SSL VPN Configuration Guide, Cisco IOS Release 12.4

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

SSL VPN

The SSL VPN feature (also known as WebVPN) provides support, in Cisco IOS software, for remote user access to enterprise networks from anywhere on the Internet. Remote access is provided through a Secure Socket Layer (SSL)-enabled SSL VPN gateway. The SSL VPN gateway allows remote users to establish a secure VPN tunnel using a web browser. This feature provides a comprehensive solution that allows easy access to a broad range of web resources and web-enabled applications using native HTTP over SSL (HTTPS) browser support. SSL VPN delivers three modes of SSL VPN access: clientless, thin-client, and full-tunnel client support.

This document is primarily for system administrators. If you are a remote user, see the document *SSL VPN Remote User Guide*.

Note The Cisco AnyConnect VPN Client is introduced in Cisco IOS Release 12.4(15)T. This feature is the next-generation SSL VPN Client. If you are using Cisco software earlier than Cisco IOS Release 12.4(15)T, you should be using the SSL VPN Client and see the GUI for the SSL VPN Client when you are web browsing. However, if you are using Cisco Release 12.4(15)T or a later release, you should be using the Cisco AnyConnect VPN Client and see the GUI for Cisco AnyConnect VPN Client when you are web browsing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SSL VPN

To securely access resources on a private network behind an SSL VPN gateway, the remote user of an SSL VPN service must have the following:

- An account (login name and password)
- An SSL-enabled browser (for example, Internet Explorer, Netscape, Mozilla, or Firefox)
- Operating system support
- "Thin client" support used for TCP port-forwarding applications requires administrative privileges on the computer of the remote user.
- "Tunnel mode" for Cisco SSL VPN requires administrative privileges for initial installation of the full tunnel client.
- The remote user must have local administrative privileges to use thin client or full tunnel client features.
- The SSL VPN gateway and context configuration must be completed before a remote user can access resources on a private network behind an SSL VPN. For more information, see the How to Configure SSL VPN Services on a Router, page 35 section.
- ACL Support--The time range should have already been configured.
- Single SignOn Netegrity Cookie Support--A Cisco plug-in must be installed on a Netegrity SiteMinder server.
- Licensing--In Cisco IOS Release 15.0(1)M, the SSL VPN gateway is a seat-counted licensing feature on Cisco 880, Cisco 890, Cisco 1900, Cisco 2900, and Cisco 3900 platforms. A valid licence is required for a successful SSL VPN session.
- SSL VPN-supported browser--The following browsers have been verified for SSL VPN. Other browsers might not fully support SSL VPN features.

  Note Later versions of the following software are also supported.

  - Firefox 2.0 (Windows and Linux)
  - Internet Explorer 6.0 or 7.0
  - Linux (Redhat RHEL 3.0 +, FEDORA 5, or FEDORA 6)
  - Macintosh OS X 10.4.6
  - Microsoft Windows 2000, Windows XP, or Windows Vista
  - Safari 2.0.3

Restrictions for SSL VPN

General Restrictions for SSL VPN

- URLs referred by the Macromedia Flash player cannot be modified for secure retrieval by the SSL VPN gateway.
- Cisco Secure Desktop (CSD) 3.1 and later versions are not supported.

Cisco AnyConnect VPN Client

The Cisco AnyConnect VPN Client is not supported on Windows Mobile when the client connects to a Cisco IOS headend router (supported in Cisco IOS Release 15.0(1)M and later releases). The Cisco AnyConnect VPN Client does not support the following:

- Client-side authentication (supported in Cisco IOS Release 15.0(1)M and later releases)
- Compression support
- IPsec
- IPv6 VPN access
- Language translation (localization)
- Sequencing
- Standalone Mode (supported in Cisco IOS Release 12.4(20)T and later releases)

Thin Client Control List Support

Although there is no limitation on the maximum number of filtering rules that can be applied for each access control list (ACL) entry, keeping the number below 50 should have no impact on router performance.

HTTP Proxy

The HTTP Proxy feature works only with Microsoft Internet Explorer.

The HTTP Proxy feature will not work if the browser proxy setup cannot be modified because of any security policies that have been placed on the client workstation.

Features Not Supported on the Cisco IOS SSL VPN

The following features are not supported on the Cisco IOS SSL VPN:

- Application Profile Customization Framework (APCF): an XML-based rule set for clientless SSL VPN
- Java and ActiveX Client Server Plugins
- On Board Built-in Single Sign On
- Smart Tunnels
- SharePoint Support
- Portal Page Customization
- Using Smartcard for Authentication (supported in Cisco IOS Release 15.0(1)M and later releases)
- Support for External Statistics Reporting and Monitoring Tools
- Lightweight Directory Access Protocol (LDAP) Support
- Dynamic Access Policies (DAP)
- Cisco Unified Communications Manager (Cisco UCM) 8.0.1 VPN-enabled 7900 series IP phones

- The following features introduced in the AnyConnect 2.5.217 release:
  - AnyConnect Profile Editor
  - Captive Portal Hotspot Detection
  - Captive Portal Remediation
  - Client Firewall with Local Printer and Tethered Device Support
  - Connect Failure Policy
  - Optimal Gateway Selection
  - Post Log-in Always-on VPN
  - Quarantine
- Although you can connect to a Cisco IOS headend using AnyConnect 2.5, the features introduced in AnyConnect 2.5 will not be supported. However, features introduced in AnyConnect 2.4 and earlier releases are supported when you are using AnyConnect 2.5 with a Cisco IOS headend.

Note AnyConnect 3.0 is not supported when you are connecting to a Cisco IOS headend.

Information About SSL VPN

SSL VPN Overview

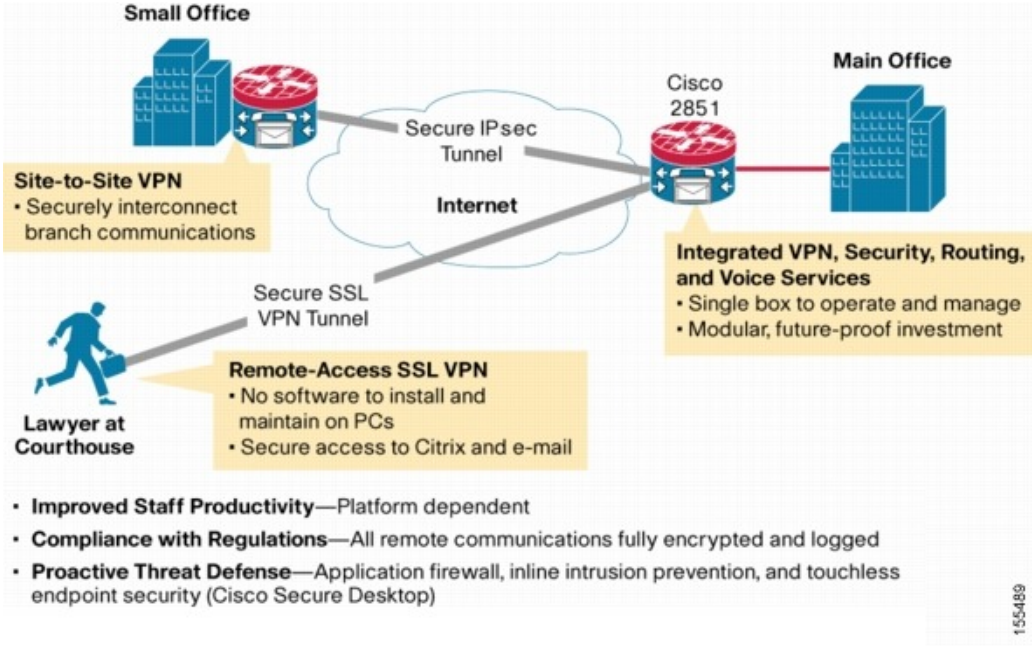
Cisco IOS SSL VPN provides SSL VPN remote-access connectivity from almost any Internet-enabled location using only a web browser that natively supports SSL encryption. This feature allows your company to extend access to its secure enterprise network to any authorized user by providing remote-access connectivity to corporate resources from any Internet-enabled location.

Cisco IOS SSL VPN can also support access from noncorporate-owned machines, including home computers, Internet kiosks, and wireless hot spots. These locations are difficult places to deploy and manage VPN client software and the remote configuration required to support IPsec VPN connections.

The figure below shows how a mobile worker (the lawyer at the courthouse) can access protected resources from the main office and branch offices. Site-to-site IPsec connectivity between the main and remote sites

is unaltered. The mobile worker needs only Internet access and supported software (web browser and operating system) to securely access the corporate network.

Figure 1 Secure SSL VPN Access Model



SSL VPN delivers the following three modes of SSL VPN access:

- Clientless--Clientless mode provides secure access to private web resources and will provide access to web content. This mode is useful for accessing most content that you would expect to access in a web browser, such as Internet access, databases, and online tools that employ a web interface.
- Thin client (port-forwarding Java applet)--Thin-client mode extends the capability of the cryptographic functions of the web browser to enable remote access to TCP-based applications such as Post Office Protocol version 3 (POP3), Simple Mail Transfer Protocol (SMTP), Internet Message Access protocol (IMAP), Telnet, and Secure Shell (SSH).
- Tunnel mode--Full tunnel client mode offers extensive application support through its dynamically downloaded Cisco AnyConnect VPN Client (next-generation SSL VPN Client) for SSL VPN. Full tunnel client mode delivers a lightweight, centrally configured and easy-to-support SSL VPN tunneling client that provides network layer access to virtually any application.

