP-based applications are eligible for smart tunnel access.
- The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. Neither port forwarding nor the smart tunnel supports MAPI. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.
- Users of Microsoft Windows Vista who use smart tunnel or port forwarding must add the URL of the ASA to the Trusted Site zone. To access the Trusted Site zone, they must start Internet Explorer and choose the **Tools > Internet Options > Security** tab. Vista users can also disable Protected Mode to facilitate smart tunnel access; however, we recommend against this method because it increases vulnerability to attack.

Mac OS Requirements and Limitations

The following requirements and limitations apply to Mac OS only:

- Safari 3.1.1 or later, or Firefox 3.0 or later.
- Sun JRE 1.5 or later.
- Only applications started from the portal page can establish smart tunnel connections. This requirement includes smart tunnel support for Firefox. Using Firefox to start another instance of Firefox during the first use of a smart tunnel requires the user profile named `cisco_st`. If this user profile is not present, the session prompts the user to create one.
- Applications using TCP that are dynamically linked to the SSL library can work over a smart tunnel.
- The PowerPC MAC operating system is not supported with smart tunnel.
- Smart tunnel does not support the following on Mac OS:
 - Proxy services.
 - Auto sign-on.
 - Applications that use two-level name spaces.
 - Console-based applications, such as Telnet, SSH, and cURL.
 - Applications using `dlopen` or `dlsym` to locate libsocket calls.
 - Statically linked applications to locate libsocket calls.

Adding Applications to Be Eligible for Smart Tunnel Access

The clientless SSL VPN configuration of each security appliance supports *smart tunnel lists*, each of which identifies one or more applications eligible for smart tunnel access. Because each group policy or username supports only one smart tunnel list, you must group each set of applications to be supported into a smart tunnel list.

To add an entry to a list of applications that can use a clientless SSL VPN session to connect to private sites, enter the following command in webvpn configuration mode:

```
smart-tunnel list list application path [platform OS] [hash]
```

To remove an application from a list, use the **no** form of the command, specifying both the list and the name of the application.

no smart-tunnel list *list application*

To remove an entire list of applications from the security appliance configuration, use the **no** form of the command, specifying only the list.

no smart-tunnel list *list*

- *list* is the name for a list of applications or programs. Use quotation marks around the name if it includes a space. The string can be up to 64 characters. The CLI creates the list if it is not present in the configuration. Otherwise, it adds the entry to the list.



Note

To view the smart tunnel list entries in the SSL VPN configuration, enter the **show running-config webvpn** command in privileged EXEC mode.

- *application* is a string that serves as a unique index to each entry in the smart tunnel list. It typically names the application to be granted smart tunnel access. To support multiple versions of an application for which you choose to specify different paths or hash values, you can use this attribute to differentiate entries, specifying the OS, and name and version of the application supported by each list entry. The string can be up to 64 characters. To change an entry already present in a smart tunnel list, enter the name of the entry to be changed.
- *path* is the filename and extension of the application; or a path to the application, including its filename and extension. The string can be up to 128 characters.

Windows requires an exact match of this value to the right side of the application path on the remote host to qualify the application for smart tunnel access. If you specify only the filename for Windows, SSL VPN does not enforce a location restriction on the remote host to qualify the application for smart tunnel access.

If you specify a path and the user installed the application in another location, that application does not qualify. The application can reside on any path as long as the right side of the string matches the value you enter.

To authorize an application for smart tunnel access if it is present on one of several paths on the remote host, either specify only the name and extension of the application when you enter the *path* value; or enter the **smart-tunnel list** command once for each path, entering the same *list* string, but specifying the unique *application* string and *path* value in each command.



Note

A sudden problem with smart tunnel access may be an indication that a *Process Name* value is not up-to-date with an application upgrade. For example, the default path to an application sometimes changes following the acquisition of the company that produces the application and the next application upgrade.

For Windows, if you want to add smart tunnel access to an application started from the command prompt, you must specify “cmd.exe” in the Process Name of one entry in the smart tunnel list, and specify the path to the application itself in another entry, because “cmd.exe” is the parent of the application.

Mac OS requires the full path to the process, and is case-sensitive. To avoid specifying a path for each username, insert a tilde (~) before the partial path (e.g., ~/bin/vnc).

- **platform** is **windows** or **mac** to indicate the host OS of the application. The default value is **platform windows**.

- *hash* (Optional) To obtain this value, enter the checksum of the application (that is, the checksum of the executable file) into a utility that calculates a hash using the SHA-1 algorithm. One example of such a utility is the Microsoft File Checksum Integrity Verifier (FCIV), which is available at <http://support.microsoft.com/kb/841290/>. After installing FCIV, place a temporary copy of the application to be hashed on a path that contains no spaces (for example, c:/fciv.exe), then enter **fciv.exe -sha1 application** at the command line (for example, **fciv.exe -sha1 c:\msimn.exe**) to display the SHA-1 hash.

The SHA-1 hash is always 40 hexadecimal characters.

Before authorizing an application for smart tunnel access, clientless SSL VPN calculates the hash of the application matching the *path*. It qualifies the application for smart tunnel access if the result matches the value of *hash*.

Entering a hash provides a reasonable assurance that SSL VPN does not qualify an illegitimate file that matches the string you specified in the *path*. Because the checksum varies with each version or patch of an application, the *hash* you enter can only match one version or patch on the remote host. To specify a *hash* for more than one version of an application, enter the **smart-tunnel list** command once for each version, entering the same *list* string, but specifying a unique *application* string and a unique *hash* value.



Note

You must maintain the smart tunnel list in the future if you enter *hash* values and you want to support future versions or patches of an application with smart tunnel access. A sudden problem with smart tunnel access may be an indication that the application list containing *hash* values is not up-to-date with an application upgrade. You can avoid this problem by not entering a *hash*.

Table 39-6 Example smart-tunnel Commands

| Function | OS | Commands |
|--|----------------------------|--|
| Add Lotus SameTime to a smart tunnel list named lotus. | Windows (default platform) | smart-tunnel list lotus LotusSametime connect.exe |
| Add the Lotus 6.0 thick client with Domino Server 6.5.5 | Windows | smart-tunnel list lotus lotusnotes notes.exe
smart-tunnel list lotus lotusnlnotes nlnotes.exe
smart-tunnel list lotus lotusntaskldr ntaskldr.exe
smart-tunnel list lotus lotusnfileret nfileret.exe |
| Add the command prompt to a smart tunnel list named apps.

Note: This is necessary to provide smart tunnel access to a Microsoft Windows application started from the command prompt. You must also add the application itself to the list. | Windows | smart-tunnel list apps CommandPrompt cmd.exe |
| Add Windows Outlook Express. | Windows | smart-tunnel list apps OutlookExpress msimn.exe |
| Add Windows Outlook Express, permitting smart tunnel support for it only if its path on the remote host matches the string. | Windows | smart-tunnel list apps OutlookExpress "\Program Files\Outlook Express\msimn.exe" |

Table 39-6 Example smart-tunnel Commands

| Function | OS | Commands |
|---|---------|---|
| Add Windows Outlook Express, permitting smart tunnel support for it only if its hash matches the string. | Windows | <code>smart-tunnel list apps OutlookExpress msimn.exe 4739647b255d3ea865554e27c3f96b9476e75061</code> |
| Add Safari, permitting smart tunnel support for it only if its path on the remote host matches the string. | Mac OS | <code>smart-tunnel list apps Safari "/Applications/Safari" platform mac</code> |
| Add smart tunnel support for a new Terminal window. | Mac OS | <code>smart-tunnel list apps Terminal terminal platform mac</code> |
| Add smart tunnel support for an application started from a Mac Terminal window. All words after Terminal inside the quotation marks enter the command line. | Mac OS | <code>smart-tunnel list apps Terminal "terminal open -a MacTelnet" platform mac</code> |
| Add smart tunnel support for VNC, regardless of the user path to the VNC executable file. | Mac OS | <code>smart-tunnel list apps vnc "~/bin/vnc" platform mac</code> |

Following the configuration of a smart tunnel list, assign the list to group policies or usernames, as described in the next section.

Assigning a Smart Tunnel List

For each group policy and username, you can configure clientless SSL VPN to do one of the following:

- Start smart tunnel access automatically upon user login.
- Enable smart tunnel access upon user login, but require the user to start it manually, using the **Application Access > Start Smart Tunnels** button on the clientless SSL VPN Portal Page.



Note

These options are mutually exclusive for each group policy and username. Use only one.

Table 39-7 lists the smart tunnel commands available to each group policy and username. The configuration of each group policy and username supports only one of these commands at a time, so when you enter one, the security appliance replaces the one present in the configuration of the group policy or username in question with the new one, or in the case of the last command, simply removes the smart-tunnel command already present in the group policy or username.

Table 39-7 group-policy and username webvpn Smart Tunnel Commands

| Command | Description |
|---|--|
| <code>smart-tunnel auto-start list</code> | Starts smart tunnel access automatically upon user login. |
| <code>smart-tunnel enable list</code> | Enables smart tunnel access upon user login, but requires the user to start smart tunnel access manually, using the Application Access > Start Smart Tunnels button on the clientless SSL VPN portal page. |

Table 39-7 *group-policy and username webvpn Smart Tunnel Commands*

| Command | Description |
|--|---|
| smart-tunnel disable | Prevents smart tunnel access. |
| no smart-tunnel
[auto-start <i>list</i> enable <i>list</i> disable] | Removes a smart-tunnel command from the group policy or username configuration, which then inherits the [no] smart-tunnel command from the default group-policy. The keywords following the no smart-tunnel command are optional, however, they restrict the removal to the named smart-tunnel command. |

For details, go to the section that addresses the option you want to use.

Configuring Smart Tunnel Auto Sign-on

The following sections describe how to list the servers for which to provide auto sign-on in smart tunnel connections, and assign the lists to group policies or usernames.

Specifying Servers for Smart Tunnel Auto Sign-on

The Add Smart Tunnel Auto Sign-on Server List dialog box lets you add one or more lists of servers for which to automate the submission of login credentials during smart tunnel setup. The Edit Smart Tunnel Auto-signon Server List dialog box lets you modify the contents of these lists.

To create a list of servers for which to automate the submission of credentials in smart tunnel connections, enter the command in webvpn configuration mode.

[no] smart-tunnel auto-signon *list* [**use-domain**] {**ip** *ip-address* [*netmask*] | **host** *hostname-mask*}

Use this command for each server you want to add to a list. To remove an entry from a list, use the **no** form of the command, specifying both the list and the IP address or hostname, as it appears in the security appliance configuration. To display the smart tunnel auto sign-on list entries, enter the **show running-config webvpn smart-tunnel** command in privileged EXEC mode.

To remove an entire list of servers from the security appliance configuration, use the **no** form of the command, specifying only the list, as follows:

no smart-tunnel auto-signon *list*

- *list* names the list of remote servers. Use quotation marks around the name if it includes a space. The string can be up to 64 characters. The security appliance creates the list if it is not already present in the configuration. Otherwise, it adds the entry to the list. Assign a name that will help you to distinguish its contents or purpose from other lists are likely to be configured.
- **use-domain** (optional) adds the Windows domain to the username if authentication requires it. If you enter this keyword, be sure to specify the domain name when assigning the smart tunnel list to one or more group policies, or usernames.
- **ip** specifies the server by its IP address and netmask.
- *ip-address* [*netmask*] identifies the sub-network of hosts to auto-authenticate to.
- **host** specifies the server by its host name or wildcard mask. Using this option protects the configuration from dynamic changes to IP addresses.
- *hostname-mask* is the host name or wildcard mask to auto-authenticate to.

The following command adds all hosts in the subnet and adds the Windows domain to the username if authentication requires it:

```
asa2(config-webvpn) # smart-tunnel auto-signon HR use-domain ip 192.32.22.56 255.255.255.0
```

The following command removes that entry from the list:

```
asa2(config-webvpn) # no smart-tunnel auto-signon HR use-domain ip 192.32.22.56 255.255.255.0
```

The command shown above also removes the list named HR if the entry removed is the only entry in the list. Otherwise, the following command removes the entire list from the security appliance configuration:

```
asa2(config-webvpn) # no smart-tunnel auto-signon HR
```

The following command adds all hosts in the domain to the smart tunnel auto sign-on list named intranet:

```
asa2(config-webvpn) # smart-tunnel auto-signon intranet host *.exampledomain.com
```

The following command removes that entry from the list:

```
asa2(config-webvpn) # no smart-tunnel auto-signon intranet host *.exampledomain.com
```

Following the configuration of the smart tunnel auto sign-on server list, you must assign it to a group policy or a local user policy for it to become active, as described in the next section.

Adding or Editing a Smart Tunnel Auto Sign-on Server Entry

To enable smart tunnel auto sign-on in clientless (browser-based) SSL VPN sessions, use the **smart-tunnel auto-signon enable** command in group-policy webvpn configuration mode or username webvpn configuration mode.

[no] smart-tunnel auto-signon enable *list* [**domain** *domain*]

To remove the **smart-tunnel auto-signon enable** command from the group policy or username and inherit it from the default group-policy, use the **no** form of the command.

- *list* is the name of a smart tunnel auto sign-on list already present in the security appliance webvpn configuration. To view the smart tunnel auto sign-on list entries in the SSL VPN configuration, enter the **show running-config webvpn smart-tunnel** command in privileged EXEC mode.
- **domain** *domain* (optional) is the name of the domain to be added to the username during authentication. If you enter a domain, enter the **use-domain** keyword in the list entries.

The smart-tunnel auto sign-on feature supports only applications communicating HTTP and HTTPS using the Microsoft WININET library. For example, Microsoft Internet Explorer uses the WININET dynamic linked library to communicate with web servers.

You must use the **smart-tunnel auto-signon** *list* command to create a list of servers first. You can assign only one list to a group policy or username.

The following commands enable the smart tunnel auto sign-on list named HR:

```
hostname(config-group-policy) # webvpn
hostname(config-group-webvpn) # smart-tunnel auto-signon enable HR
hostname(config-group-webvpn)
```

The following command enables the smart tunnel auto sign-on list named HR and adds the domain named CISCO to the username during authentication:

```
hostname(config-group-webvpn) # smart-tunnel auto-signon enable HR domain CISCO
```


The following command removes the smart tunnel auto sign-on list named HR from the group policy and inherits the smart tunnel auto sign-on list command from the default group policy:

```
hostname(config-group-webvpn) # no smart-tunnel auto-signon enable HR
```

Automating Smart Tunnel Access

To start smart tunnel access automatically upon user login, enter the following command in group-policy webvpn configuration mode or username webvpn configuration mode:

smart-tunnel auto-start *list*

list is the name of the smart tunnel list already present in the security appliance webvpn configuration. You cannot assign more than smart tunnel list to a group policy or username. To view the smart tunnel list entries in the SSL VPN configuration, enter the **show running-config webvpn** command in privileged EXEC mode.