SSL VPN application accessibility is somewhat constrained relative to IPsec VPNs; however, SSL-based VPNs provide access to a growing set of common software applications, including web page access, web-enabled services such as file access, e-mail, and TCP-based applications (by way of a downloadable thin-client applet). SSL-based VPN requires slight changes to user workflow because some applications are presented through a web browser interface, not through their native GUI. The advantage for SSL VPN comes from accessibility from almost any Internet-connected system without needing to install additional desktop software.

Licensing

Starting in Cisco IOS Release 15.0(1)M, the SSL VPN gateway is a seat-counted licensing feature on the Cisco 880, Cisco 890, Cisco 1900, Cisco 2900, and Cisco 3900 platforms. A license count is associated

with each license, and the count indicates the instances of the feature available for use in the system. In the case of SSL VPN, a seat refers to the maximum number of sessions allowed at a time.

You can get the license at http://www.cisco.com/go/license.

For instructions on installing a license using Cisco License Manager (CLM), see the *User Guide for Cisco License Manager, Release 2.2* at http://www.cisco.com/en/US/docs/net_mgmt/license_manager/lm_2_2/2.2_user_guide/clm_book.html.

For instructions on installing a license using Cisco CLI, see the "Cisco IOS Software Activation Tasks and Commands" chapter of the *Software Activation Configuration Guide* at http://www.cisco.com/en/US/docs/ios/csa/configuration/guide/csa_commands_ps6441_TSD_Products_Configuration_Guide_Chapter.html.

SSL VPN supports the following types of licenses:

- Permanent licenses--No usage period is associated with these licenses. All permanent licenses are node locked and validated during installation and usage.
- Evaluation licenses--These are metered licenses that are valid for a limited period. The usage period of a license is based on a system clock. The evaluation licenses are built into the image and are not node locked. The evaluation licenses are used only when there are no permanent, extension or grace period licenses available for a feature. An end-user license agreement (EULA) has to be accepted before using an evaluation license.
- Extension licenses--Extension licenses are node-locked metered licenses. These licenses are installed using the management interfaces on the device. A EULA has to be accepted as part of installation.
- Grace-rehost licenses--Grace period licenses are node locked metered licenses. These licenses are installed on the device as part of the rehost operation. A EULA has to be accepted as a part of the rehost operation.

For all the license types, except the evaluation license, a EULA has to be accepted during the license installation. This means that all the license types except the evaluation license are activated after installation. In the case of an evaluation license, a EULA is presented during an SSL VPN gateway configuration or an SSL VPN context configuration.

An SSL VPN session corresponds to a successful login to the SSL VPN service. An SSL VPN session is created when a valid license is installed and the user credentials are successfully validated. On a successful user validation, a request is made to the licensing module to get a seat. An SSL VPN session is created only when the request is successful. If a valid license is not installed, the SSL VPN gateway configuration and SSL VPN context configurations are successful, but the user cannot login successfully. When multiple gateways and contexts are configured, the total number of sessions are equal to the total sessions allowed by the license.

The same user can create multiple sessions and for each session a seat count is reserved. The seat reservation does not happen in the following cases:

- Multiple TCP connections such as web server content, Outlook Web Access (OWA) and Common Intermediate Format (CIF) file shares.
- Port forward session initiation.
- Full tunnel session creation from a browser session.
- Full tunnel session is up and a crypto rekey is done.

When the total active sessions are equal to the maximum license count of the current active license, no more new sessions are allowed.

The reserved seat count or session is released when

- a user logs out.
- a Dead Peer Detection (DPD) failure happens.

- a session timeout occurs.
- an idle timeout occurs.
- a session is cleared administratively using the **clear webvpn session** command.
- disconnected from the tunnel.
- context is removed even when there are active sessions.

You can use the **show webvpn license** command to display the available count and the current usage. To display the current license type and time period left in case of a nonpermanent license, use the **show license** command. To get information related to license operations, events, and errors, use the **debug webvpn license** command.

For migrating from any Cisco IOS 12.4T release to Cisco IOS 15.x release, use the license migration tool at https://tools.cisco.com/SWIFT/Licensing/LicenseAdminServlet/migrateLicense.

New Cisco IOS SSL VPN licenses that are generated are cumulative. Therefore the old licenses become inactive when a new license is applied. For example, when you are upgrading your license from 10 counts to 20 counts (an increase of 10 counts on the current 10 counts), Cisco provides a single 20 count license. The old license for 10 counts is not required when a permanent license for a higher count is available. However, the old license will exist in an inactive state as there is no reliable method to clear the old license.

In Cisco IOS Release 15.1(4)M1 and later releases, a Crypto Export Restrictions Manager (CERM) license is reserved only after the user logs in. If you have an Integrated Services Router Generation 2 (ISR G2) router with a CERM license, you must upgrade to Cisco IOS Release 15.1(4)M1 or later releases. Before Cisco IOS Release 15.1(4)M1, a CERM license is reserved for every SSL or Transport Layer Security (TLS) session.

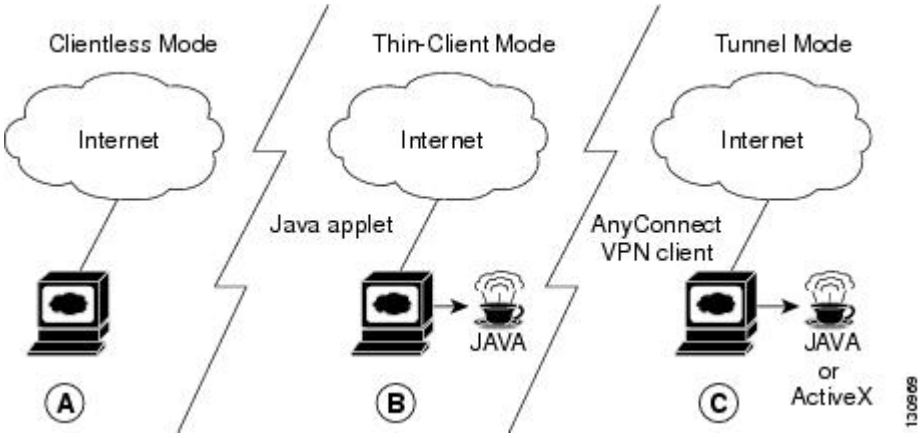
Modes of Remote Access

Remote Access Overview

End-user login and authentication is performed by the web browser to the secure gateway using an HTTP request. This process creates a session that is referenced by a cookie. After authentication, the remote user is shown a portal page that allows access to the SSL VPN networks. All requests sent by the browser include the authentication cookie. The portal page provides all the resources available on the internal networks. For example, the portal page could provide a link to allow the remote user to download and install a thin-client Java applet (for TCP port forwarding) or a tunneling client.

The figure below shows an overview of the remote access modes.

Figure 2 Modes of Remote Access Overview



The following table summarizes the level of SSL VPN support that is provided by each access mode.

Table 1 Access Mode Summary

A-- Clientless Mode	B--Thin-Client Mode	C--Tunnel Mode
• Browser-based (clientless) • Microsoft Windows or Linux • Web-enabled applications, file sharing, Outlook Web Access • Gateway performs address or protocol conversion and content parsing and rewriting	• TCP port forwarding • Uses Java Applet • Extends application support • Telnet, e-mail, SSH, Meeting Maker, Sametime Connect • Static port-based applications	• Works like “clientless” IPsec VPN • Tunnel client loaded through Java or ActiveX (approximately 500 kB) • Application agnostic--supports all IP-based applications • Scalable • Local administrative permissions required for installation

Clientless Mode

In clientless mode, the remote user accesses the internal or corporate network using the web browser on the client machine. The PC of the remote user must run the Windows 2000, Windows XP, or Linux operating systems.

The following applications are supported in clientless mode:

- Web browsing (using HTTP and HTTPS)--provides a URL box and a list of web server links in the portal page that allows the remote user to browse the web.
- File sharing (using common Internet file system [CIFS])--provides a list of file server links in the portal page that allows the remote user to do the following operations:
  - Browse a network (listing of domains)
  - Browse a domain (listing of servers)
  - Browse a server (listing of shares)
  - List the files in a share
  - Create a new file
  - Create a directory
  - Rename a directory
  - Update a file
  - Download a file
  - Remove a file
  - Rename a file

Note Linux requires that the Samba application is installed before CIFS file shares can be remotely accessed.

- Web-based e-mail, such as Microsoft Outlook Web Access (OWA) 2003 (using HTTP and HTTPS) with Web Distributed Authoring and Versioning (WebDAV) extensions--provides a link that allows the remote user to connect to the exchange server and read web-based e-mail.

Thin-Client Mode

Thin-client mode, also called TCP port forwarding, assumes that the client application uses TCP to connect to a well-known server and port. In thin-client mode, the remote user downloads a Java applet by clicking the link provided on the portal page, or the Java applet is downloaded automatically (see the Options for Configuring HTTP Proxy and the Portal Page, page 10 section). The Java applet acts as a TCP proxy on the client machine for the services that you configure on the gateway.

The applications that are supported in thin-client mode are mainly e-mail-based (SMTP, POP3, and Internet Map Access Protocol version 4 [IMAP4]) applications.

Note The TCP port-forwarding proxy works only with the Sun Microsystems Java Runtime Environment (JRE) version 1.4 or later versions. A Java applet is loaded through the browser that verifies the JRE version. The Java applet will refuse to run if a compatible JRE version is not detected.

The Java applet initiates an HTTP request from the remote user client to the SSL VPN gateway. The name and port number of the internal e-mail server is included in the HTTP request (POST or CONNECT). The SSL VPN gateway creates a TCP connection to that internal e-mail server and port.

The Java applet starts a new SSL connection for every client connection.

You should observe the following restrictions when using thin-client mode:

- The remote user must allow the Java applet to download and install.
- You cannot use thin-client mode for applications such as FTP, where the ports are negotiated dynamically. You can use TCP port forwarding only with static ports.

Note There is a known compatibility issue with the encryption type and Java. If the Java port-forwarding applet does not download properly and the configuration line **ssl encryption 3des-sha1 aes-sha1** is present, you should remove the line from the WebVPN gateway subconfiguration.

- Options for Configuring HTTP Proxy and the Portal Page, page 10

Options for Configuring HTTP Proxy and the Portal Page

Effective with Cisco IOS Release 12.4(11)T, administrators have more options for configuring the HTTP proxy and the portal page. If HTTP proxy is enabled, the Java applet acts as the proxy for the browser of the user, thereby connecting the client workstation with the gateway. The home page of the user (as defined by the user group) is opened automatically or, if configured by the administrator, the user is directed to a new website.

HTTP proxy supports both HTTP and HTTPS.

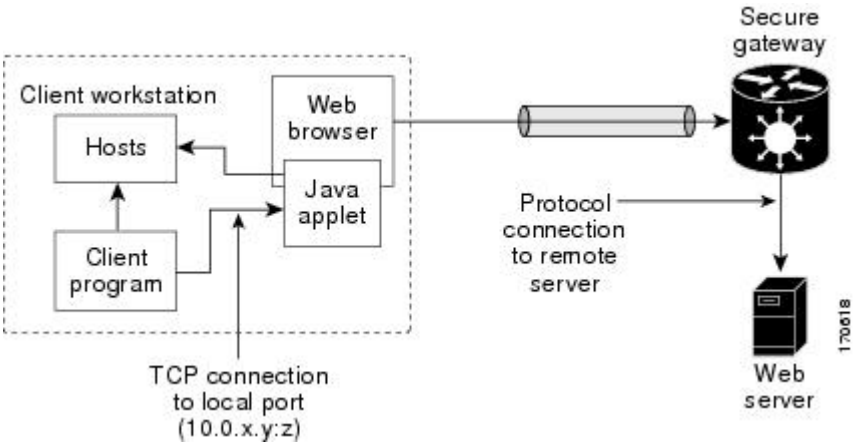
Benefits of Configuring HTTP Proxy

HTTP supports all client-side web technologies (including HTML, Cascading Style Sheets [CSS], JavaScript, VBScript, ActiveX, Java, and flash), HTTP Digest authentication, and client certificate authentication. Remote users can use their own bookmarks, and there is no limit on cookies. Because there is no mangling involved and the client can cache the objects, performance is much improved over previous options for configuring the HTTP proxy and portal page.

Illustrations of Port Forwarding with and Without an HTTP Proxy Configuration

The figure below illustrates TCP port forwarding without HTTP proxy configured.

Figure 3 TCP Port Forwarding Without HTTP Proxy Configured



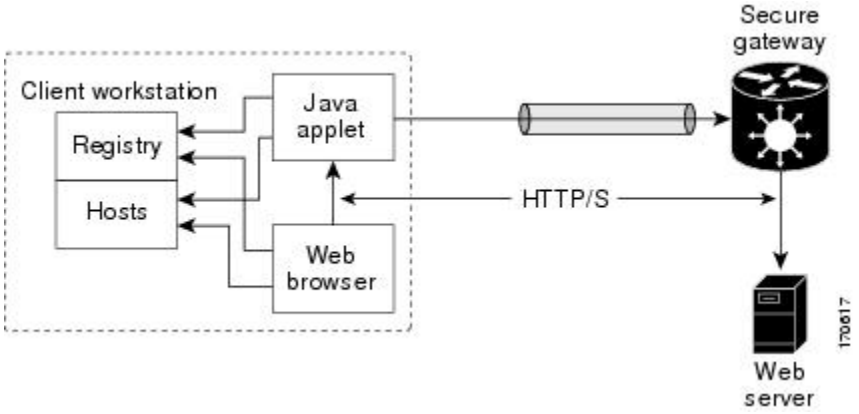
In the figure above, the following steps occur:

1. User downloads the proxy applet.
2. Applet updates the registry to add HTTP as a Remote Procedure Call (RPC) transport.
3. Applet examines the registry to determine the exchange (and local catalog) server and create server entries that refer to those servers.

4 Applet opens local port 80 and listens for connections.
5 User starts Outlook, and Outlook connects to 10.0.0.254:80.
6 Applet opens a connection to the secure gateway and delivers the requests from Outlook.
7 Secure gateway examines the requests to determine the endpoint exchange server.
8 Data flows from Outlook, through the applet and the secure gateway, to the exchange server.
9 User terminates Outlook.
10 User closes the applet. Before closing, the applet undoes configuration Steps 3 and 4.

The figure below illustrates TCP port forwarding when HTTP proxy is configured.

Figure 4 HTTP Proxy



In the figure above, the following steps occur:

1 Proxy applet is downloaded automatically.
2 Applet saves the original proxy configuration of the browser.
3 Applet updates the proxy configuration of the browser to be the local loopback address with an available local port (by default, port 8080).
4 Applet opens the available local port and listens for connections.
5 Applet, if so configured, opens the home page of the user, or the user browses to a new website.
6 Applet accepts and looks at the HTTP or HTTPS request to determine the destination web server.
7 Applet opens a connection to the secure gateway and delivers the requests from the browser.
8 Secure gateway examines the requests to determine the endpoint web server.
9 Data flows from the browser, through the applet and the secure gateway, to the web server.
10 User closes applet. Before closing, the applet undoes configuration Steps 2 and 3.

Note HTTP proxy can also be enabled on an authentication, authorization, and accounting (AAA) server. See the table SSL VPN RADIUS Attribute-Value Pairs in the Configuring RADIUS Attribute Support for SSL VPN, page 51 section (port-forward-http-proxy and port-forward-http-proxy-url attributes).

Tunnel Mode

In a typical clientless remote access scenario, remote users establish an SSL tunnel to move data to and from the internal networks at the application layer (for example, web and e-mail). In tunnel mode, remote users use an SSL tunnel to move data at the network (IP) layer. Therefore, tunnel mode supports most IP-

based applications. Tunnel mode supports many popular corporate applications (for example, Microsoft Outlook, Microsoft Exchange, Lotus Notes E-mail, and Telnet).

The tunnel connection is determined by the group policy configuration. The Cisco AnyConnect VPN Client is downloaded and installed on the remote user PC, and the tunnel connection is established when the remote user logs into the SSL VPN gateway.

By default, the Cisco AnyConnect VPN Client is removed from the client PC after the connection is closed. However, you have the option to keep the Cisco AnyConnect VPN Client installed on the client PC.

SSL VPN Features

Access Control Enhancements

Effective with Cisco IOS Release 12.4(20)T, administrators can configure automatic authentication and authorization for users. Users provide their usernames and passwords via the gateway page URL and do not have to reenter their usernames and passwords from the login page. Authorization is enhanced to support more generic authorization, including local authorization. In previous releases, only RADIUS authorization was supported.

For information about configuring this feature, see the Configuring Automatic Authentication and Authorization, page 88 section.

SSL VPN Client-Side Certificate-Based Authentication

This feature enables SSL VPN to authenticate clients based on the client's AAA username and password and also supports WebVPN gateway authentication of clients using AAA certificates.

SSL VPN Client-Side Certificate-Based Authentication feature includes the following features:

Certificate-Only Authentication and Authorization Mode

Certificate-only authorization requires the user to provide a AAA authentication certificate as part of the WebVPN request, but does not require the username and password for authorization. The user requests WebVPN access with the AAA authentication certificate from the WebVPN gateway. The WebVPN gateway validates the identity of the client using the AAA authentication certificate presented to it. The WebVPN extracts the username from the AAA authentication certificate presented to it and uses it as the username in the AAA request. AAA authentication and AAA authorization are then completed with a hard-coded password. To configure certificate-only authorization use the **authentication certificate** command.

Two-Factor Authentication and Authorization Mode

Two-factor authorization requires the user to request WebVPN access and present a AAA authentication certificate. The AAA authentication certificate is validated and the client's identity is verified. The WebVPN gateway then presents the login page to the user. The user enters their username and password and WebVPN sends AAA authentication and AAA authorization requests to the AAA server. The AAA authentication list and the AAA authorization lists configured on the server are then used for authentication and authorization. To configure two-factor authentication and authorization mode use the **authentication certificate aaa** command.

Note If the **username-prefill** command is configured, the username textbox on the login page will be disabled. The user will be asked only for their password on the login page.