To remove the **smart-tunnel** command from the group policy or username and inherit the **[no]** **smart-tunnel** command from the default group-policy, use the **no** form of the command.

no smart-tunnel

The following commands assign the smart tunnel list named apps1 to the group policy:

```
hostname(config-group-policy) # webvpn
hostname(config-group-webvpn) # smart-tunnel auto-start apps1
```

Enabling and Disabling Smart Tunnel Access

By default, smart tunnels are disabled. If you enable smart tunnel access, the user will have to start it manually, using the **Application Access > Start Smart Tunnels** button on the clientless SSL VPN portal page. If you enter the **smart-tunnel auto-start *list*** command described in the previous section instead of the **smart-tunnel enable *list*** command, the user will not have to start smart tunnel access manually.

To enable smart tunnel access, enter the following command in group-policy webvpn configuration mode or username webvpn configuration mode:

smart-tunnel [enable *list* | disable]

list is the name of the smart tunnel list already present in the security appliance webvpn configuration. You cannot assign more than smart tunnel list to a group policy or username. To view the smart tunnel list entries in the SSL VPN configuration, enter the **show running-config webvpn** command in privileged EXEC mode.

To remove the **smart-tunnel** command from the group policy or local user policy, and inherit the **[no]** **smart-tunnel** command from the default group-policy, use the **no** form of the command.

no smart-tunnel

The following commands assign the smart tunnel list named apps1 to the group policy:

```
hostname(config-group-policy) # webvpn
hostname(config-group-webvpn) # smart-tunnel enable apps1
```

The following command disables smart tunnel access:

```
hostname(config-group-webvpn) # smart-tunnel disable
```

Configuring Port Forwarding

The following sections describe port forwarding and how to configure it:

- [About Port Forwarding](#)
- [Why Port Forwarding?](#)
- [Port Forwarding Requirements and Restrictions](#)
- [Configuring DNS for Port Forwarding](#)
- [Adding Applications to Be Eligible for Port Forwarding](#)
- [Assigning a Port Forwarding List](#)
- [Automating Port Forwarding](#)
- [Enabling and Disabling Port Forwarding](#)

About Port Forwarding

Port forwarding lets users access TCP-based applications over a clientless SSL VPN connection. Such applications include the following:

- Lotus Notes
- Microsoft Outlook
- Microsoft Outlook Express
- Perforce
- Sametime
- Secure FTP (FTP over SSH)
- SSH
- TELNET
- Windows Terminal Service
- XDDTS

Other TCP-based applications may also work, but we have not tested them. Protocols that use UDP do not work.

Why Port Forwarding?

Port forwarding is the legacy technology for supporting TCP-based applications over a clientless SSL VPN connection. You may choose to use port forwarding because you have built earlier configurations that support this technology.

Please consider the following alternatives to port forwarding:

- Smart tunnel access offers the following advantages to users:
 - Smart tunnel offers better performance than plug-ins.
 - Unlike port forwarding, smart tunnel simplifies the user experience by not requiring the user connection of the local application to the local port.
 - Unlike port forwarding, smart tunnel does not require users to have administrator privileges.

- Unlike port forwarding and smart tunnel access, a plug-in does not require the client application to be installed on the remote computer.

When configuring port forwarding on the security appliance, you specify the port the application uses. When configuring smart tunnel access, you specify the name of the executable file or its path.

Port Forwarding Requirements and Restrictions

The following restrictions apply to port forwarding:

- The remote host must be running a 32-bit version of one of the following:
 - Microsoft Windows Vista, Windows XP SP2 or SP3; or Windows 2000 SP4.
 - Apple Mac OS X 10.4 or 10.5 with Safari 2.0.4(419.3).
 - Fedora Core 4
- The remote host must also be running Sun JRE 1.5 or later.
- Browser-based users of Safari on Mac OS X 10.5.3 must identify a client certificate for use with the URL of the security appliance, once with the trailing slash and once without it, because of the way Safari interprets URLs. For example,
 - <https://example.com/>
 - <https://example.com>

For details, go to the [Safari, Mac OS X 10.5.3: Changes in client certificate authentication](#).

- Users of Microsoft Windows Vista who use port forwarding or smart tunnels must add the URL of the ASA to the Trusted Site zone. To access the Trusted Site zone, they must start Internet Explorer and choose the **Tools > Internet Options > Security** tab. Vista users can also disable Protected Mode to facilitate smart tunnel access; however, we recommend against this method because it increases the computer's vulnerability to attack.
- Port forwarding supports only TCP applications that use static TCP ports. Applications that use dynamic ports or multiple TCP ports are not supported. For example, SecureFTP, which uses port 22, works over clientless SSL VPN port forwarding, but standard FTP, which uses ports 20 and 21, does not.
- Port forwarding does not support protocols that use UDP.
- The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.
- A stateful failover does not retain sessions established using Application Access (either port forwarding or smart tunnel access). Users must reconnect following a failover.
- Port forwarding does not support connections to personal digital assistants.
- Because port forwarding requires downloading the Java applet and configuring the local client, and because doing so requires administrator permissions on the local system, it is unlikely that users will be able to use applications when they connect from public remote systems.



Caution

Make sure Sun Microsystems Java Runtime Environment (JRE) 1.5.x or later is installed on the remote computers to support port forwarding (application access) and digital certificates. If JRE 1.4.x is running and the user authenticates with a digital certificate, the application fails to start because JRE cannot access the web browser certificate store.

The Java applet displays in its own window on the end user HTML interface. It shows the contents of the list of forwarded ports available to the user, as well as which ports are active, and amount of traffic in bytes sent and received.

- Neither port forwarding nor the ASDM Java applet work with user authentication using digital certificates. Java does not have the ability to access the web browser keystore. Therefore Java cannot use certificates that the browser uses to authenticate users, and the application cannot start.
- The port forwarding applet displays the local port and the remote port as the same when the local IP address 127.0.0.1 is being used and cannot be updated by the clientless SSL VPN connection from the ASA. As a result, the ASA creates new IP addresses 127.0.0.2, 127.0.0.3, and so on for local proxy IDs. Because you can modify the hosts file and use different loopbacks, the remote port is used as the local port in the applet. To connect, you can use Telnet with the host name, without specifying the port. The correct local IP addresses are available in the local hosts file.

Configuring DNS for Port Forwarding

Port Forwarding forwards the domain name of the remote server or its IP address to the ASA for resolution and connection. In other words, the port forwarding applet accepts a request from the application and forwards it to the ASA. The ASA makes the appropriate DNS queries and establishes the connection on behalf of the port forwarding applet. The port forwarding applet only makes DNS queries to the ASA. It updates the host file so that when a port forwarding application attempts a DNS query, the query redirects to a loopback address. Configure the security appliance to accept the DNS requests from the port forwarding applet as follows:

- Step 1** Use the **dns server-group** command in global configuration mode to enter the dns server-group mode; then use the **domain-name** command to specify the domain name and **name-server** command to resolve the domain name to an IP address. The default setting of domain-name is DefaultDNS.

The following example configures a DNS server group named example.com:

```
hostname(config)# dns server-group example.com
hostname(config-dns-server-group)# domain-name example.com
hostname(config-dns-server-group)# name-server 192.168.10.10
```

- Step 2** (Required only if you are using a domain name other than the default one [DefaultDNS])—Use the **dns-group** command in tunnel-group webvpn configuration mode to specify the domain name the tunnel groups will use. By default, the security appliance assigns the DefaultWEBVPNGroup as the default tunnel group for clientless connections; follow this instruction if the security appliance uses that tunnel group to assign settings to the clientless connections. Otherwise, follow this step for each tunnel configured for clientless connections.

For example,

```
asa2(config-dns-server-group)# exit
asa2(config)# tunnel-group DefaultWEBVPNGroup webvpn-attributes
asa2(config-tunnel-webvpn)# dns-group example.com
```

Adding Applications to Be Eligible for Port Forwarding

The clientless SSL VPN configuration of each security appliance supports *port forwarding lists*, each of which specifies local and remote ports used by the applications for which you want to provide access. Because each group policy or username supports only one port forwarding list, you must group each set of applications to be supported into a list. To display the port forwarding list entries already present in the security appliance configuration, enter the following command in privileged EXEC mode:

show run webvpn port-forward

To add a port forwarding entry to a list, enter the following command in webvpn configuration mode:

port-forward {*list_name local_port remote_server remote_port description*}

list_name—Name for a set of applications (technically, a set of forwarded TCP ports) for users of clientless SSL VPN sessions to access. The security appliance creates a list using the name you enter if it does not recognize it. Otherwise, it adds the port forwarding entry to the list. Maximum 64 characters.

local_port—Port that listens for TCP traffic for an application running on the user's computer. You can use a local port number only once for each port forwarding list. Enter a port number in the range 1-65535 or port name. To avoid conflicts with existing services, use a port number greater than 1024.

remote_server—DNS name or IP address of the remote server for an application. The IP address can be in IPv4 or IPv6 format. We recommend a DNS name so that you do not have to configure the client applications for a specific IP address.

**Caution**

The DNS name must match the one assigned to the tunnel group to establish the tunnel and resolve to an IP address, per the instructions in the previous section. The default setting for both the **domain-name group** and **dns-group** commands described in that section is DefaultDNS.

remote_port—Port to connect to for this application on the remote server. This is the actual port the application uses. Enter a port number in the range 1-65535 or port name.

description—Application name or short description that displays on the end user Port Forwarding Java applet screen. Maximum 64 characters.

To remove an entry from a list, use the **no** form of the command, specifying both the list and the local port. In this case, the *remoteserver*, *remoteport*, and *description* are optional.

no port-forward *list_name local_port*

The following table shows the values used for example applications.

| Application | Local Port | Server DNS Name | Remote Port | Description |
|---------------|------------|-----------------|-------------|---------------|
| IMAP4S e-mail | 20143 | IMAP4Sserver | 143 | Get Mail |
| SMTPS e-mail | 20025 | SMTPSserver | 25 | Send Mail |
| DDTS over SSH | 20022 | DDTSserver | 22 | DDTS over SSH |
| Telnet | 20023 | Telnetserver | 23 | Telnet |

The following example shows how to create a port forwarding list called *SalesGroupPorts* that provides access to these applications:

```
hostname(config)# webvpn
hostname(config-webvpn)# port-forward SalesGroupPorts 20143 IMAP4Sserver 143 Get Mail
hostname(config-webvpn)# port-forward SalesGroupPorts 20025 SMTPSserver 25 Send Mail
hostname(config-webvpn)# port-forward SalesGroupPorts 20022 DDTSserver 22 DDTS over SSH
hostname(config-webvpn)# port-forward SalesGroupPorts 20023 Telnetserver 23 Telnet
```

Following the configuration of a port forwarding list and assign the list to group policies or usernames, as described in the next section.

Assigning a Port Forwarding List

For each group policy and username, you can configure clientless SSL VPN to do one of the following:

- Start port forwarding access automatically upon user login.
- Enable port forwarding access upon user login, but require the user to start it manually, using the **Application Access > Start Applications** button on the clientless SSL VPN Portal Page.



Note

These options are mutually exclusive for each group policy and username. Use only one.

Table 39-8 lists the **port-forward** commands available to each group policy and username. The configuration of each group policy and username supports only one of these commands at a time, so when you enter one, the security appliance replaces the one present in the configuration of the group policy or username in question with the new one, or in the case of the last command, simply removes the **port-forward** command from the group policy or username configuration.

Table 39-8 *group-policy and username webvpn port-forward Commands*

| Command | Description |
|---|--|
| port-forward auto-start <i>list_name</i> | Starts port forwarding automatically upon user login. |
| port-forward enable <i>list_name</i> | Enables port forwarding upon user login, but requires the user to start port forwarding manually, using the Application Access > Start Applications button on the clientless SSL VPN portal page. |
| port-forward disable | Prevents port forwarding. |
| no port-forward
[auto-start <i>list_name</i>
enable <i>list_name</i> disable] | Removes a port-forward command from the group policy or username configuration, which then inherits the [no] port-forward command from the default group-policy. The keywords following the no port-forward command are optional, however, they restrict the removal to the named port-forward command. |

For details, go to the section that addresses the option you want to use.

Automating Port Forwarding

To start port forwarding automatically upon user login, enter the following command in group-policy webvpn configuration mode or username webvpn configuration mode:

port-forward auto-start *list_name*

list_name names the port forwarding list already present in the security appliance webvpn configuration. You cannot assign more than one port forwarding list to a group policy or username. To display the port forwarding list entries present in the security appliance configuration, enter the **show run webvpn port-forward** command in privileged EXEC mode.

To remove the **port-forward** command from the group policy or username and inherit the [**no**] **port-forward** command from the default group-policy, use the **no** form of the command.

no port-forward

The following commands assign the port forwarding list named `apps1` to the group policy:

```
hostname(config-group-policy) # webvpn
hostname(config-group-webvpn) # port-forward auto-start apps1
```

Enabling and Disabling Port Forwarding

By default, port forwarding is disabled. If you enable port forwarding, the user will have to start it manually, using the **Application Access > Start Applications** button on the clientless SSL VPN portal page. If you enter the **port-forward auto-start *list_name*** command described in the previous section instead of the **port-forward enable *list_name*** command, the user will not have to start port forwarding manually to use it.

To enable or disable port forwarding, enter the following command in group-policy webvpn configuration mode or username webvpn configuration mode:

port-forward [**enable *list_name*** | **disable**]

list_name is the name of the port forwarding list already present in the security appliance webvpn configuration. You cannot assign more than one port forwarding list to a group policy or username. To view the port forwarding list entries, enter the **show running-config port-forward** command in privileged EXEC mode.

To remove the **port-forward** command from the group policy or username and inherit the **[no]** **port-forward** command from the default group-policy, use the **no** form of the command.

no port-forward

The following commands assign the port forwarding list named `apps1` to the group policy:

```
hostname(config-group-policy) # webvpn
hostname(config-group-webvpn) # port-forward enable apps1
```

The following command disables port forwarding:

```
hostname(config-group-webvpn) # port-forward disable
```

Application Access User Notes

The following sections provide information about using application access:

- [Using Application Access on Vista](#)
- [Closing Application Access to Prevent hosts File Errors](#)
- [Recovering from hosts File Errors When Using Application Access](#)



Note

The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. Neither the smart tunnel feature nor port forwarding supports MAPI. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.

Using Application Access on Vista

Users of Microsoft Windows Vista who use smart tunnels or port forwarding must add the URL of the ASA to the Trusted Site zone. To access the Trusted Site zone, they must start Internet Explorer and choose the Tools > Internet Options > Security tab. Vista users can also disable Protected Mode to facilitate smart tunnel access; however, we recommend against this method because it increases the computer's vulnerability to attack.

Closing Application Access to Prevent hosts File Errors

To prevent hosts file errors that can interfere with Application Access, close the Application Access window properly when you finish using Application Access. To do so, click the close icon.

Recovering from hosts File Errors When Using Application Access

The following errors can occur if you do not close the Application Access window properly:

- The next time you try to start Application Access, it might be disabled; you receive a Backup HOSTS File Found error message.
- The applications themselves might be disabled or might malfunction, even when you are running them locally.

These errors can result from terminating the Application Access window in any improper way. For example:

- Your browser crashes while you are using Application Access.
- A power outage or system shutdown occurs while you are using Application Access.
- You minimize the Application Access window while you are working, then shut down your computer with the window active (but minimized).

This section includes the following topics:

- [Understanding the hosts File](#)
- [Stopping Application Access Improperly](#)
- [Reconfiguring a hosts File Automatically Using Clientless SSL VPN](#)
- [Reconfiguring hosts File Manually](#)

Understanding the hosts File

The hosts file on your local system maps IP addresses to host names. When you start Application Access, clientless SSL VPN modifies the hosts file, adding clientless SSL VPN-specific entries. Stopping Application Access by properly closing the Application Access window returns the file to its original state.

| | |
|---------------------------------------|---|
| Before invoking Application Access... | hosts file is in original state. |
| When Application Access starts.... | <ul style="list-style-type: none"> • Clientless SSL VPN copies the hosts file to hosts.webvpn, thus creating a backup. • Clientless SSL VPN then edits the hosts file, inserting clientless SSL VPN-specific information. |

| | |
|---------------------------------------|---|
| When Application Access stops... | <ul style="list-style-type: none"> Clientless SSL VPN copies the backup file to the <code>hosts</code> file, thus restoring the hosts file to its original state. Clientless SSL VPN deletes <code>hosts.webvpn</code>. |
| After finishing Application Access... | hosts file is in original state. |

**Note**

Microsoft anti-spyware software blocks changes that the port forwarding Java applet makes to the hosts file. See www.microsoft.com for information on how to allow hosts file changes when using anti-spyware software.

Stopping Application Access Improperly

When Application Access terminates abnormally, the `hosts` file remains in a clientless SSL VPN-customized state. clientless SSL VPN checks the state the next time you start Application Access by searching for a `hosts.webvpn` file. If it finds one, a Backup HOSTS File Found error message (Figure 39-5) appears, and Application Access is temporarily disabled.

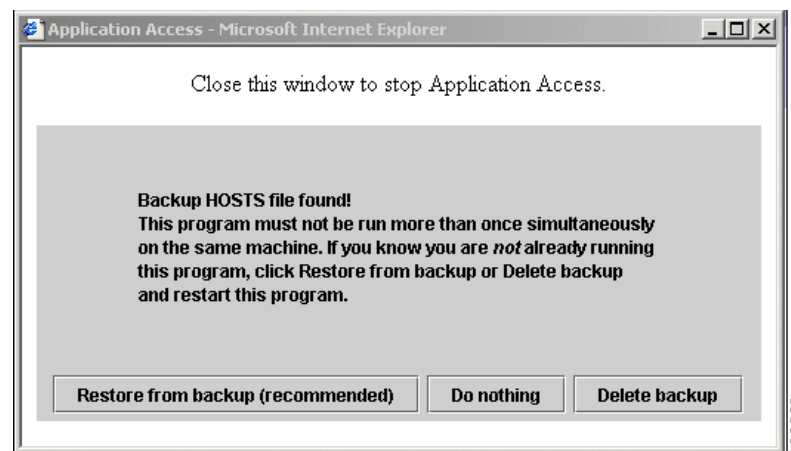
Once you shut down Application Access improperly, you leave your remote access client/server applications in limbo. If you try to start these applications without using clientless SSL VPN, they might malfunction. You might find that hosts that you normally connect to are unavailable. This situation could commonly occur if you run applications remotely from home, fail to quit the Application Access window before shutting down the computer, then try to run the applications later from the office.

Reconfiguring a hosts File Automatically Using Clientless SSL VPN

If you are able to connect to your remote access server, follow these steps to reconfigure the hosts file and reenale both Application Access and the applications.

- Step 1** Start clientless SSL VPN and log in. The home page opens.
- Step 2** Click the **Applications Access** link. A Backup HOSTS File Found message appears. (See Figure 39-5.)

Figure 39-5 Backup HOSTS File Found Message



- Step 3** Choose one of the following options:

- **Restore from backup**—Clientless SSL VPN forces a proper shutdown. It copies the `hosts.webvpn` backup file to the `hosts` file, restoring it to its original state, then deletes `hosts.webvpn`. You then have to restart Application Access.
- **Do nothing**—Application Access does not start. The remote access home page reappears.
- **Delete backup**—Clientless SSL VPN deletes the `hosts.webvpn` file, leaving the `hosts` file in its clientless SSL VPN-customized state. The original `hosts` file settings are lost. Application Access then starts, using the clientless SSL VPN-customized `hosts` file as the new original. Choose this option only if you are unconcerned about losing `hosts` file settings. If you or a program you use might have edited the `hosts` file after Application Access has shut down improperly, choose one of the other options, or edit the `hosts` file manually. (See “[Reconfiguring hosts File Manually](#).”)

Reconfiguring hosts File Manually

If you are not able to connect to your remote access server from your current location, or if you have customized the `hosts` file and do not want to lose your edits, follow these steps to reconfigure the `hosts` file and reenable both Application Access and the applications.

Step 1 Locate and edit your `hosts` file. The most common location is `c:\windows\system32\drivers\etc\hosts`.

Step 2 Check to see if any lines contain the string: `# added by WebVpnPortForward`. If any lines contain this string, your `hosts` file is clientless SSL VPN-customized. If your `hosts` file is clientless SSL VPN-customized, it looks similar to the following example:

```
123.0.0.3 server1 # added by WebVpnPortForward
123.0.0.3 server1.example.com vpn3000.com # added by WebVpnPortForward
123.0.0.4 server2 # added by WebVpnPortForward
123.0.0.4 server2.example.com.vpn3000.com # added by WebVpnPortForward
123.0.0.5 server3 # added by WebVpnPortForward
123.0.0.5 server3.example.com vpn3000.com # added by WebVpnPortForward

# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97      cisco.example.com          # source server
#       38.25.63.10     x.example.com              # x client host
#
123.0.0.1      localhost
```

Step 3 Delete the lines that contain the string: `# added by WebVpnPortForward`

Step 4 Save and close the file.

Step 5 Start clientless SSL VPN and log in.

The home page appears.

Step 6 Click the Application Access link.

The Application Access window appears. Application Access is now enabled.

Configuring File Access

Clientless SSL VPN serves remote users with HTTPS portal pages that interface with proxy CIFS and/or FTP clients running on the security appliance. Using either CIFS or FTP, clientless SSL VPN provides users with network access to the files on the network, to the extent that the users meet user authentication requirements and the file properties do not restrict access. The CIFS and FTP clients are transparent; the portal pages delivered by clientless SSL VPN provide the appearance of direct access to the file systems.

When a user requests a list of files, clientless SSL VPN queries the server designated as the master browser for the IP address of the server containing the list. The security appliance gets the list and delivers it to the remote user on a portal page.

Clientless SSL VPN lets the user invoke the following CIFS and FTP functions, depending on user authentication requirements and file properties:

- Navigate and list domains and workgroups, servers within a domain or workgroup, shares within a server, and files within a share or directory
- Create directories
- Download, upload, rename, move, and delete files

The security appliance uses a master browser, WINS server, or DNS server, typically on the same network as the security appliance or reachable from that network, to query the network for a list of servers when the remote user clicks Browse Networks in the menu of the portal page or on the toolbar displayed during the clientless SSL VPN session.

The master browser or DNS server provides the CIFS/FTP client on the security appliance with a list of the resources on the network, which clientless SSL VPN serves to the remote user.



Note

Before configuring file access, you must configure the shares on the servers for user access.

CIFS File Access Requirement

To access `\\server\share\subfolder\personal` folder, the user must have list permission for all points above `personal` folder.

Adding Support for File Access

Configure file access as follows:



Note

Step 1 of this procedure describes how to specify the master browser and WINS servers. As an alternative, you can use ASDM to configure URL lists and entries that provide access to file shares.

Adding a share in ASDM does not require a master browser or a WINS server. However, it does not

provide support for the Browse Networks link. You can use a hostname or an IP address to refer to ServerA when entering this command. If you use a hostname, the security appliance requires a DNS server to resolve it to an IP address.

- Step 1** Use the **nbns-server** command in tunnel-group webvpn configuration mode once for each NetBIOS Name Server (NBNS). This step lets you browse a network or domain.

nbns-server {IPaddress | hostname} [master] [timeout timeout] [retry retries]

master is the computer designated as the master browser. The master browser maintains the list of computers and shared resources. Any NBNS server you identify with this command without entering the master portion of the command must be a Windows Internet Naming Server (WINS). Specify the master browser first, then specify the WINS servers. You can specify up to three servers, including the master browser, for a connection profile.

retries is the number of times to retry queries to the NBNS server. The security appliance recycles through the list of servers this number of times before sending an error message. The default value is 2; the range is 1 through 10.

timeout is the number of seconds the security appliance waits before sending the query again, to the same server if it is the only one, or another server if there are more than one. The default timeout is 2 seconds; the range is 1 to 30 seconds.

For example,

```
hostname(config-tunnel-webvpn)# nbns-server 192.168.1.20 master
hostname(config-tunnel-webvpn)# nbns-server 192.168.1.41
hostname(config-tunnel-webvpn)# nbns-server 192.168.1.47
```



Note Use the **show tunnel-group webvpn-attributes** command if you want to display the NBNS servers already present in the connection profile configuration.

- Step 2** (Optional) Use the **character-encoding** command to specify the character set to encode in clientless SSL VPN portal pages to be delivered to remote users. By default, the encoding type set on the remote browser determines the character set for clientless SSL VPN portal pages, so you need to set the character encoding only if it is necessary to ensure proper encoding on the browser.

character-encoding charset

Charset is a string consisting of up to 40 characters, and equal to one of the valid character sets identified in <http://www.iana.org/assignments/character-sets>. You can use either the name or the alias of a character set listed on that page. Examples include iso-8859-1, shift_jis, and ibm850.



Note The character-encoding and file-encoding values do not exclude the font family to be used by the browser. You need to complement the setting of one these values with the **page style** command in webvpn customization command mode to replace the font family if you are using Japanese Shift_JIS character encoding, as shown in the following example, or enter the **no page style** command in webvpn customization command mode to remove the font family.

The following example sets the character-encoding attribute to support Japanese Shift_JIS characters, removes the font family, and retains the default background color:

```
hostname(config-webvpn)# character-encoding shift_jis
hostname(config-webvpn)# customization DfltCustomization
hostname(config-webvpn-custom)# page style background-color:white
```

- Step 3** (Optional) Use the **file-encoding** command to specify the encoding for clientless SSL VPN portal pages from specific CIFS servers. Thus, you can use different file-encoding values for CIFS servers that require different character encodings.

file-encoding {server-name | server-ip-address} charset

The following example sets the file-encoding attribute of the CIFS server 10.86.5.174 to support IBM860 (alias “CP860”) characters:

```
hostname(config-webvpn)# file-encoding 10.86.5.174 cp860
```

For a complete description of these commands, see the *Cisco Security Appliance Command Reference*.

Ensuring Clock Accuracy for SharePoint Access

The clientless SSL VPN server on the security appliance uses cookies to interact with applications such as Microsoft Word on the endpoint. The cookie expiration time set by the security appliance can cause Word to malfunction when accessing documents on a SharePoint server if the time on the security appliance is incorrect. To prevent this malfunction, set the ASA clock properly. We recommend configuring the security appliance to dynamically synchronize the time with an NTP server. For instructions, see “[Setting the Date and Time](#).”

Using Clientless SSL VPN with PDAs

You can access clientless SSL VPN from your Pocket PC or other certified personal digital assistant device. Neither the security appliance administrator nor the Clientless SSL VPN user need do anything special to use clientless SSL VPN with a certified PDA.

Cisco has certified the following PDA platform:

HP iPaq H4150
Pocket PC 2003
Windows CE 4.20.0, build 14053
Pocket Internet Explorer (PIE)
ROM version 1.10.03ENG
ROM Date: 7/16/2004

Some differences in the PDA version of clientless SSL VPN exist:

- A banner web page replaces the popup clientless SSL VPN window.
- An icon bar replaces the standard clientless SSL VPN floating toolbar. This bar displays the Go, Home and Logout buttons.
- The Show Toolbar icon is not included on the main clientless SSL VPN portal page.
- Upon clientless SSL VPN logout, a warning message provides instructions for closing the PIE browser properly. If you do not follow these instructions and you close the browser window in the common way, PIE does not disconnect from clientless SSL VPN or any secure website that uses HTTPS.
- Clientless SSL VPN supports OWA 2000 and OWA 2003 Basic Authentication. If Basic Authentication is not configured on an OWA server and a clientless SSL VPN user attempts to access that server, access is denied.

- Unsupported clientless SSL VPN features:
 - Application Access and other Java-dependent features.
 - HTTP proxy.
 - Cisco Secure Desktop provides limited support for Microsoft Windows CE.
 - Microsoft Outlook Web Access (OWA) 5.5.
 - The Citrix Metaframe feature (if the PDA does not have the corresponding Citrix ICA client software).

Using E-Mail over Clientless SSL VPN

Clientless SSL VPN supports several ways to access e-mail. This section includes the following methods:

- [Configuring E-mail Proxies](#)
- [Configuring Web E-mail: MS Outlook Web Access](#)

Configuring E-mail Proxies

Clientless SSL VPN supports IMAP4S, POP3S, and SMTPS e-mail proxies. [Table 39-9](#) lists attributes that apply globally to e-mail proxy users:

Table 39-9 *Attributes for E-mail Proxy Users over Clientless SSL VPN*

| Function | Command | Default Value |
|---|------------------------------------|--|
| Specifies the previously configured accounting servers to use with e-mail proxy. | accounting-server-group | None |
| Specifies the authentication method(s) for e-mail proxy users. | authentication | IMAP4S: Mailhost (required)
POP3S Mailhost (required)
SMTPS: AAA |
| Specifies the previously configured authentication servers to use with e-mail proxy. | authentication-server-group | LOCAL |
| Specifies the previously configured authorization servers to use with clientless SSL VPN. | authorization-server-group | None |
| Requires users to authorize successfully to connect. | authorization-required | Disabled |
| Identifies the DN of the peer certificate to use as a username for authorization. | authorization-dn-attributes | Primary attribute: CN
Secondary attribute: OU |
| Specifies the name of the group policy to use. | default-group-policy | DfltGrpPolicy |
| Enables e-mail proxy on the specified interface. | enable | Disabled |
| Defines the separator between the e-mail and VPN usernames and passwords. | name-separator | “:” (colon) |
| Configures the maximum number of outstanding non-authenticated sessions. | outstanding | 20 |

Table 39-9 **Attributes for E-mail Proxy Users over Clientless SSL VPN**

| Function | Command | Default Value |
|--|-------------------------|---|
| Sets the port the e-mail proxy listens to. | port | IMAP4S:993
POP3S: 995
SMTPS: 988 ¹ |
| Specifies the default e-mail server. | server | None. |
| Defines the separator between the e-mail and server names. | server-separator | “@” |

1. With the Eudora e-mail client, SMTPS works only on port 465, even though the default port for SMTPS connections is 988.