Identification of WebVPN Context at Runtime Using Certificate Map Match Rules

Certificate map match rules are used by SSL VPN to identify the WebVPN context at runtime. The WebVPN context is required for AAA authentication and authorization mode and trustpoint configuration. When the user does not provide the WebVPN context, the identification of the WebVPN context at runtime is possible using certificate map matching by matching the certificate presented by the client with the certificate map match rules. To configure certificate map matching in WebVPN use the **match-certificate** command.

Support for AnyConnect Client to Implement Certificate Matching Based on Client Profile Attributes

Cisco AnyConnect client has certificate match functionality allowing it to select a suitable certificate while initiating tunnel connection with SSL VPN. In the case of standalone mode, the certificate selection is made based on the certificate match. When selecting a certificate, Cisco AnyConnect client can select the

appropriate certificate based on the AnyConnect client profile attributes. This requires SSL VPN to support AnyConnect client profiles. The profile file is imported after modification by the administrator using the **svc profile** command. To create an AnyConnect client profile use the template that appears after installing Cisco AnyConnect in this location: \Documents and Settings\All Users\Application Data\Cisco\ CiscoAnyConnectVPNClient\Profile\AnyConnectProfile.tmpl.

The following are the certificate match types available with Cisco AnyConnect client:

Certificate Key Usage Matching

Certificate key usage matching offers a set of constraints based on the broad types of operations that can be performed with a given certificate.

Extended Certificate Key Usage Matching

This matching allows an administrator to limit the certificates that can be used by the client based on the Extended Key Usage fields.

Certificate Distinguished Name Mapping

This certificate matching capability allows an administrator to limit the certificates that can be used by the client to those matching the specified criteria and criteria match conditions. This includes the ability to specify that a certificate must or must not have a specified string and also if wild carding for the string should be allowed.

AnyConnect Client Support

Effective with Cisco IOS Release 12.4(20)T, AnyConnect Client support is added for several client-side platforms, such as Microsoft Windows, Apple-Mac, and Linux. The ability to install AnyConnect in a standalone mode is also added. In addition, the Release 12.4(20)T allows you to install multiple AnyConnect VPN client packages to a gateway. For information on configuring multiple packages, see the section “Configuring the SSL VPN Gateway to Distribute CSD and Cisco AnyConnect VPN Client Package Files.”

Application ACL Support

Effective with Cisco IOS Release 12.4(11)T, the Application ACL Support feature provides administrators with the flexibility to fine-tune access control at the application layer level, for example, on the basis of a URL.

For information about configuring this feature, see the Configuring ACL Rules, page 75, and Associating an ACL Attribute with a Policy Group, page 78 sections.

Automatic Applet Download

Effective with Cisco IOS Release 12.4(9)T, administrators have the option of automatically downloading the port-forwarding Java applet. The Automatic Applet Download feature must be configured on a group policy basis.

Note Users still have to allow the Java applet to be downloaded. The dialog box appears, asking for permission.

To configure the automatic download, see the Configuring an SSL VPN Policy Group, page 43 section.

Backend HTTP Proxy

The Backend HTTP Proxy feature, added in Cisco IOS Release 12.4(20)T, allows administrators to route user requests through a backend HTTP proxy, providing more flexibility and control than routing requests through internal web servers. This feature adds the following new AAA attributes:

```
http-proxy-server
http-proxy-server-port
```

For information about configuring this feature, see the Configuring a Backend HTTP Proxy, page 92 section.

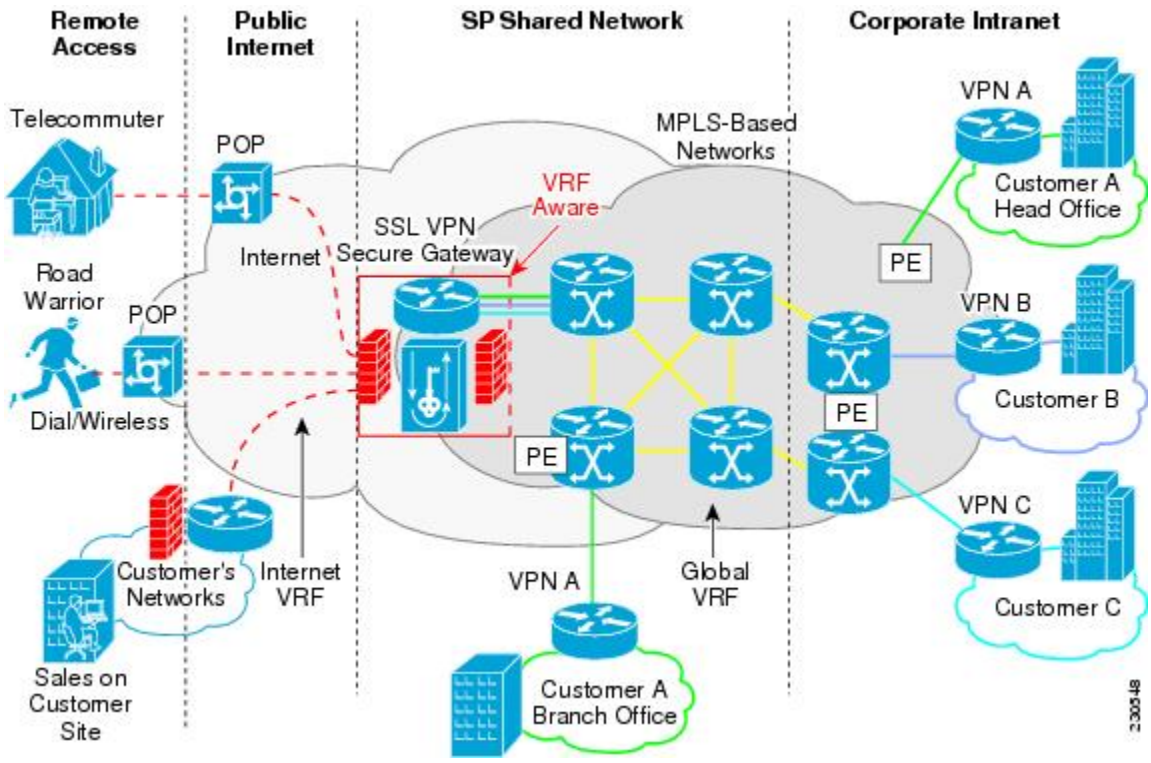
Front-Door VRF Support

Effective with Cisco IOS Release 12.4(15)T, front-door virtual routing and forwarding (FVRF) support, coupled with the already supported internal virtual routing and forwarding (IVRF), provides for increased security. The feature allows the SSL VPN gateway to be fully integrated into a Multiprotocol Label Switching (MPLS) or non-MPLS network (wherever the VRFs are deployed). The virtual gateway can be placed into a VRF that is separate from the Internet to avoid internal MPLS and IP network exposure. This placement reduces the vulnerability of the router by separating the Internet routes or the global routing table. Clients can now reach the gateway by way of the FVRF, which can be separate from the global VRF. The backend, or IVRF, functionality remains the same.

This FVRF feature provides for overlapping IP addresses.

The figure below is a scenario in which FVRF has been applied.

Figure 5 Scenario in Which FVRF Has Been Applied



To configure FVRF, see the Configuring FVRF, page 85 section.

Full-Tunnel Cisco Express Forwarding Support

Effective with Cisco IOS Release 12.4(20)T, Full-Tunnel Cisco Express Forwarding support is added for better throughput performance than in earlier releases. This feature is enabled by default. To turn off full-tunnel Cisco Express Forwarding support, use the **no webvpn cef** command.

Note To take full advantage of Cisco Express Forwarding support, the hardware crypto engine is required.

For sample output showing Cisco Express Forwarding-processed packets, see the Example: Cisco Express Forwarding-Processed Packets, page 125.

Network Address Translation (NAT) configuration is sometimes used to forward TCP port 443 traffic destined to the WAN interface of a router through an internal webserver.

There are two methods of implementing Cisco IOS SSL VPN on a preexisting NAT configuration. The Cisco-recommended method is to use the WebVPN gateway IP address as the secondary address on the WAN interface. This method helps improve the WebVPN throughput performance. The following is a sample configuration of the recommended method on Cisco IOS SSL VPN:

```
interface GigabitEthernet 0/0
  ip address 10.1.1.1 255.255.255.0
  ip address 10.1.1.2 255.255.255.0 secondary !
webvpn gateway ssl_vpn
  ip address 10.1.1.2 port 443
```

In the second method the WebVPN gateway uses a private IP address configured on a loopback interface and performs a NAT operation to convert the private IP address to a publically routable address. The following configuration is not supported on Cisco IOS SSL VPN because this configuration causes packets to become process-switched instead of being Cisco Express Forwarding-switched:

```
interface Loopback 10
  ip address 192.0.2.1 255.255.255.0
!
interface GigabitEthernet 0/0
  description WAN interface
  ip address 10.1.1.1 255.0.0.0
!
ip nat inside source static 192.0.2.1 10.1.1.2 !
webvpn gateway ssl_vpn
  ip address 192.0.2.1 port 443
```