E-mail Proxy Certificate Authentication

E-mail clients such as MS Outlook, MS Outlook Express, and Eudora lack the ability to access the certificate store.

Configuring Web E-mail: MS Outlook Web Access

The adaptive security appliance supports Microsoft Outlook Web Access to Exchange Server 2000, 2003, and 2007. It requires that users perform the following tasks:

- Enter the URL of the mail server in a browser in your clientless SSL VPN session.
- When prompted, enter the e-mail server username in the format *domain\username*.
- Enter the e-mail password.

Optimizing Clientless SSL VPN Performance

The security appliance provides several ways to optimize clientless SSL VPN performance and functionality. Performance improvements include caching and compressing web objects. Functionality tuning includes setting limits on content transformation and proxy-bypass. APCF provides an additional method of tuning content transformation. The following sections explain these features:

- [Configuring Caching](#)
- [Configuring Content Transformation](#)

Configuring Caching

Caching enhances clientless SSL VPN performance. It stores frequently reused objects in the system cache, which reduces the need to perform repeated rewriting and compressing of content. It reduces traffic between clientless SSL VPN and the remote servers, with the result that many applications run much more efficiently.

By default, caching is enabled. You can customize the way caching works for your environment by using the caching commands in cache mode, which you enter from webvpn mode, as in the following example.

```
hostname(config)#
hostname(config)# webvpn
hostname(config-webvpn)# cache
```

A list of caching commands and their functions follows:

| Cache Command | Function |
|-----------------------------|--|
| disable | Disables caching. |
| expiry-time | Configures an expiration time for caching objects. |
| lmfactor | Configures terms for revalidating cached objects. |
| max-object-size | Sets a maximum size for objects to cache. |
| min-object-size | Sets a minimum size for objects to cache. |
| cache-static-content | Caches all cacheable web objects, content not subject to rewriting. Examples include images and PDF files. |

Configuring Content Transformation

By default, the security appliance processes all clientless SSL VPN traffic through a content transformation/rewriting engine that includes advanced elements such as JavaScript and Java to proxy HTTP traffic that may have different semantics and access control rules depending on whether the user is accessing an application within or independently of an SSL VPN device.

Some web resources require highly individualized treatment. The following sections describe functionality that provides such treatment:

- [Configuring a Certificate for Signing Rewritten Java Content](#)
- [Disabling Content Rewrite](#)
- [Using Proxy Bypass](#)
- [Configuring Application Profile Customization Framework](#)

Subject to the requirements of your organization and the web content involved, you might use one of these features.

Configuring a Certificate for Signing Rewritten Java Content

Java objects which have been transformed by clientless SSL VPN can subsequently be signed using a PKCS12 digital certificate associated with a trustpoint. You import and employ the certificate using a combination of the **crypto ca import** and **java-trustpoint** commands.

The following example commands show the creation of a trustpoint named mytrustpoint and its assignment to signing Java objects:

```
hostname(config)# crypto ca import mytrustpoint pkcs12 mypassphrase
Enter the base 64 encoded PKCS12.
End with the word "quit" on a line by itself.
[ PKCS12 data omitted ]
quit
INFO: Import PKCS12 operation completed successfully.
hostname(config)# webvpn
hostname(config)# java-trustpoint mytrustpoint
```


Disabling Content Rewrite

You might not want some applications and web resources, for example, public websites, to go through the security appliance. The security appliance therefore lets you create rewrite rules that let users browse certain sites and applications without going through the security appliance. This is similar to split-tunneling in an IPsec VPN connection.

Use the **rewrite** command with the **disable** option in webvpn mode to specify applications and resources to access outside a clientless SSL VPN tunnel.

You can use the rewrite command multiple times. The order number of rules is important because the security appliance searches rewrite rules by order number, starting with the lowest, and applies the first rule that matches.

Using Proxy Bypass

You can configure the security appliance to use proxy bypass when applications and web resources work better with the special content rewriting this feature provides. Proxy bypass is an alternative method of content rewriting that makes minimal changes to the original content. It is often useful with custom web applications.

You can use this command multiple times. The order in which you configure entries is unimportant. The interface and path mask or interface and port uniquely identify a proxy bypass rule.

If you configure proxy bypass using ports rather than path masks, depending on your network configuration, you might need to change your firewall configuration to allow these ports access to the security appliance. Use path masks to avoid this restriction. Be aware, however, that path masks can change, so you might need to use multiple pathmask statements to exhaust the possibilities.

A path is everything in a URL after the .com or .org or other types of domain name. For example, in the URL `www.mycompany.com/hrbenefits`, *hrbenefits* is the path. Similarly, for the URL `www.mycompany.com/hrinsurance`, *hrinsurance* is the path. If you want to use proxy bypass for all hr sites, you can avoid using the command multiple times by using the * wildcard as follows: `/hr*`.

To configure proxy bypass, use the **proxy-bypass** command in webvpn mode.

Configuring Application Profile Customization Framework

An Apcf profile for clientless SSL VPN lets the security appliance handle non-standard applications and web resources so that they display correctly over a clientless SSL VPN connection. An Apcf profile contains a script that specifies when (pre, post), where (header, body, request, response), and what data to transform for a particular application. The script is in XML and uses sed (stream editor) syntax for string/text transformation. Multiple Apcf profiles can run in parallel on a security appliance. Within an Apcf profile script, multiple Apcf rules can apply. In this case, the security appliance processes the oldest rule first (based on configuration history), then the next oldest rule, and so forth.

You can store Apcf profiles on the security appliance flash memory, or on an HTTP, HTTPS, or TFTP server. Use the **apcf** command in webvpn mode to identify and locate an Apcf profile that you want to load on the security appliance.



Note

We recommend that you configure an Apcf profile only with the assistance of Cisco personnel.

The following example shows how to enable an Apcf profile named `apcf1.xml`, located on flash memory.

```
hostname(config)# webvpn
```

```
hostname(config-webvpn)# apcf flash:/apcf/apcf1.xml
```

This example shows how to enable an APCF profile named apcf2.xml, located on an https server called myserver, port 1440 with the path being /apcf.

```
hostname(config)# webvpn
hostname(config-webvpn)# apcf https://myserver:1440/apcf/apcf2.xml
```

APCF Syntax



Caution

Misuse of an APCF profile can result in reduced performance and undesired rendering of content. In most cases, Cisco Engineering supplies APCF profiles to solve specific application rendering issues.

APCF profiles use XML format, and sed script syntax, with the XML tags in [Table 39-10](#).

Table 39-10 **APCF XML Tags**

| Tag | Use |
|--|---|
| <APCF>...</APCF> | The mandatory root element that opens any APCF XML file. |
| <version>1.0</version> | The mandatory tag that specifies the APCF implementation version. Currently the only version is 1.0. |
| <application>...</application> | The mandatory tag that wraps the body of the XML description. |
| <id> text </id> | The mandatory tag that describes this particular APCF functionality. |
| <apcf-entities>...</apcf-entities> | The mandatory tag that wraps a single or multiple APCF entities. |
| <js-object>...</js-object>
<html-object>...</html-object>
<process-request-header>...</process-request-header>
<process-response-header>...</process-response-header>
<preprocess-response-body>...</preprocess-response-body>
<postprocess-response-body>...</postprocess-response-body> | One of these tags specifying type of content or the stage at which the APCF processing should take place is required. |

Table 39-10 **APCF XML Tags (continued)**

| Tag | Use |
|--|---|
| <code><conditions>... </conditions></code> | <p>A child element of the pre/post-process tags that specifies criteria for processing such as:</p> <ul style="list-style-type: none"> <code>http-version</code> (such as 1.1, 1.0, 0.9) <code>http-method</code> (get, put, post, webdav) <code>http-scheme</code> (http, https, other) <code>server-regexp</code> regular expression containing ("a".. "z" "A".. "Z" "0".. "9" ".-_*[]?") <code>server-fnmatch</code> (regular expression containing ("a".. "z" "A".. "Z" "0".. "9" ".-_*[]?+()\{\},")), <code>user-agent-regexp</code> <code>user-agent-fnmatch</code> <code>request-uri-regexp</code> <code>request-uri-fnmatch</code> <p>If more than one of condition tags is present, the security appliance performs a logical AND for all tags.</p> |
| <code><action> ... </action></code> | <p>Wraps one or more actions to perform on the content under specified conditions; you can use the following tags to define these actions (shown below): <code><do></code>, <code><sed-script></code>, <code><rewrite-header></code>, <code><add-header></code>, <code><delete-header></code>.</p> |
| <code><do>...</do></code> | <p>Child element of the action tag used to define one of the following actions:</p> <ul style="list-style-type: none"> <code><no-rewrite/></code>—Do not mangle the content received from the remote server. <code><no-toolbar/></code>—Do not insert the toolbar. <code><no-gzip/></code>—Do not compress the content. <code><force-cache/></code>—Preserve the original caching instructions. <code><force-no-cache/></code>—Make object non-cacheable. <code><downgrade-http-version-on-backend></code>—Use HTTP/1.0 when sending the request to remote server. |
| <code><sed-script> TEXT </sed-script></code> | <p>Child element of the action tag used to change the content of text-based objects. The Text must be a valid Sed script. The <code><sed-script></code> applies to the <code><conditions></code> tag defined before it.</p> |
| <code><rewrite-header></rewrite-header></code> | <p>Child element of the action tag. Changes the value of the HTTP header specified in the child element <code><header></code> tag shown below.</p> |
| <code><add-header></add-header></code> | <p>Child element of the action tag used to add a new HTTP header specified in the child element <code><header></code> tag shown below.</p> |

Table 39-10 **APCF XML Tags (continued)**

| Tag | Use |
|--|--|
| <code><delete-header></delete-header></code> | Child element of the action tag used to delete the specified HTTP header specified by the child element <code><header></code> tag shown below. |
| <code><header></header></code> | Specifies the name HTTP header to be rewritten, added, or deleted. For example, the following tag changes the value of the HTTP header named Connection:

<pre> <rewrite-header> <header>Connection</header> <value>close</value> </rewrite-header> </pre> |

APCF Example 1

```

<APCF>
<version>1.0</version>
<application>
  <id>Do not compress content from notsogood.com</id>
  <apcf-entities>
    <process-request-header>
      <conditions>
        <server-fnmatch>*.notsogood.com</server-fnmatch>
      </conditions>
      <action>
        <do><no-gzip/></do>
      </action>
    </process-request-header>
  </apcf-entities>
</application>
</APCF>

```

APCF Example 2

```

<APCF>
<version>1.0</version>
<application>
  <id>Change MIME type for all .xyz objects</id>
  <apcf-entities>
    <process-response-header>
      <conditions>
        <request-uri-fnmatch>*.xyz</request-uri-fnmatch>
      </conditions>
      <action>
        <rewrite-header>
          <header>Content-Type</header>
          <value>text/html</value>
        </rewrite-header>
      </action>
    </process-response-header>
  </apcf-entities>
</application>
</APCF>

```

Clientless SSL VPN End User Setup

This section is for the system administrator who sets up clientless SSL VPN for end users. It describes how to customize the end-user interface.

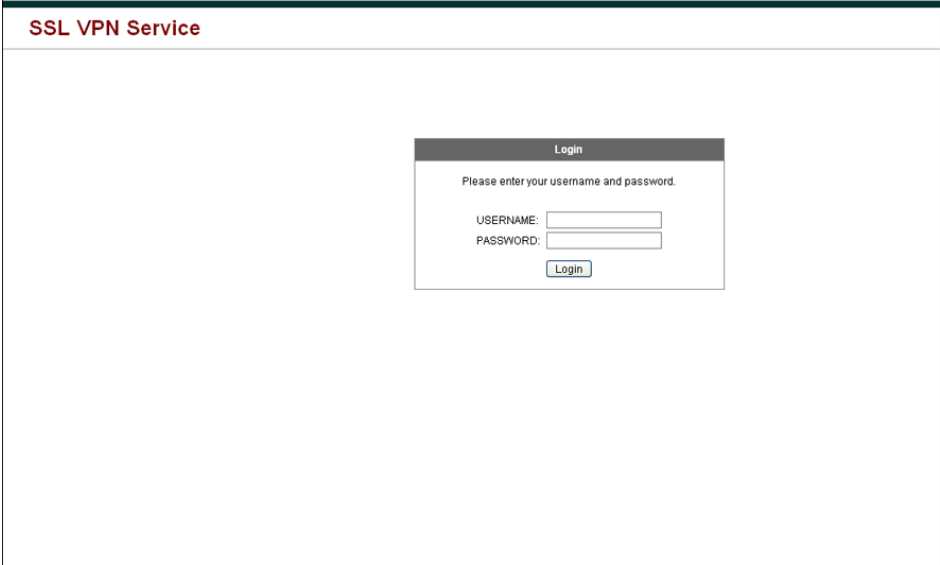
This section summarizes configuration requirements and tasks for a remote system. It specifies information to communicate to users to get them started using clientless SSL VPN. It includes the following topics:

- [Defining the End User Interface](#)
- [Customizing Clientless SSL VPN Pages, page 39-63](#)
- [Customizing Help, page 39-76](#)
- [Requiring Usernames and Passwords](#)
- [Communicating Security Tips](#)
- [Configuring Remote Systems to Use Clientless SSL VPN Features](#)
- [Translating the Language of User Messages](#)

Defining the End User Interface

The clientless SSL VPN end user interface consists of a series of HTML panels. A user logs on to clientless SSL VPN by entering the IP address of a security appliance interface in the format `https://address`. The first panel that displays is the login screen ([Figure 39-6](#)).

Figure 39-6 *Clientless SSL VPN Login Screen*



Viewing the Clientless SSL VPN Home Page

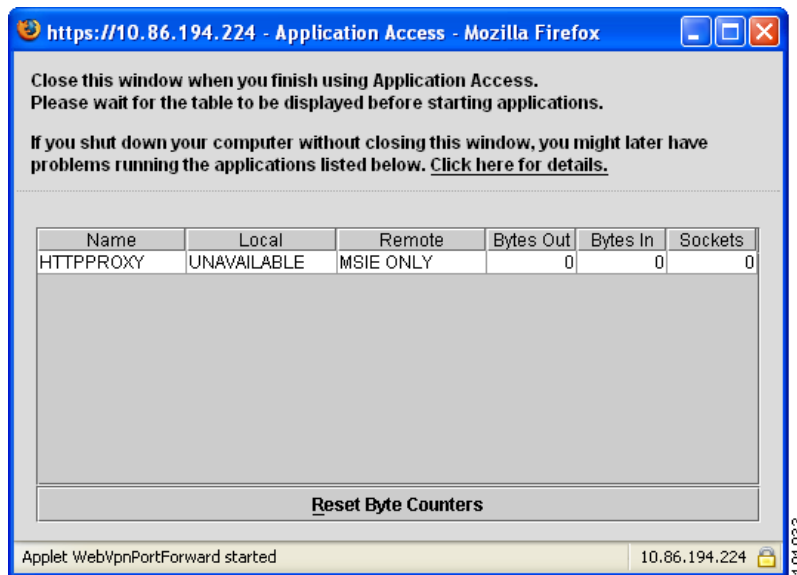
After the user logs in, the portal page opens.

The home page displays all of the clientless SSL VPN features you have configured, and its appearance reflects the logo, text, and colors you have selected. This sample home page includes all available clientless SSL VPN features with the exception of identifying specific file shares. It lets users browse the network, enter URLs, access specific websites, and use Application Access (port forwarding and smart tunnels) to access TCP applications.

Viewing the Clientless SSL VPN Application Access Panel

To start port forwarding or smart tunnels, a user clicks the Go button in the Application Access box. The Application Access window opens (Figure 39-7).

Figure 39-7 Clientless SSL VPN Application Access Window



This window displays the TCP applications configured for this clientless SSL VPN connection. To use an application with this panel open, the user starts the application in the normal way.



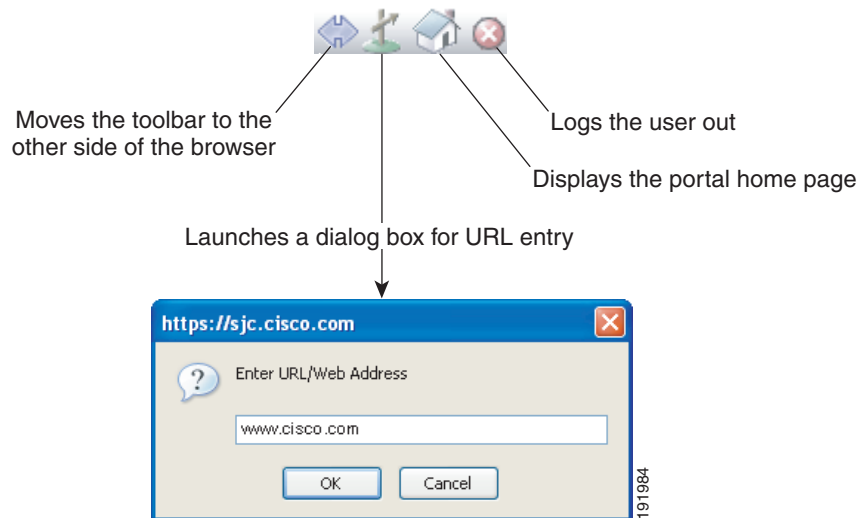
Note

A stateful failover does not retain sessions established using Application Access. Users must reconnect following a failover.

Viewing the Floating Toolbar

The floating toolbar shown in [Figure 39-8](#) represents the current clientless SSL VPN session.

Figure 39-8 *Clientless SSL VPN Floating Toolbar*



Be aware of the following characteristics of the floating toolbar:

- The toolbar lets you enter URLs, browse file locations, and choose preconfigured web connections without interfering with the main browser window.
- If you configure your browser to block popups, the floating toolbar cannot display.
- If you close the toolbar, the security appliance prompts you to confirm that you want to end the clientless SSL VPN session.

See [Table 39-13 on page 39-80](#) for detailed information about using clientless SSL VPN.

Customizing Clientless SSL VPN Pages

You can change the appearance of the portal pages displayed to clientless SSL VPN users. This includes the Login page displayed to users when they connect to the security appliance, the Home page displayed to users after the security appliance authenticates them, the Application Access window displayed when users launch an application, and the Logout page displayed when users logout of clientless SSL VPN sessions.

After you customize the portal pages, you can save your customization and apply it to a specific connection profile, group policy, or user. You can create and save many customization objects, enabling the security appliance to change the appearance of portal pages for individual users or groups of users.

This section contains the following topics and tasks:

- [How Customization Works, page 39-64](#)
- [Exporting a Customization Template, page 39-64](#)
- [Editing the Customization Template, page 39-64](#)
- [Importing a Customization Object, page 39-70](#)

- [Applying Customizations to Connection Profiles, Group Policies and Users, page 39-71](#)
- [Login Screen Advanced Customization, page 39-72](#)

How Customization Works

The security appliance uses customization objects to define the appearance of user screens. A customization object is compiled from an XML file which contains XML tags for all the customizable screen items displayed to remote users. The security appliance software contains a customization template that you can export to a remote PC. You can edit this template and import the template back into the security appliance as a new customization object.

When you export a customization object, an XML file containing XML tags is created at the URL you specify. The XML file created by the customization object named *Template* contains empty XML tags, and provides the basis for creating new customization objects. This object cannot be changed or deleted from cache memory, but can be exported, edited, and imported back into the security appliance as a new customization object.

Customization Objects, Connection Profiles, and Group Policies

Initially, when a user first connects, the default customization object (named *DfltCustomization*) identified in the connection profile (tunnel group) determines how the logon screen appears. If the connection profile list is enabled, and the user selects a different group, and that group has its own customization, the screen changes to reflect the customization object for that new group.

After the remote user is authenticated, the screen appearance is determined by whether a customization object that has been assigned to the group policy.

Exporting a Customization Template

When you export a customization object, an XML file is created at the URL you specify. The customization template (named *Template*) contains empty XML tags, and provides the basis for creating new customization objects. This object cannot be changed or deleted from cache memory, but can be exported, edited, and imported back into the security appliance as a new customization object.

You can export a customization object using the **export webvpn customization** command, make changes to the XML tags, and import the file as a new object using the **import webvpn customization** command.

The following example exports the default customization object (DfltCustomization) and creates the XML file named *dflt_custom*:

```
hostname# export webvpn customization DfltCustomization tftp://209.165.200.225/dflt_custom
!!!!!!!!!!!!!!!!!!!!!!INFO: Customization object 'DfltCustomization' was exported to
tftp://10.86.240.197/dflt_custom
hostname#
```

Editing the Customization Template

This section shows the contents of the customization template and has convenient figures to help you quickly choose the correct XML tag and make changes that affect the screens.

You can use a text editor or an XML editor to edit the XML file. The following example shows the XML tags of the customization template. Some redundant tags have been removed for easier viewing:

```
<custom>
  <localization>
    <languages>en, ja, zh, ru, ua</languages>
```



```

    <default-language>en</default-language>
  </localization>
  <auth-page>
    <window>
      <title-text l10n="yes"><![CDATA[SSL VPN Service]]></title-text>
    </window>
    <full-customization>
      <mode>disable</mode>
      <url></url>
    </full-customization>
    <language-selector>
      <mode>disable</mode>
      <title l10n="yes">Language:</title>
      <language>
        <code>en</code>
        <text>English</text>
      </language>
      <language>
        <code>zh</code>
        <text>ä¸­æ–›ä½­ (Chinese)</text>
      </language>
      <language>
        <code>ja</code>
        <text>æ—æœ­èª­ (Japanese)</text>
      </language>
      <language>
        <code>ru</code>
        <text>Ð½ÑÑÐ°Ð½Ð° (Russian)</text>
      </language>
      <language>
        <code>ua</code>
        <text>Ð£Ð°ÐºÑ°Ð½Ð° Ð°Ð½Ð°Ð½Ð° (Ukrainian)</text>
      </language>
    </language-selector>
    <logon-form>
      <title-text l10n="yes"><![CDATA[Login]]></title-text>
      <title-background-color><![CDATA[#666666]]></title-background-color>
      <title-font-color><![CDATA[#ffffff]]></title-font-color>
      <message-text l10n="yes"><![CDATA[Please enter your username and
password.]]></message-text>
      <username-prompt-text l10n="yes"><![CDATA[USERNAME:]]></username-prompt-text>
      <password-prompt-text l10n="yes"><![CDATA[PASSWORD:]]></password-prompt-text>
      <internal-password-prompt-text l10n="yes">Internal
Password:</internal-password-prompt-text>
      <internal-password-first>no</internal-password-first>
      <group-prompt-text l10n="yes"><![CDATA[GROUP:]]></group-prompt-text>
      <submit-button-text l10n="yes"><![CDATA[Login]]></submit-button-text>
      <title-font-color><![CDATA[#ffffff]]></title-font-color>
      <title-background-color><![CDATA[#666666]]></title-background-color>
      <font-color>#000000</font-color>
      <background-color>#ffffff</background-color>
      <border-color>#858A91</border-color>
    </logon-form>
    <logout-form>
      <title-text l10n="yes"><![CDATA[Logout]]></title-text>
      <message-text l10n="yes"><![CDATA[Goodbye.<br>

For your own security, please:<br>

<li>Clear the browser's cache

<li>Delete any downloaded files

<li>Close the browser's window]]></message-text>

```

```

    <login-button-text l10n="yes">Logon</login-button-text>
    <hide-login-button>no</hide-login-button>
    <title-background-color><![CDATA[#666666]]></title-background-color>
    <title-font-color><![CDATA[#ffffff]]></title-font-color>
    <title-font-color><![CDATA[#ffffff]]></title-font-color>
    <title-background-color><![CDATA[#666666]]></title-background-color>
    <font-color>#000000</font-color>
    <background-color>#ffffff</background-color>
    <border-color>#858A91</border-color>
</logout-form>
<title-panel>
    <mode>enable</mode>
    <text l10n="yes"><![CDATA[SSL VPN Service]]></text>
    <logo-url l10n="yes">/+CSCOU+/cisco_logo.gif</logo-url>
    <gradient>yes</gradient>
    <style></style>
    <background-color><![CDATA[#ffffff]]></background-color>
    <font-size><![CDATA[larger]]></font-size>
    <font-color><![CDATA[#800000]]></font-color>
    <font-weight><![CDATA[bold]]></font-weight>
</title-panel>
<info-panel>
    <mode>disable</mode>
    <image-url l10n="yes">/+CSCOU+/clear.gif</image-url>
    <image-position>above</image-position>
    <text l10n="yes"></text>
</info-panel>
<copyright-panel>
    <mode>disable</mode>
    <text l10n="yes"></text>
</copyright-panel>
</auth-page>
<portal>
    <title-panel>
        <mode>enable</mode>
        <text l10n="yes"><![CDATA[SSL VPN Service]]></text>
        <logo-url l10n="yes">/+CSCOU+/cisco_logo.gif</logo-url>
        <gradient>yes</gradient>
        <style></style>
        <background-color><![CDATA[#ffffff]]></background-color>
        <font-size><![CDATA[larger]]></font-size>
        <font-color><![CDATA[#800000]]></font-color>
        <font-weight><![CDATA[bold]]></font-weight>
    </title-panel>
    <browse-network-title l10n="yes">Browse Entire Network</browse-network-title>
    <access-network-title l10n="yes">Start AnyConnect</access-network-title>
    <application>
        <mode>enable</mode>
        <id>home</id>
        <tab-title l10n="yes">Home</tab-title>
        <order>1</order>
    </application>
    <application>
        <mode>enable</mode>
        <id>web-access</id>
        <tab-title l10n="yes"><![CDATA[Web Applications]]></tab-title>
        <url-list-title l10n="yes"><![CDATA[Web Bookmarks]]></url-list-title>
        <order>2</order>
    </application>
    <application>
        <mode>enable</mode>
        <id>file-access</id>
        <tab-title l10n="yes"><![CDATA[Browse Networks]]></tab-title>
        <url-list-title l10n="yes"><![CDATA[File Folder Bookmarks]]></url-list-title>

```

```

        <order>3</order>
    </application>
    <application>
        <mode>enable</mode>
        <id>app-access</id>
        <tab-title l10n="yes"><![CDATA[Application Access]]></tab-title>
        <order>4</order>
    </application>
    <application>
        <mode>enable</mode>
        <id>net-access</id>
        <tab-title l10n="yes">AnyConnect</tab-title>
        <order>4</order>
    </application>
    <application>
        <mode>enable</mode>
        <id>help</id>
        <tab-title l10n="yes">Help</tab-title>
        <order>1000000</order>
    </application>
    <toolbar>
        <mode>enable</mode>
        <logout-prompt-text l10n="yes">Logout</logout-prompt-text>
        <prompt-box-title l10n="yes">Address</prompt-box-title>
        <browse-button-text l10n="yes">Browse</browse-button-text>
    </toolbar>
    <column>
        <width>100%</width>
        <order>1</order>
    </column>
    <pane>
        <type>TEXT</type>
        <mode>disable</mode>
        <title></title>
        <text></text>
        <notitle></notitle>
        <column></column>
        <row></row>
        <height></height>
    </pane>
    <pane>
        <type>IMAGE</type>
        <mode>disable</mode>
        <title></title>
        <url l10n="yes"></url>
        <notitle></notitle>
        <column></column>
        <row></row>
        <height></height>
    </pane>
    <pane>
        <type>HTML</type>
        <mode>disable</mode>
        <title></title>
        <url l10n="yes"></url>
        <notitle></notitle>
        <column></column>
        <row></row>
        <height></height>
    </pane>
    <pane>
        <type>RSS</type>
        <mode>disable</mode>
        <title></title>

```

```

        <url l10n="yes"></url>
        <notitle></notitle>
        <column></column>
        <row></row>
        <height></height>
    </pane>
    <url-lists>
        <mode>group</mode>
    </url-lists>
    <home-page>
        <mode>standard</mode>
        <url></url>
    </home-page>
</portal>
</custom>

```

Figure 39-9 shows the Logon page and its customizing XML tags. All these tags are nested within the higher-level tag `<auth-page>`.