GUI Enhancements

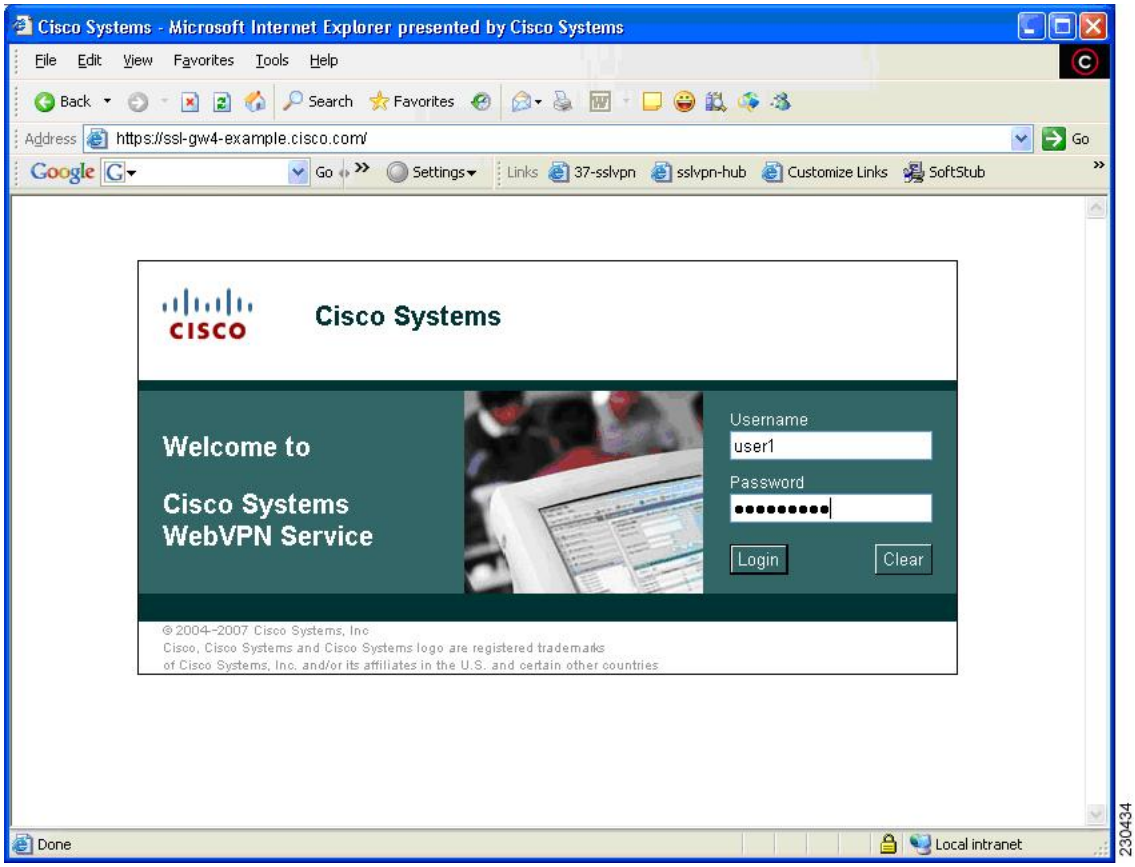
In Cisco IOS Release 12.4(15)T, ergonomic improvements are made to the GUI of the Cisco IOS SSL VPN gateway. The improved customization of the user interface provides for greater flexibility and the ability to tailor portal pages for individualized views. Enhancements are made to the following web screens:

- Login Screen, page 16
- Banner, page 17
- Customization of a Login Page, page 18
- Portal Page, page 18

Login Screen

The figure below is an example of a typical login screen.

Figure 6 Typical Login Screen



Banner

The banner is a small popup box (see GUID-13305E90-FC6F-436E-A2F0-379CF6BFF4EE9) that appears after the user is logged in and before the portal page appears.

The message in the popup box is configured using the **banner** command.

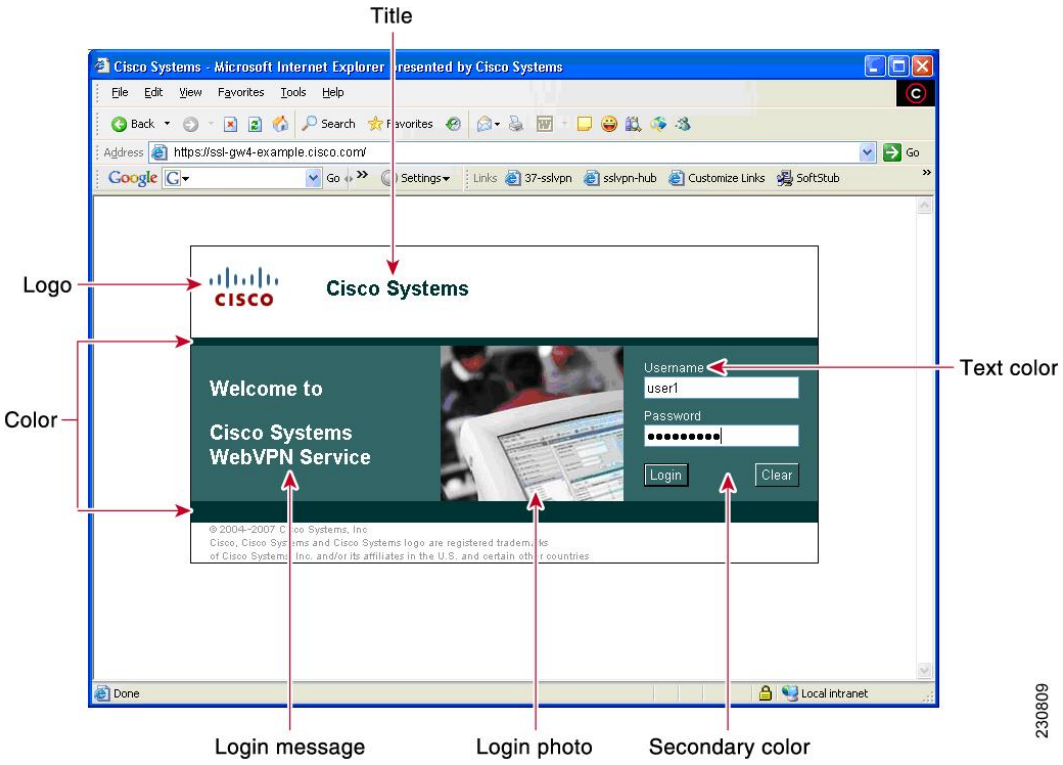
Figure 7 Banner

Customization of a Login Page

Login screens can be customized by an administrator. GUID-0474E4CE-EA60-4496-9BB4-D1C9C613A15F3 shows the fields that can be customized.

For information about setting various elements of the login page, see the document *Cisco IOS Security Command Reference*, Release 12.4T, for the **color**, **logo**, **login-message**, **login-photo**, **secondary-color**, **text-color**, **title**, **title-color**, and **text-color** commands.

Figure 8 Login Page with Callouts of the Fields That Can Be Customized



Portal Page

The portal page (see the figure below) is the main page for the SSL VPN functionality. You can customize this page to contain the following:

- Custom logo (the default is the Cisco bridge logo)
- Custom title (the default is “WebVPN Services”)
- Custom banner (the default is an empty string)
- Custom colors (the default is a combination of white and greens)
- List of web server links (can be customized)

Note The Bookmark links are listed under the Personal folder, and the server links are listed under Network File in the figure below.

- URL entry box (may be present or can be hidden using the **hide-url-bar** command)
- Thin Client link (may or may not be present)

Note The Application Access box allows you to download and install the Tunnel Connection and Thin Client Application.

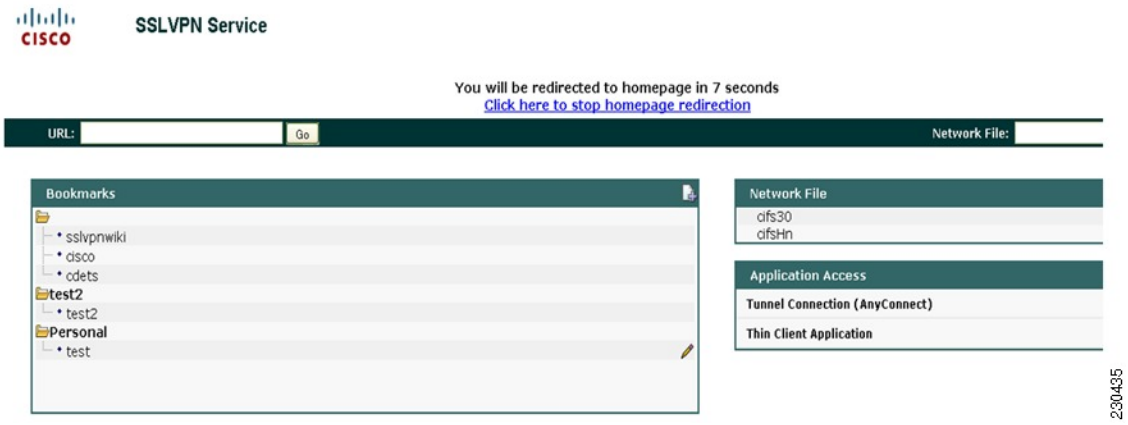
- Links for Help, Home (that is, the portal page), and Logout

Items that you have not configured are not displayed on the portal page.

Note E-mail access is supported by thin-client mode, which is downloaded using the Thin Client link.

The figure below is an example of a typical portal page.