Figure 39-9 Logon Page and Associated XML Tags

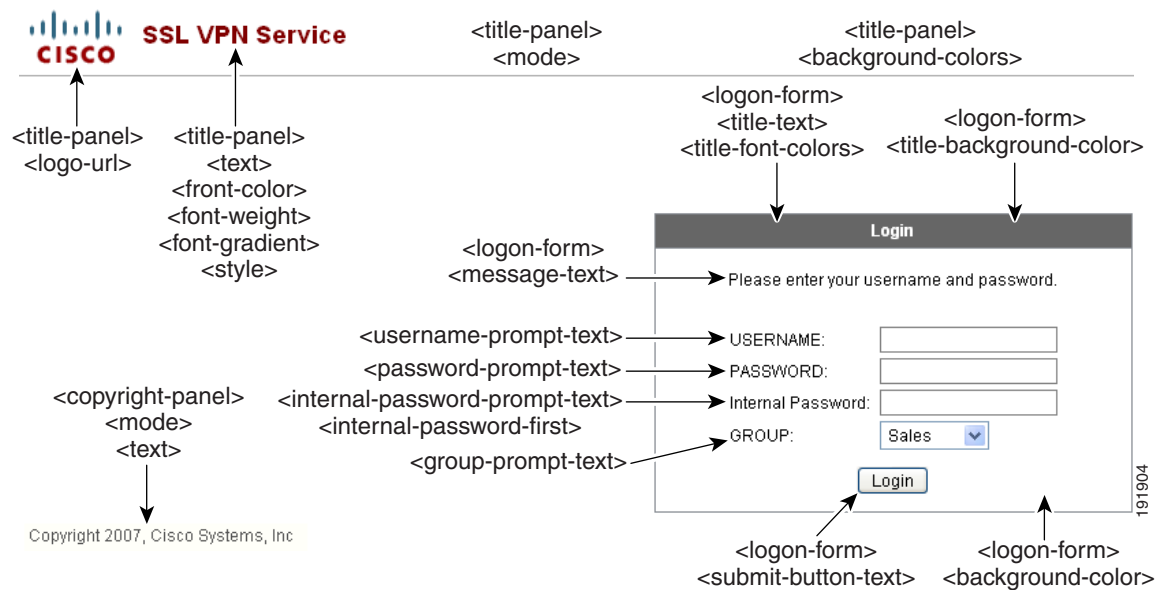


Figure 39-10 shows the Language Selector drop-down list that is available on the Logon page, and the XML tags for customizing this feature. All these tags are nested within the higher-level `<auth-page>` tag.

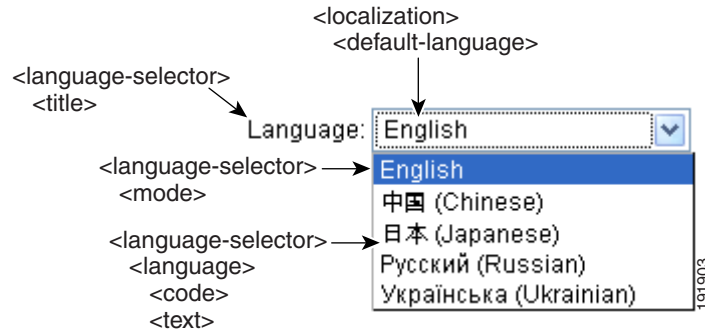
Figure 39-10 Language Selector on Logon Screen and Associated XML Tags

Figure 39-11 shows the Information Panel that is available on the Logon page, and the XML tags for customizing this feature. This information can appear to the left or right of the login box. These tags are nested within the higher-level `<auth-page>` tag.

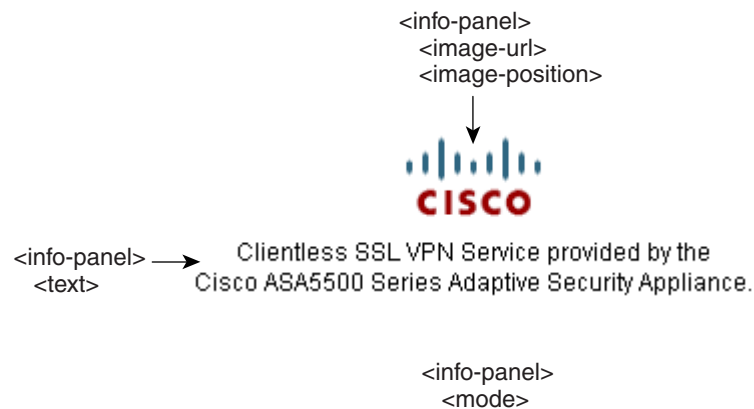
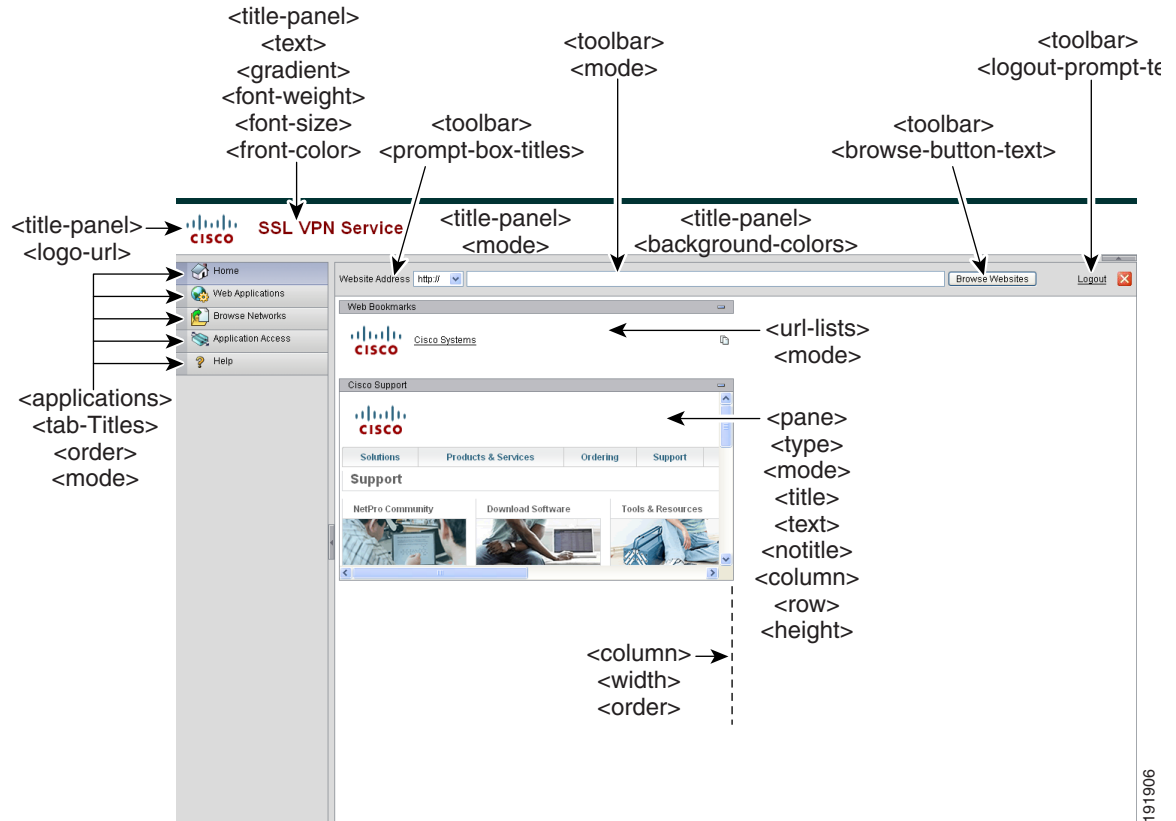
Figure 39-11 Information Panel on Logon Screen and Associated XML Tags

Figure 39-12 shows the Portal page and the XML tags for customizing this feature. These tags are nested within the higher-level `<auth-page>` tag.

Figure 39-12 Portal Page and Associated XML Tags

191906

Importing a Customization Object

After you edit and save the XML file, import it into cache memory of the security appliance using the **import webvpn customization** command from EXEC mode. When you import the customization object, the security appliance checks the XML code for validity. If the code is valid, the security appliance stores the object in a hidden location in cache memory.

The following example imports the customization object *General.xml* from the URL 209.165.201.22/customization and names it *custom1*.

```
hostname# import webvpn customization custom1 tftp://209.165.201.22/customization
/General.xml
Accessing
tftp://209.165.201.22/customization/General.xml...!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Writing file disk0:/cisco_config/97/custom1...
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
329994 bytes copied in 5.350 secs (65998 bytes/sec)
```

Applying Customizations to Connection Profiles, Group Policies and Users

After you create a customization, you can apply the customization to a connection profile, a group, or a user, with the **customization** command. The options displayed with this command are different depending on the mode you are in.



Note

Connection profiles were previously referred to as tunnel groups.

For more information about configuring connection profiles, group policies, and users, see [Chapter 32, “Configuring Connection Profiles, Group Policies, and Users.”](#)

Applying Customizations to Connection Profiles

To apply a customization to a connection profile, use the **customization** command from tunnel-group webvpn mode:

[no] customization name

name is the name of a customization to apply to the connection profile.

To remove the command from the configuration, and remove a customization from the connection profile, use the **no** form of the command.

Enter the **customization command followed by a question mark (?)** to view a list of existing customizations.

In the following example, the user enters tunnel-group webvpn mode and enables the customization *cisco* for the connection profile *cisco_telecommuters*:

```
hostname(config)# tunnel-group cisco_telecommuters webvpn-attributes
hostname(tunnel-group-webvpn)# customization cisco
```

Applying Customizations to Groups and Users

To apply a customization to a group or user, use the **customization** command from group policy webvpn mode or username webvpn mode. In these modes, the **none** and **value** options are included:

[no] customization {none | value name}

none disables the customization for the group or user, prevents the value from being inherited, and displays the default clientless SSL VPN pages.

value name is the name of a customization to apply to the group or user.

To remove the command from the configuration, and cause the value to be inherited, use the **no** form of the command.

Enter the **customization value command followed by a question mark (?)** to view a list of existing customizations.

In the following example, the user enters group policy webvpn mode, queries the security appliance for a list of customizations, and enables the customization *cisco* for the group policy *cisco_sales*:

```
hostname(config)# group-policy cisco_sales attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# customization value ?

config-username-webvpn mode commands/options:
Available configured customization profiles:
    DfltCustomization
    cisco
hostname(config-group-webvpn)# customization value cisco
```

In the next example, the user enters username webvpn mode and enables the customization *cisco* for the user *cisco_employee*:

```
hostname(config)# username cisco_employee attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# customization value cisco
```

Login Screen Advanced Customization

If you prefer to use your own, custom login screen, rather than changing specific screen elements of the login screen we provide, you can perform this advanced customization using the *Full Customization* feature.

With Full Customization, you provide the HTML for your own login screen, and you insert Cisco HTML code that calls functions on the security appliance that create the Login form and the Language Selector drop-down list.

This section describes the modifications you need to make to your HTML code and the tasks required to configure the security appliance to use your code.

Figure 39-13 shows the standard Cisco login screen that displays to clientless SSL VPN users. The Login form is displayed by a function called by the HTML code.

Figure 39-13 Standard Cisco Login Page

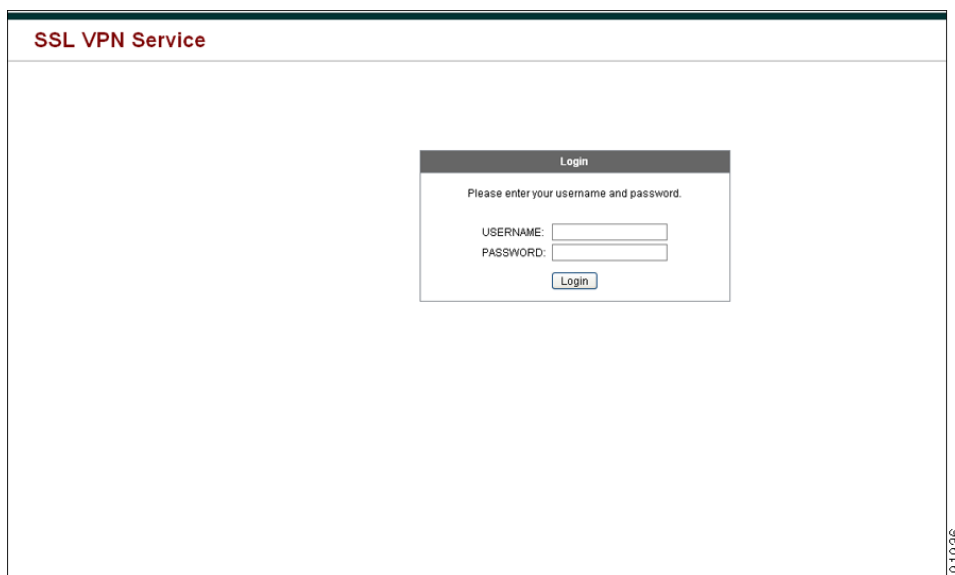
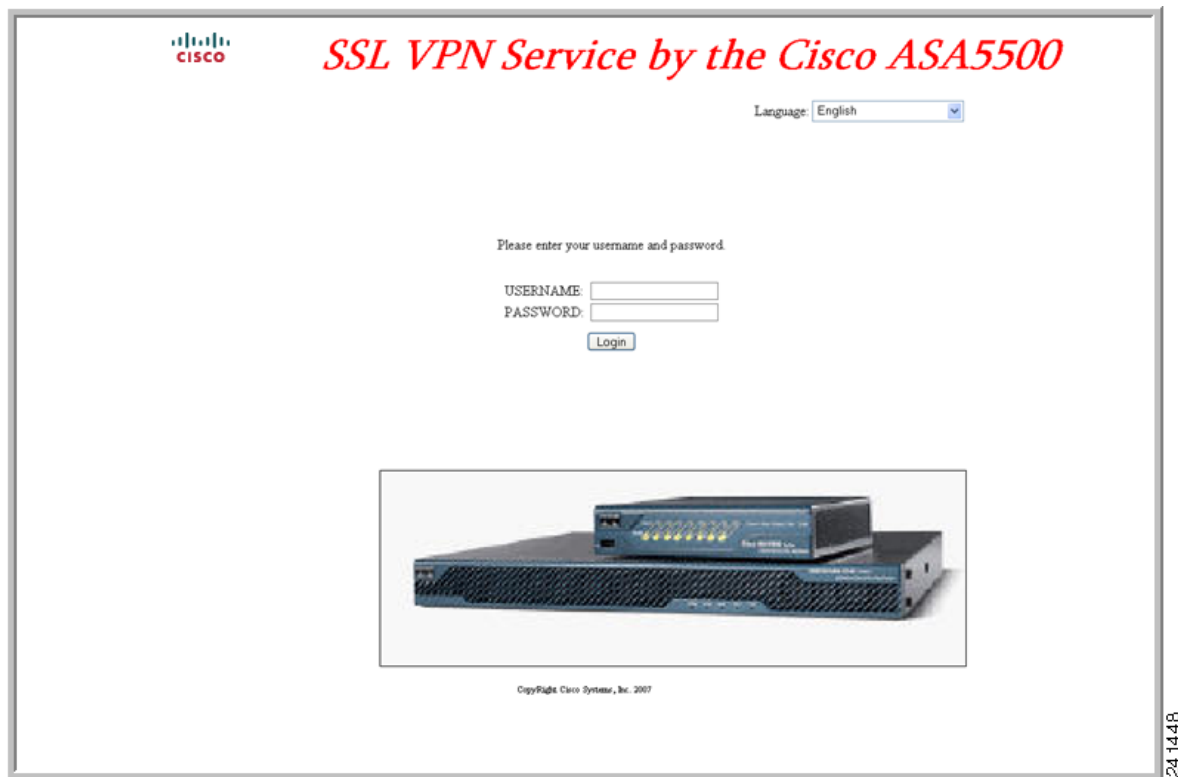


Figure 39-14 shows the Language Selector drop-down list. This feature is an option for clientless SSL VPN users, and is also called by a function in the HTML code of the login screen.