Figure 9 Typical Portal Page



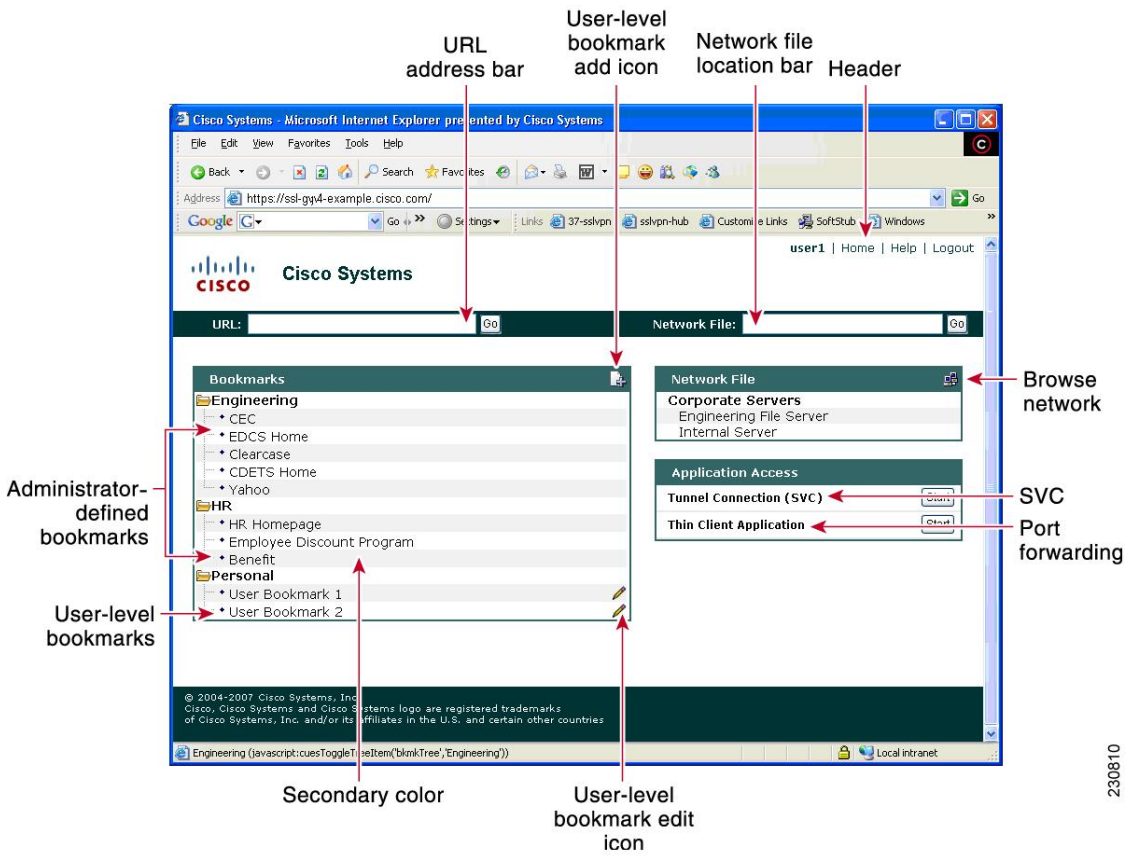
Note Time to redirect to the home page is displayed on the WebVPN portal page if you have configured the home page redirect time using the **webvpn-homepage** command. See the *Cisco IOS Security Command Reference* for information about the **webvpn-homepage** command. You can click the "Click here to stop homepage redirection" link to stop redirection.

Customization of a Portal Page

Portal pages can be customized by an administrator. GUID-3652D4EB-E7DF-4CC2-B3C3-4A55FD48A247D shows various fields, including the fields that can be customized by an administrator. The fields that can be customized by an administrator are as follows:

- Title
- Logo
- Secondary color
- Administrator-defined bookmarks
- Color

Figure 10 Portal Page with Callouts of Various Fields, Including Those That Can Be Customized



The table below provides information about various fields on the portal page. For information about setting elements such as color or titles, see command information in the *Cisco IOS Security Command Reference*, Release 12.4T, for the **color**, **functions**, **hide-url-bar**, **logo**, **port-forward**, **title**, **title-color**, **secondary-color**, **secondary-text-color**, and **url-list** commands.

Table 2 Information About Fields on the Portal Page

Field	Description
User-level bookmark add icon	If a user clicks it, a dialog box is added so that a new bookmark can be added to the Personal folder.

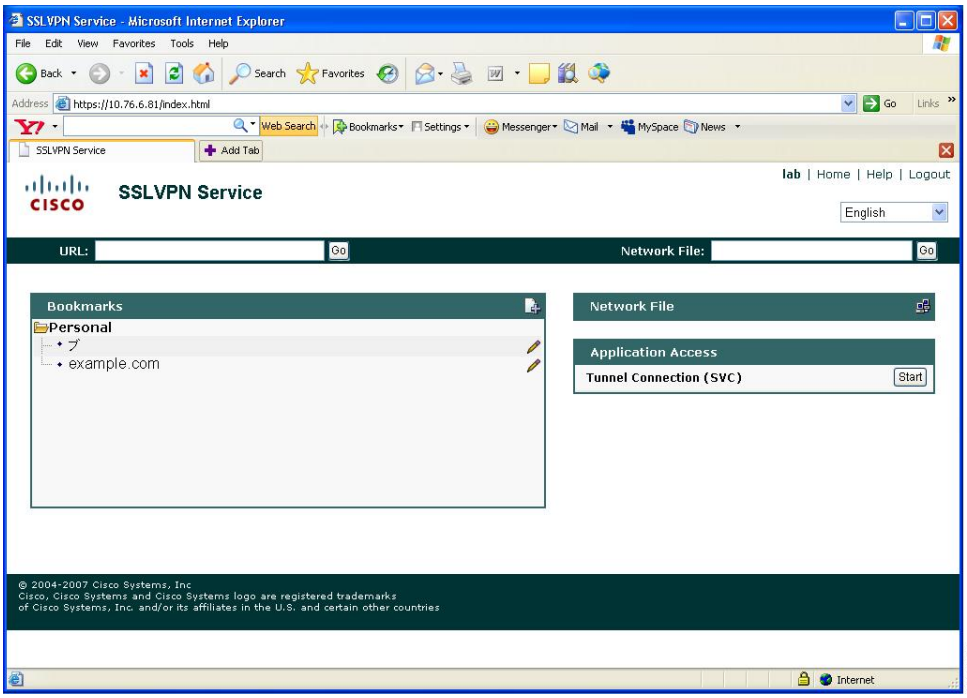
Field	Description
Network File location bar	A user can enter the file server here. Both of the **functions file-access** and **functions file-entry** commands must be configured for the input box to appear.
Header	Shares the same color value as the title.
Last login	Time stamp of the last login.
Browse network	Allows a user to browse the file network. The **functions file-access** and **functions file-browse** commands must be configured for the icon to appear.
Tunnel Connection	A user can choose when to start the tunnel connection by configuring the **functions svc-enabled** command.
Port forwarding	Downloads the applet and starts port forwarding.
User-level bookmark edit icon	Allows a user to edit or delete an existing bookmark.
User-level bookmarks	A user can add a bookmark by using the plus icon  on the bookmark panel or toolbar. See the document *SSL VPN Remote User Guide* for information about the toolbar. A new window is opened when the link is clicked.
Administrator-defined bookmarks	Administrator-defined URL lists cannot be edited by the user.
URL address bar	A new window is opened when a user clicks Go.

Internationalization

The Internationalization feature provides multilanguage support for messages initiated by the headend for SSL VPN clients, such as Cisco Secure Desktop (CSD) and SSL VPN Client (SVC). With the Internationalization feature, administrators can import their own attribute files in an XML format so that other languages can be imported using an editor that supports multilanguages.

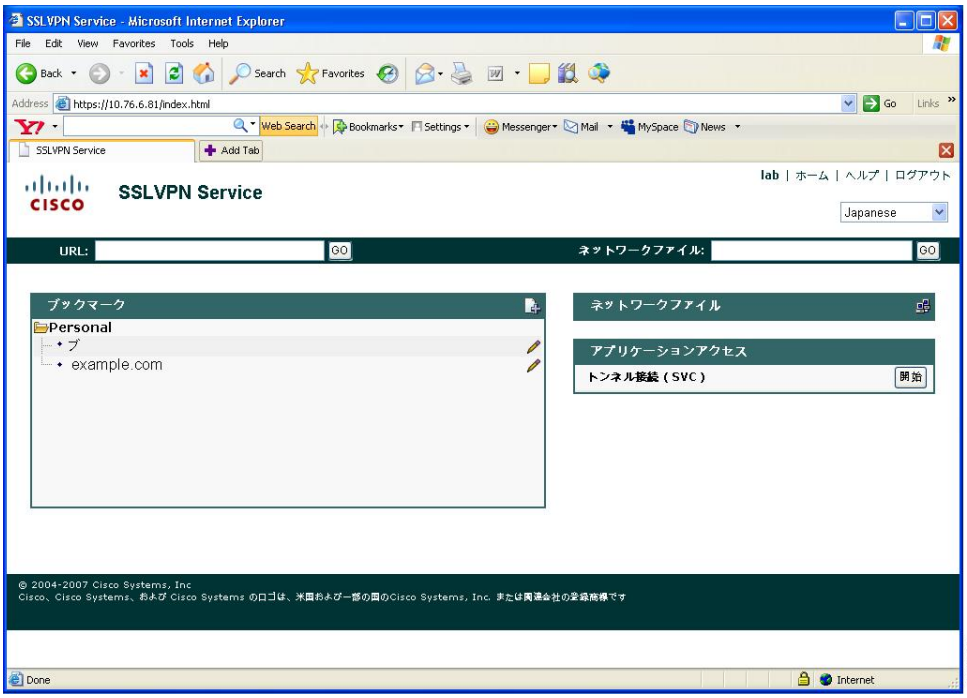
The figure below shows a portal page in English. Users can select any language you have imported for certain SSL VPN web pages (login message, title page, and URL lists).

Figure 11 Portal Page in English



The figure below shows that an administrator has imported files in Japanese. A user has selected Japanese as the language for certain SSL VPN web pages (login message, title, and URL lists).

Figure 12 Portal Page in Japanese



For information about configuring this feature, see the Configuring Internationalization, page 94 section. For examples relating to this feature, see the Example: Internationalization, page 127 section.

Max-User Limit Message

A user that tries to log in to a Web VPN context when the maximum user limit has been reached receives a "Max-user limit reached" message.

Netegrity Cookie-Based Single SignOn Support

The Netegrity SiteMinder product provides a Single SignOn feature that allows a user to log in a single time for various web applications. The benefit of this feature is that users are prompted to log in only once. This feature is accomplished by setting a cookie in the browser of a user when the user initially logs in.

Effective with Cisco IOS Release 12.4(11)T, Netegrity cookie-based SSO is integrated with SSL VPN. It allows administrators to configure an SSO server that sets a SiteMinder cookie in the browser of a user when the user initially logs in. This cookie is validated by a SiteMinder agent on subsequent user requests to resources that are protected by a SiteMinder realm. The agent decrypts the cookie and verifies whether the user has already been authenticated.

For information about configuring SSO Netegrity Cookie Support and associating it with a policy group using the CLI, see the sections Configuring SSO Netegrity Cookie Support for a Virtual Context, page 79 and Associating an SSO Server with a Policy Group, page 81section.

An SSO server can also be associated with a policy group using RADIUS attributes, as in the following example:

```
webvpn:sso-server-name=server1
```

For a list of RADIUS attribute-value (AV) pairs that support SSL VPN, see the Configuring RADIUS Attribute Support for SSL VPN, page 51 section.

NTLM Authentication

NT LAN Manager (NTLM) is supported for SSL VPN effective with Cisco IOS Release 12.4(9)T. The feature is configured by default.

RADIUS Accounting

Effective with Cisco IOS Release 12.4(9)T, this feature provides for RADIUS accounting of SSL VPN user sessions.

For information about configuring SSL VPN RADIUS accounting for SSL VPN user sessions, see the Configuring RADIUS Accounting for SSL VPN User Sessions, page 50 section.

For more information about configuring RADIUS accounting, see the Configuring RADIUS module in the *Cisco IOS Security Configuration Guide: Securing User Services*.

For a list of RADIUS AV pairs that support SSL VPN, see the Configuring RADIUS Attribute Support for SSL VPN, page 51 section.

Stateless High Availability with Hot Standby Router Protocol

Hot Standby Router Protocol (HSRP) provides high network availability by routing IP traffic from hosts on Ethernet networks without having to rely on the availability of any single router. HSRP is particularly useful for hosts that do not support a router discovery protocol, such as ICMP Router Discovery Protocol

(IRDP), and that do not have the functionality to switch to a new router when their selected router reloads or loses power. Without this functionality, a router that loses its default gateway because of a router failure is unable to communicate with the network.

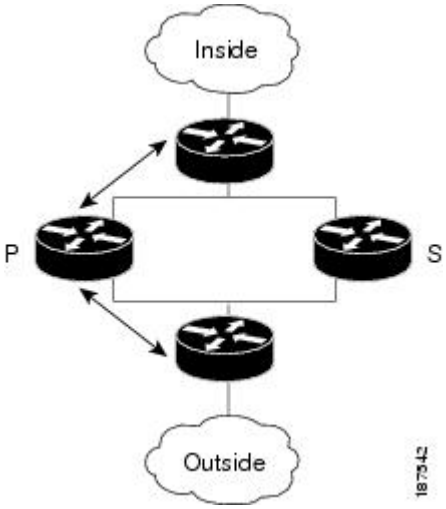
HSRP is configurable on LAN interfaces using standby CLI. It is possible to use the standby IP address from an interface as the local IPsec identity, or local tunnel endpoint.

You can use the standby IP address as the SSL VPN gateway address to apply failover to VPN routers by using HSRP. Remote SSLVPN users connect to the local VPN gateway using the standby address that belongs to the active device in the HSRP group. In the event of failover, the standby device takes over ownership of the standby IP address and begins to service remote VPN users.

Using the Stateless High Availability with Hot Standby Router Protocol feature, the remote user has to be aware of only the HSRP standby address instead of a list of gateway addresses.

The figure below shows the enhanced HSRP functionality topology. Traffic is serviced by the active Router P, the active device in the standby group. In the event of failover, traffic is diverted to Router S, the original standby device. Router S assumes the role of the new active router and takes ownership of the standby IP address.

Figure 13 Stateless High Availability with HSRP for SSL VPN



For information about configuring Stateless High Availability with HSRP, see the Configuring Stateless High Availability with HSRP for SSL VPN, page 93 section.

Note In the case of a failover, HSRP does not facilitate SSL VPN state information transfer between VPN gateways. Without this state transfer, existing SSL VPN sessions with the remote users will be deleted, requiring users to reauthenticate and establish SSL VPN sessions with the new active gateway.

TCP Port Forwarding and Thin Client

Note The TCP Port Forwarding and Thin Client feature requires the Java Runtime Environment (JRE) version 1.4 or later releases to properly support SSL connections.

Note Because this feature requires installing JRE and configuring the local clients, and because doing so requires administrator permissions on the local system, it is unlikely that remote users will be able to use applications when they connect from public remote systems.

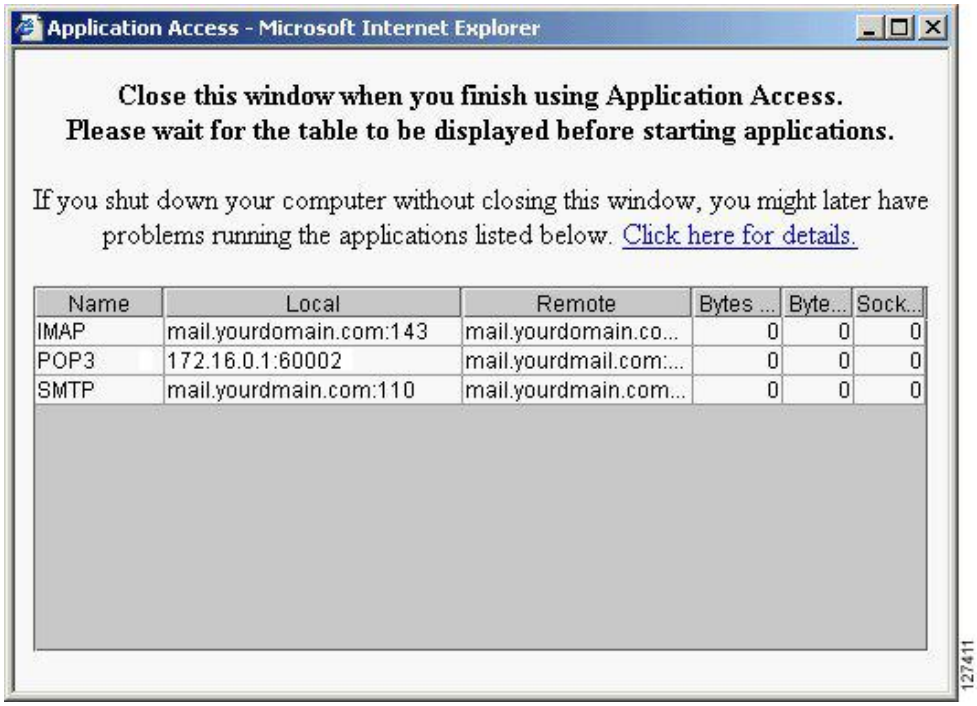
When the remote user clicks the Start button of the Thin Client Application (under "Application Access), a new window is displayed. This window initiates the downloading of a port-forwarding applet. Another window is then displayed. This window asks the remote user to verify the certificate with which this applet is signed. When the remote user accepts the certificate, the applet starts running, and port-forwarding entries are displayed (see the figure below). The number of active connections and bytes that are sent and received is also listed on this window.

Note When remote users launch Thin Client, their system may display a dialog box regarding digital certificates, and this dialog box may appear behind other browser windows. If the remote user connection hangs, tell the remote user to minimize the browser windows to check for this dialog box.

You should have configured IP addresses, Domain Name System (DNS) names, and port numbers for the e-mail servers. The remote user can then launch the e-mail client, which is configured to contact the e-mail servers and send and receive e-mails. POP3, IMAP, and SMTP protocols are supported.