Figure 39-14 Language Selector Drop-down List

Figure 39-15 shows a simple example of a custom login screen enabled by the Full Customization feature.

Figure 39-15 Example of Full Customization of Login Screen

Example HTML Code for Custom Login Screen File

The following HTML code is used as an example and is the code that displays the screen shown in [Figure 39-15](#):

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>New Page 3</title>
<base target="_self">
</head>

<p align="center">
<font face="Snap
ITC" size="6" color="#FF00FF">
</font><font face="Snap ITC" color="#FF00FF" size="7">&nbsp;</font><i><b><font
color="#FF0000" size="7" face="Sylfaen"> SSL VPN Service by the Cisco
ASA5500</font></b></i></p>

<body onload="cscs_ShowLoginForm('lform');cscs_ShowLanguageSelector('selector')">

<table>

<tr><td colspan=3 height=20 align=right><div id="selector" style="width:
300px"></div></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr>
<td height="379"></td>
<td height="379"></td>
<td align=middle valign=middle>
<div id=lform >
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>Loading...</p>
</div>
</td>
</tr>
<tr>
<td width="251"></td>
<td width="1"></td>
<td align=right valign=right width="800">

</td></tr>

</table>
```

The indented code injects the Login form and the Language Selector on the screen. The function **cscs_ShowLoginForm('lform')** injects the logon form. **cscs_ShowLanguageSelector('selector')** injects the Language Selector.

Full Customization Procedure

Follow these steps to modify your HTML file and configure the security appliance to use the new file:

-
- Step 1** Name your file **logon.inc**. When you import the file, the security appliance recognizes this filename as the logon screen.
 - Step 2** Modify the paths of images used by the file to include **/+CSCOU+/.**

Files that are displayed to remote users before authentication must reside in a specific area of the security appliance cache memory represented by the path `/+CSCOU+/. Therefore, the source for each image in the file must include this path. For example:`

```
src="/+CSCOU+/asa5520.gif"
```

- Step 3** Insert the special HTML code below. This code contains the Cisco functions, described earlier, that inject the login form and language selector onto the screen.

```
<body onload="cisco_ShowLoginForm('lform');cisco_ShowLanguageSelector('selector')">

<table>

<tr><td colspan=3 height=20 align=right><div id="selector" style="width:
300px"></div></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr>
<td height="379"></td>
<td height="379"></td>
<td align=middle valign=middle>
<div id=lform >
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>Loading...</p>
</div>
</td>
</tr>
<tr>
<td width="251"></td>
<td width="1"></td>
<td align=right valign=right width="800">

</td></tr>

</table>
```

- Step 4** Import the file and images as Web Content using the **import webvpn webcontent** command from Privileged EXEC mode. For example:

```
hostname# import webvpn webcontent /+CSCOU+/login.inc tftp://209.165.200.225/login.inc
!!!!* Web resource `+CSCOU+/login.inc' was successfully initialized
hostname#
```

- Step 5** Enable Full Customization in a customization object. First, export a customization template with the **export webvpn customization template** command. For example:

```
hostname2# export webvpn customization template tftp://209.165.200.225/sales_vpn_login
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%INFO: Customization object 'Template' was exported to tftp://10.21.50.120/sales
_vpn_login
```

Then change the full customization mode tag in the file to enable, and supply the URL of the login file stored in the security appliance memory. For example:

```
<full-customization>
  <mode>enable</mode>
  <url>+CSCOU+/login.inc</url>
</full-customization>
```

Now import the file as a new customization object. For example:

```
hostname# import webvpn customization sales_vpn_login tftp://10.21.50.120/sales_vpn_login$
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
%INFO: customization object 'sales_vpn_login' was successfully imported
```

Step 6 Apply the customization object to a Connection Profile (tunnel group). For example:

```
hostname(config)# tunnel-group Sales webvpn-attributes
hostname(config-tunnel-webvpn)#customization sales_vpn_login
```

Customizing Help

The security appliance displays help content on the application panels during clientless SSL VPN sessions. You can customize the help files provided by Cisco or create help files in other languages. You then import them to flash memory for display during subsequent clientless sessions. You can also retrieve previously imported help content files, modify them, and reimport them to flash memory.

Each clientless application panel displays its own help file content using a predetermined filename. The prospective location of each is in the `/+CSCOE+/help/language/` URL within flash memory of the security appliance. [Table 39-11](#) shows the details about each of the help files you can maintain for clientless SSL VPN sessions.

Table 39-11 Clientless SSL VPN Application Help Files

Application Type	Panel	URL of Help File in Flash Memory of the Security Appliance	Help File Provided By Cisco in English?
Standard	Application Access	<code>/+CSCOE+/help/language/app-access-hlp.inc</code>	Yes
Standard	Browse Networks	<code>/+CSCOE+/help/language/file-access-hlp.inc</code>	Yes
Standard	AnyConnect Client	<code>/+CSCOE+/help/language/net-access-hlp.inc</code>	Yes
Standard	Web Access	<code>/+CSCOE+/help/language/web-access-hlp.inc</code>	Yes
Plug-in	MetaFrame Access	<code>/+CSCOE+/help/language/ica-hlp.inc</code>	No
Plug-in	Terminal Servers	<code>/+CSCOE+/help/language/rdp-hlp.inc</code>	Yes
Plug-in	Telnet/SSH Servers	<code>/+CSCOE+/help/language/ssh,telnet-hlp.inc</code>	Yes
Plug-in	VNC Connections	<code>/+CSCOE+/help/language/vnc-hlp.inc</code>	Yes

language is the abbreviation of the language rendered by the browser. This field is *not* used for file translation; it indicates the language used in the file. To specify a particular language code, copy the language abbreviation from the list of languages rendered by your browser. For example, a dialog window displays the languages and associated language codes when you use one of the following procedures:

- Open Internet Explorer and choose **Tools > Internet Options > Languages > Add**.
- Open Mozilla Firefox and choose **Tools > Options > Advanced > General**, click **Choose** next to Languages, and click **Select a language to add**.

The following sections describe how to customize the help content visible on clientless sessions:

- [Customizing a Help File Provided By Cisco, page 39-77](#)
- [Creating Help Files for Languages Not Provided by Cisco, page 39-77](#)
- [Importing a Help File to Flash Memory, page 39-78](#)

- [Exporting a Previously Imported Help File from Flash Memory, page 39-78](#)

Customizing a Help File Provided By Cisco

To customize a help file provided by Cisco, you need to get a copy of the file from the flash memory card first. Get the copy and customize it as follows:

- Step 1** Use your browser to establish a clientless SSL VPN session with the security appliance.
- Step 2** Display the help file by appending the string in “URL of Help File in Flash Memory of the Security Appliance” in [Table 39-11](#), to the address of the security appliance, then press Enter.



Note Enter **en** in place of *language* to get the help file in English.

The following example address displays the English version of the Terminal Servers help:

`https://address_of_security_appliance/+CSCOE+/help/en/rdp-hlp.inc`

- Step 3** Choose File > Save (Page) As.



Caution Do not change the contents of the File name box.

- Step 4** Change the Save as type option to “Web Page, HTML only” and click Save.

- Step 5** Use your preferred HTML editor to modify the file.



Note You can use most HTML tags, but do *not* use tags that define the document and its structure (e.g., do not use <html>, <title>, <body>, <head>, <h1>, <h2>, etc. You can use character tags, such as the tag, and the <p>, , , and tags to structure content.

- Step 6** Save the file as HTML only, using the original filename and extension.

- Step 7** Make sure the filename matches the one in [Table 39-11](#), and that it does not have an extra filename extension.

See “[Importing a Help File to Flash Memory](#)” to import the modified file for display in clientless SSL VPN sessions.

Creating Help Files for Languages Not Provided by Cisco

Use HTML to create help files in other languages.



Note You can use most HTML tags, but do *not* use tags that define the document and its structure (e.g., do not use <html>, <title>, <body>, <head>, <h1>, <h2>, etc. You can use character tags, such as the tag, and the <p>, , , and tags to structure content.

We recommend creating a separate folder for each language you want to support.

Save the file as HTML only. Use the filename following the last slash in “URL of Help File in Flash Memory of the Security Appliance” in [Table 39-11](#).

See the next section to import the files for display in clientless SSL VPN sessions.

Importing a Help File to Flash Memory

To import a help content file to flash memory for display in clientless SSL VPN sessions, enter the following command in Privileged EXEC mode:

```
import webvpn webcontent destination_url source_url
```

destination_url is the string in “URL of Help File in Flash Memory of the Security Appliance” in [Table 39-11](#).

source_url is the URL of the file to import. Valid prefixes are ftp://, http://, and tftp://.

The following example command copies the help file *app-access-hlp.inc* to flash memory from the TFTP server at 209.165.200.225. The URL includes the abbreviation *en* for the English language.

```
hostname# import webvpn webcontent /+CSCOE+/help/en/app-access-hlp.inc  
tftp://209.165.200.225/app-access-hlp.inc
```

Exporting a Previously Imported Help File from Flash Memory

To retrieve a previously imported help content file for subsequent edits, enter the following command in Privileged EXEC mode:

```
export webvpn webcontent source_url destination_url
```

source_url is the string in “URL of Help File in Flash Memory of the Security Appliance” in [Table 39-11](#).

destination_url is **the target URL**. Valid prefixes are ftp:// and tftp://. The maximum number of characters is 255.

The following example command copies the English language help file *file-access-hlp.inc* displayed on the Browse Networks panel to TFTP Server 209.165.200.225:

```
hostname# export webvpn webcontent /+CSCOE+/help/en/file-access-hlp.inc  
tftp://209.165.200.225/file-access-hlp.inc
```

Requiring Usernames and Passwords

Depending on your network, during a remote session users might have to log in to any or all of the following: the computer itself, an Internet service provider, clientless SSL VPN, mail or file servers, or corporate applications. Users might have to authenticate in many different contexts, requiring different information, such as a unique username, password, or PIN.

[Table 39-12](#) lists the type of usernames and passwords that clientless SSL VPN users might need to know.

Table 39-12 *Username and Passwords to Give to Users of Clientless SSL VPN Sessions*

Login Username/ Password Type	Purpose	Entered When
Computer	Access the computer	Starting the computer
Internet Service Provider	Access the Internet	Connecting to an Internet service provider
Clientless SSL VPN	Access remote network	Starting clientless SSL VPN
File Server	Access remote file server	Using the clientless SSL VPN file browsing feature to access a remote file server
Corporate Application Login	Access firewall-protected internal server	Using the clientless SSL VPN web browsing feature to access an internal protected website
Mail Server	Access remote mail server via clientless SSL VPN	Sending or receiving e-mail messages

Communicating Security Tips

Advise users to always click the logout icon on the toolbar to close the clientless SSL VPN session. (Closing the browser window does not close the session.)

Advise users that using clientless SSL VPN does not ensure that communication with every site is secure. Clientless SSL VPN ensures the security of data transmission between the remote PC or workstation and the security appliance on the corporate network. If a user then accesses a non-HTTPS web resource (located on the Internet or on the internal network), the communication from the corporate security appliance to the destination web server is not private because it is not encrypted.

“Observing Clientless SSL VPN Security Precautions” on page 2 addresses an additional tip to communicate with users, depending on the steps you follow within that section.

Configuring Remote Systems to Use Clientless SSL VPN Features

Table 39-13 includes the following information about setting up remote systems to use clientless SSL VPN:

- Starting clientless SSL VPN
- Using the clientless SSL VPN Floating Toolbar
- Web Browsing
- Network Browsing and File Management
- Using Applications (Port Forwarding)
- Using E-mail via Port Forwarding
- Using E-mail via Web Access
- Using E-mail via e-mail proxy

Table 39-13 also provides information about the following:

- Clientless SSL VPN requirements, by feature

- Applications supported by clientless SSL VPN
- Client application installation and configuration requirements
- Information you might need to provide end users
- Tips and use suggestions for end users

It is possible you have configured user accounts differently and that different clientless SSL VPN features are available to each user. [Table 39-13](#) organizes information by feature, so you can skip over the information for unavailable features.

Table 39-13 Remote System Configuration and End User Requirements for Clientless SSL VPN

Task	Remote System or End User Requirements	Specifications or Use Suggestions
Starting clientless SSL VPN	Connection to the Internet	Any Internet connection is supported, including: <ul style="list-style-type: none"> • Home DSL, cable, or dial-ups • Public kiosks • Hotel hook-ups • Airport wireless nodes • Internet cafes
	Web browsers supported by clientless SSL VPN	See the Cisco ASA 5500 Series VPN Compatibility Reference
	Cookies enabled on browser	Cookies must be enabled on the browser in order to access applications via port forwarding.
	URL for clientless SSL VPN	An https address in the following form: https://address where <i>address</i> is the IP address or DNS hostname of an interface of the security appliance (or load balancing cluster) on which clientless SSL VPN is enabled. For example: https://10.89.192.163 or https://cisco.example.com.
	Clientless SSL VPN username and password	—
	[Optional] Local printer	Clientless SSL VPN does not support printing from a web browser to a network printer. Printing to a local printer is supported.

Table 39-13 Remote System Configuration and End User Requirements for Clientless SSL VPN (continued)


Task	Remote System or End User Requirements	Specifications or Use Suggestions
Using the floating toolbar displayed during a clientless SSL VPN session		<p>A floating toolbar is available to simplify the use of clientless SSL VPN. The toolbar lets you enter URLs, browse file locations, and choose preconfigured web connections without interfering with the main browser window.</p> <p>If you configure your browser to block popups, the floating toolbar cannot display.</p> <p>The floating toolbar represents the current clientless SSL VPN session. If you click the Close button, the security appliance prompts you to confirm that you want to close the clientless SSL VPN session.</p> <div>  <p>Tip TIP: To paste text into a text field, use Ctrl-V. (Right-clicking is disabled on the toolbar displayed during the clientless SSL VPN session.)</p> </div>
Web browsing	<p>Username and passwords for protected websites</p>	<p>Using clientless SSL VPN does not ensure that communication with every site is secure. See “Communicating Security Tips.”</p> <p>The look and feel of web browsing with clientless SSL VPN might be different from what users are accustomed to. For example:</p> <ul style="list-style-type: none"> • The title bar for clientless SSL VPN appears above each web page. • You access websites by: <ul style="list-style-type: none"> – Entering the URL in the Enter Web Address field on the clientless SSL VPN Home page – Clicking on a preconfigured website link on the clientless SSL VPN Home page – Clicking a link on a webpage accessed via one of the previous two methods <p>Also, depending on how you configured a particular account, it might be that:</p> <ul style="list-style-type: none"> • Some websites are blocked • Only the websites that appear as links on the clientless SSL VPN Home page are available

Table 39-13 Remote System Configuration and End User Requirements for Clientless SSL VPN (continued)

Task	Remote System or End User Requirements	Specifications or Use Suggestions
Network browsing and file management	File permissions configured for shared remote access	Only shared folders and files are accessible via clientless SSL VPN.
	Server name and passwords for protected file servers	—
	Domain, workgroup, and server names where folders and files reside	Users might not be familiar with how to locate their files through your organization network.
	—	Do not interrupt the Copy File to Server command or navigate to a different screen while the copying is in progress. Interrupting the operation can cause an incomplete file to be saved on the server.

Table 39-13 Remote System Configuration and End User Requirements for Clientless SSL VPN (continued)


Task	Remote System or End User Requirements	Specifications or Use Suggestions
Using Application Access	Note On Macintosh OS X, only the Safari browser supports this feature.	
	Note Because this feature requires installing Sun Microsystems Java™ Runtime Environment and configuring the local clients, and because doing so requires administrator permissions on the local system or full control of C:\windows\System32\drivers\etc, it is unlikely that users will be able to use applications when they connect from public remote systems.	
	 Caution Users should always close the Application Access window when they finish using applications by clicking the Close icon. Failure to quit the window properly can cause Application Access or the applications themselves to be disabled. See Recovering from hosts File Errors When Using Application Access for details.	
	Client applications installed	—
	Cookies enabled on browser	—
	Administrator privileges	User must have administrator access on the PC if you use DNS names to specify servers because modifying the hosts file requires it.
	Sun Microsystems Java Runtime Environment (JRE) version 1.4.x and 1.5.x installed. Javascript must be enabled on the browser. By default, it is enabled.	If JRE is not installed, a pop-up window displays, directing users to a site where it is available. On rare occasions, the port forwarding applet fails with JAVA exception errors. If this happens, do the following: <ol style="list-style-type: none"> 1. Clear the browser cache and close the browser. 2. Verify that no JAVA icons are in the computer task bar. Close all instances of JAVA. 3. Establish a clientless SSL VPN session and launch the port forwarding JAVA applet.
	Client applications configured, if necessary. Note The Microsoft Outlook client does not require this configuration step. All non-Windows client applications require configuration. To see if configuration is necessary for a Windows application, check the value of the Remote Server. <ul style="list-style-type: none"> • If the Remote Server contains the server hostname, you do not need to configure the client application. • If the Remote Server field contains an IP address, you must configure the client application. 	To configure the client application, use the server's locally mapped IP address and port number. To find this information: <ol style="list-style-type: none"> 1. Start a clientless SSL VPN session and click the Application Access link on the Home page. The Application Access window appears. 2. In the Name column, find the name of the server you want to use, then identify its corresponding client IP address and port number (in the Local column). 3. Use this IP address and port number to configure the client application. Configuration steps vary for each client application.
	Note Clicking a URL (such as one in an -e-mail message) in an application running over a clientless SSL VPN session does not open the site over that session. To open a site over the session, paste the URL into the Enter Clientless SSL VPN (URL) Address field.	

Table 39-13 Remote System Configuration and End User Requirements for Clientless SSL VPN (continued)

Task	Remote System or End User Requirements	Specifications or Use Suggestions
Using e-mail via Application Access	Fulfill requirements for Application Access (See Using Applications)	To use mail, start Application Access from the clientless SSL VPN Home page. The mail client is then available for use.
	<p>Note If you are using an IMAP client and you lose your mail server connection or are unable to make a new connection, close the IMAP application and restart clientless SSL VPN.</p> <p>Other mail clients</p>	<p>We have tested Microsoft Outlook Express versions 5.5 and 6.0.</p> <p>Clientless SSL VPN should support other SMTPS, POP3S, or IMAP4S e-mail programs via port forwarding, such as Lotus Notes and Eudora, but we have not verified them.</p>
Using e-mail via Web Access	Web-based e-mail product installed	<p>Supported:</p> <ul style="list-style-type: none"> Microsoft Outlook Web Access to Exchange Server 2000, 2003, and 2007. <p>For best results, use OWA on Internet Explorer 6.x or higher, or Firefox 2.0 or higher.</p> <ul style="list-style-type: none"> Lotus iNotes <p>Other web-based e-mail products should also work, but we have not verified them.</p>
Using e-mail via E-mail Proxy (legacy feature)	<p>SSL-enabled mail application installed</p> <p>Do not set the security appliance SSL version to TLSv1 Only. Outlook and Outlook Express do not support TLS.</p>	<p>Supported mail applications:</p> <ul style="list-style-type: none"> Microsoft Outlook 2000 and 2002 Microsoft Outlook Express 5.5 and 6.0 Eudora 4.2 for Windows 2000 <p>Other SSL-enabled mail clients should also work, but we have not verified them.</p>
	Mail application configured	See instructions and examples for your mail application in “Using E-Mail over Clientless SSL VPN.”

Translating the Language of User Messages

The security appliance provides language translation for the portal and screens displayed to users that initiate browser-based, clientless SSL VPN connections, as well as the interface displayed to Cisco AnyConnect VPN Client users.

This section describes how to configure the security appliance to translate these user messages and includes the following sections:

- [Understanding Language Translation, page 39-85](#)
- [Creating Translation Tables, page 39-86](#)
- [Referencing the Language in a Customization Object, page 39-87](#)
- [Changing a Group Policy or User Attributes to Use the Customization Object, page 39-89](#)

Understanding Language Translation

Functional areas and their messages that are visible to remote users are organized into translation domains. [Table 39-14](#) shows the translation domains and the functional areas translated.

Table 39-14 Translation Domains and Functional Areas Affected

Translation Domain	Functional Areas Translated
AnyConnect	<i>Messages displayed on the user interface of the Cisco AnyConnect VPN Client.</i>
CSD	Messages for Cisco Secure Desktop.
customization	<i>Messages on the logon and logout pages, portal page, and all the messages customizable by the user.</i>
banners	Banners displayed to remote users and messages when VPN access is denied.
PortForwarder	Messages displayed to Port Forwarding users.
url-list	Text that user specifies for URL bookmarks on the portal page.
webvpn	All the layer 7, AAA and portal messages that are not customizable.
plugin-ica	Messages for the Citrix plug-in.
plugin-rdp	Messages for the Remote Desktop Protocol plug-in.
plugin-telnet,ssh	Messages for the Telnet and SSH plug-in.
plugin-vnc	Messages for the VNC plug-in.

The software image package for the security appliance includes a translation table template for each domain that is part of the standard functionality. The templates for plug-ins are included with the plug-ins and define their own translation domains.

You can export the template for a translation domain, which creates an XML file of the template at the URL you provide. The message fields in this file are empty. You can edit the messages and import the template to create a new translation table object that resides in flash memory.

You can also export an existing translation table. The XML file created displays the messages you edited previously. Reimporting this XML file with the same language name creates a new version of the translation table object, overwriting previous messages.

Some templates are static, but some change based on the configuration of the security appliance. Because you can customize the *logon and logout pages, portal page, and URL bookmarks for clientless users*, the **security appliance generates the customization** and **url-list** translation domain templates dynamically and the template automatically reflects your changes to these functional areas.

After creating translation tables, they are available to customization objects that you create and apply to group policies or user attributes. With the exception of the AnyConnect translation domain, a translation table has no affect and messages are not translated on user screens until you create a customization object, identify a translation table to use in that object, and specify that customization for the group policy or user. Changes to the translation table for the AnyConnect domain are immediately visible to AnyConnect client users.

Creating Translation Tables

The following procedure describes how to create translation tables:

- Step 1** Export a translation table template to a computer with the **export webvpn translation-table** command from privileged EXEC mode.

In the following example, the **show webvpn translation-table** command shows available translation table templates and tables.

```
hostname# show import webvpn translation-table
Translation Tables' Templates:
customization
AnyConnect
CSD
PortForwarder
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin
```

Translation Tables:

The next example exports the translation table template for the customization domain, which affects messages displayed for users in clientless SSL VPN sessions. The filename of the XML file created is *portal* (user-specified) and contains empty message fields:

```
hostname# export webvpn translation-table customization template
tftp://209.165.200.225/portal
```

- Step 2** Edit the translation table XML file.

The following example shows a portion of the template that was exported as *portal*. The end of this output includes a message ID field (msgid) and a message string field (msgstr) for the message *SSL VPN*, which is displayed on the portal page when a user establishes a clientless SSL VPN session. The complete template contains many pairs of message fields:

```
# Copyright (C) 2006 by Cisco Systems, Inc.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: ASA\n"
"Report-Msgid-Bugs-To: vkamyshe@cisco.com\n"
"POT-Creation-Date: 2007-03-12 18:57 GMT\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"

#: DfltCustomization:24 DfltCustomization:64
msgid "Clientless SSL VPN Service"
msgstr ""
```

The message ID field (msgid) contains the default translation. The message string field (msgstr) that follows msgid provides the translation. To create a translation, enter the translated text between the quotes of the msgstr string.

- Step 3** Import the translation table using the **import webvpn translation-table** command from privileged EXEC mode.

In the following example, the XML file is imported *es-us*—the abbreviation for Spanish spoken in the United States.

```
hostname# import webvpn translation-table customization language es-us
tftp://209.165.200.225/portal
hostname# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
hostname# show import webvpn translation-table
Translation Tables' Templates:
AnyConnect
PortForwarder
csd
customization
keepout
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
es-us customization
```

If you import a translation table for the AnyConnect domain, your changes are effective immediately. If you import a translation table for any other domain, you must continue to [Step 4](#), where you create a customization object, identify the translation table to use in that object, and specify that customization object for the group policy or user.

Referencing the Language in a Customization Object

Now that you have created a translation table, you need to refer to this table in a customization object.

Steps 4 through 6 describe how to export the customization template, edit it, and import it as a customization object:

- Step 4** Export a customization template to a URL where you can edit it using the **export webvpn customization template** command from privileged EXEC mode. The example below exports the template and creates the copy *sales* at the URL specified:

```
hostname# export webvpn customization template tftp://209.165.200.225/sales
```

- Step 5** Edit the customization template and reference the previously-imported translation table.

There are two areas of XML code in the customization template that pertain to translation tables. The first area, shown below, specifies the translation tables to use:

```
<localization>
  <languages>en, ja, zh, ru, ua</languages>
  <default-language>en</default-language>
</localization>
```

The `<languages>` tag in the XML code is followed by the names of the translation tables. In this example, they are *en*, *ja*, *zh*, *ru*, and *ua*. For the customization object to call these translation tables correctly, the tables must have been previously imported using the same names. These names must be compatible with language options of the browser.

The `<default-language>` tag specifies the language that the remote user first encounters when connecting to the security appliance. In the example code above, the language is English.

Figure 39-16 shows the Language Selector that displays on the logon page. The Language Selector gives the remote user establishing an SSL VPN connection the ability to choose a language.

Figure 39-16 Language Selector



The XML code below affects the display of the Language Selector, and includes the `<language-selector>` tag and the associated `<language>` tags that enable and customize the Language Selector:

```
<auth-page>
  ....
  <language-selector>
    <mode>enable</mode>
    <title l10n="yes">Language:</title>
    <language>
      <code>en</code>
      <text>English</text>
    </language>
    <language>
      <code>es-us</code>
      <text>Spanish</text>
    </language>
  </language-selector>
```

The `<language-selector>` group of tags includes the `<mode>` tag that enables and disables the displaying of the Language Selector, and the `<title>` tag that specifies the title of the drop-down box listing the languages.

The `<language>` group of tags includes the `<code>` and `<text>` tags that map the language name displayed in the Language Selector drop-down box to a specific translation table.

Make your changes to this file and save the file.

Step 6 Import the customization template as a new object using the **import webvpn customization** command from privileged EXEC mode. For example:

```
hostname# import webvpn customization sales tftp://209.165.200.225/sales
hostname# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

The output of the **show import webvpn customization** command shows the new customization object *sales*:

```
hostname(config)# show import webvpn customization
Template
sales
hostname(config)#
```


Changing a Group Policy or User Attributes to Use the Customization Object

Now that you have created the customization object, you need to activate your changes for specific groups or users. Step 7 shows how to enable the customization object in a group policy:

- Step 7** Enter the group policy webvpn configuration mode for a group policy and enable the customization object using the **customization** command. The following example shows the customization object *sales* enabled in the group policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# customization value sales
```

Capturing Data

The CLI **capture** command lets you log information about websites that do not display properly over a clientless SSL VPN session. This data can help your Cisco customer support engineer troubleshoot problems. The following sections describe how to capture and view clientless SSL VPN session data:

- [Creating a Capture File](#)
- [Using a Browser to Display Capture Data](#)



Note

Enabling clientless SSL VPN capture affects the performance of the security appliance. Be sure to disable the capture after you generate the capture files needed for troubleshooting.

Creating a Capture File

Perform the following steps to capture data about a clientless SSL VPN session to a file.

- Step 1** To start the capture utility for clientless SSL VPN, use the **capture** command from privileged EXEC mode.

```
capture capture_name type webvpn user webvpn_username
```

where:

- *capture_name* is a name you assign to the capture, which is also prepended to the name of the capture files.
- *webvpn_user* is the username to match for capture.

The capture utility starts.

- Step 2** A user logs in to begin a clientless SSL VPN session. The capture utility is capturing packets. Stop the capture by using the **no** version of the command.

```
no capture capture_name
```



Note

You must use the **no capture** command before webvpn logout.

The capture utility creates a *capture_name.zip* file, which is encrypted with the password **koleso**.

- Step 3** Send the .zip file to Cisco Systems, or attach it to a Cisco TAC service request.
- Step 4** To look at the contents of the .zip file, unzip it using the password **koleso**.
-

The following example creates a capture named *hr*, which captures traffic for user2 to a file:

```
hostname# capture hr type webvpn user user2
WebVPN capture started.
  capture name    hr
  user name       user2
hostname# no capture hr
```

Using a Browser to Display Capture Data

Perform the following steps to capture data about a clientless SSL VPN session and view it in a browser.

- Step 1** To start the capture utility for clientless SSL VPN, use the **capture** command from privileged EXEC mode.

```
capture capture_name type webvpn user webvpn_username
```

where:

- *capture_name* is a name you assign to the capture, which is also prepended to the name of the capture files.
- *webvpn_user* is the username to match for capture.

The capture utility starts.

- Step 2** A user logs in to begin a clientless SSL VPN session. The capture utility is capturing packets. Stop the capture by using the **no** version of the command.

- Step 3** Open a browser and in the address box enter

```
https://asdm_enabled_interface_of_the_security_appliance:port/admin/capture/capture_name/pcap
```

The following example command displays the capture named *hr*:

```
https://192.0.2.1:60000/admin/capture/hr/pcap
```

The captured content displays in a sniffer format.

- Step 4** When you finish examining the capture content, stop the capture by using the **no** version of the command.
-