The window attempts to close automatically if the remote user is logged out using JavaScript. If the session terminated and a new port forwarding connection is established, the applet displays an error message.

Figure 14 TCP Port Forwarding Page



Caution Users should always close the Thin Client window when finished using applications by clicking the close icon. Failure to quit the window properly can cause Thin Client or the applications to be disabled. See the section "Application Access--Recovering from Hosts File Errors" in the document *SSL VPN Remote User Guide*.

The table below lists remote system requirements for Thin Client.

Table 3 SSL VPN Remote System Thin-Client Requirements

Remote User System Requirements	Specifications or Use Suggestions
Client applications installed.	-
Cookies enabled on browser.	-
Administrator privileges.	You must be the local administrator on your PC.
Sun Microsystems JRE version 1.4 or later installed.	SSL VPN automatically checks for JRE whenever the remote user starts Thin Client. If it is necessary to install JRE, a popup window displays directing remote users to a site where it is available.
Client applications configured, if necessary. **Note** The Microsoft Outlook client does not require this configuration step.	To configure the client application, use the locally mapped IP address and port number of the server. To find this information, do the following: • Start SSL VPN on the remote system and click the Thin-Client link on the SSL VPN home page. The Thin-Client window is displayed. • In the Name column, find the name of the server that you want to use, and then identify its corresponding client IP address and port number (in the Local column). • Use this IP address and port number to configure the client application. The configuration steps vary for each client application.
Windows XP SP2 patch.	If you are running Windows XP SP2, you must install a patch from Microsoft that is available at the following address: http://support.microsoft.com/?kbid=884020 This is a known Microsoft issue.

URL Obfuscation

The URL Obfuscation feature provides administrators with the ability to obfuscate, or mask, sensitive portions of an enterprise URL, such as IP addresses, hostnames, or part numbers. For example, if URL masking is configured for a user, the URL in the address bar could have the port and hostname portion obfuscated, as in this example:

https://slvpn-gateway.examplecompany.com/http/cF9HxnBjRmSFEzBWpDtfXfigzL559MQo51Qj/cgi-bin/submit.p

For information about configuring this feature, see the Associating an SSO Server with a Policy Group, page 81 section.

URL Rewrite Splitter

Effective with Cisco IOS Release 12.4(20)T, the URL Rewrite Splitter feature allows administrators to mangle selective URLs. Mangling is a CPU-intensive and time-consuming process, so mangling only selective URLs can result in a savings of memory and time.

For information about configuring this feature, see the Configuring a URL Rewrite Splitter, page 91 section.

User-Level Bookmarking

Effective with Cisco IOS Release 12.4(15)T, users can bookmark URLs while connected through an SSL VPN tunnel. Users can access the bookmarked URLs by clicking the URLs.

User-level bookmarking is turned by default. There is no way to turn it off. To set the storage location, administrators can use the **user-profile location** command. If the **user-profile location** command is not configured, the location flash:/webvpn/{*context name*}/ is used.

Virtual Templates

A virtual template enables SSL VPN to interoperate with IP features such as Network Address Translation (NAT), firewall, and policy-based routing.

For information about configuring this feature, see the section Configuring a Virtual Template, page 102.

License String Support for the 7900 VPN Client

The Cisco IOS SSL VPN accepts license strings from Cisco IP Phones. Cisco IOS VPN concentrators support the VPN license type linksys-phone in order to support the Galactica VPN client on 79*x* 2 and 79*x* 5 phones.

In the case of a transformer platform, response to the license message (linksys-phone) will succeed if the license requirements are met. However, an Integrated Services Routers (ISR) router must always respond with a success message so that the Galactica VPN client can attempt to establish a VPN connection.

SSLVPN DVTI Support

The SSLVPN DVTI Support feature adds Dynamic Virtual Tunnel Interface (DVTI) support to the Secure Socket Layer Virtual Private Network (SSL VPN) and hence enables seamless interoperability with IP features such as Firewall, Network Address Translation (NAT), Access Control Lists (ACLs), and Virtual Routing and Forwarding (VRF). This feature also provides DVTI support, which allows IP feature configuration on a per-tunnel basis.

SSL VPN provides three modes to access a VPN: clientless, thin client, and full tunnel. The full tunnel mode uses an internal virtual interface to route the traffic to and from the SSL VPN tunnel. Before the SSL VPN DVTI Support feature was introduced, the virtual interface was created during the SSL VPN virtual interface configuration and users were not allowed to apply IP features to the SSL VPN traffic.

The SSLVPN DVTI Support feature uses a virtual template infrastructure to provide DVTI support for SSL VPN. IP features are configured in a virtual template that is associated with the SSL VPN or WebVPN

context. The IP features configured in the virtual template are used to create a virtual access interface that is internally used to tunnel SSL VPN traffic. Virtual templates in a WebVPN context are applied in two ways: per-context and per-tunnel.

Note You can configure any IP feature with SSL VPN. However, in the Cisco IOS Release 15.1(1)T, interoperability has been tested only with the firewall, NAT, ACL, policy-based routing (PBR), and VRF IP features.

The SSL VPN DVTI Support feature contains the following:

Prerequisites for SSLVPN DVTI Support

- You must have the IP features configured in a virtual template. See the Configuring a Virtual Template, page 102 section for information on configuring a virtual template.
- SSL VPN must be able to fetch configurations from the AAA server.
- The SSL VPN gateway and context configurations must be enabled and operational.
- If VRF is needed, configure it before creating the virtual template.

Restrictions for SSLVPN DVTI Support

- In order for a virtual template to work with SSL VPN, the **ip unnumbered** command must be configured on the virtual template.

Virtual Template Infrastructure

A generic interface template service is required with features such as stackability, Virtual Private Dialup Network (VPDN), Multilink PPP (MLP), and virtual profiles. Virtual template interface service delivers a generic interface template service. The virtual template interface, command buffer, and virtual access interface functions enables you to populate a virtual-access interface using a pre-defined configuration that is stored in a virtual template interface and security servers such as TACACS+ and RADIUS.

For example, in stackability, a virtual template interface is assigned to a stack group. Whenever a stack member needs a virtual interface, the virtual template interface service is called by a member to obtain a virtual access interface cloned with the same configuration as the configuration of the assigned virtual template interface.

In a virtual profile, the per-user configuration can be stored in a security server. That is, when the user dials in, the desired configuration can be cloned into the virtual access interface associated with the user. The virtual template service provides an application programming interface (API) for a virtual profile to clone a buffer of commands to a virtual access interface. The virtual profile does the actual interaction with the security server.

Note If you do not configure a virtual template, then the default virtual template (VT0) will be used for cloning the virtual access interface.

SSL VPN Phase-4 Features

The SSL VPN Phase-4 Features feature provides the following enhancements to the Cisco IOS Secure Sockets Layer Virtual Private Network (SSL VPN):

- ACL support for split tunneling
- IP mask for IP pool address assignment
- Undoing the renaming of AnyConnect or SSL VPN Client (SVC) Full Tunnel Cisco package during installation on a Cisco IOS router
- Adding per-user SSL VPN session statistics
- "Start before logon" option for the Cisco IOS SSL VPN headend

The SSL VPN Phase-4 features contains the following:

- Prerequisites for SSL VPN Phase-4 Features, page 29
- Full Tunnel Package, page 29
- SSL VPN per-User Statistics, page 29

Prerequisites for SSL VPN Phase-4 Features

You must use a valid K9 image to configure the SSL VPN Phase-4 Features.

Full Tunnel Package

When you install the AnyConnect or SVC full tunnel package using the **webvpn install svc** command on the Cisco IOS headend, the package name gets renamed to svc_pkg_<number>. This renaming omits package information and Base Station Ethernet (BSE) operating system information, and thus makes you difficult to remove or uninstall the package. This functionality was modified in Cisco IOS Release 15.1(1)T to retain the name during installation of the package.

The limit on the filename size on the Cisco IOS file system (IFS) is 120 bytes. Unless the package name is greater than this limit, the package name does not change. If the filename exceeds this limit, then the installation fails. The following error message is displayed on the router console:

```
Error: Package name exceeds 120 characters
```

SSL VPN per-User Statistics

Per-user statistics functionality provides an option to filter the cumulative statistics on a per-user basis for the Cisco IOS SSL VPN sessions. Use the **show webvpn session user** command to enable this functionality. This command is applicable only for user session statistics and tunnel statistics. See *Cisco Cisco IOS Security Command Reference* for more information on the **show webvpn session** command.

DTLS Support for IOS SSL VPN

The DTLS Support for IOS SSL VPN feature enables DTLS as a transport protocol for the traffic tunneled through SSL VPN.

An AnyConnect client with a Transport Layer Security (TLS) tunnel can face problems for real-time traffic and the traffic that is not sensitive to data loss, such as VoIP. This happens because of the delay introduced

by the TCP channel (AnyConnect client uses TLS over TCP channel). Also, when the TCP sessions are channeled over the TLS tunnel we have TCP in TCP. Here both the TCPs try to control the flow and achieve in-sequence reliable delivery. This causes slow down of the application and also increases the network bandwidth utilization. DTLS solves this problem by hosting TLS over UDP after making the necessary changes to TLS.

The DTLS Support for IOS SSL VPN feature is enabled by default on the Cisco IOS SSL VPN. You can use the **no svc dtls** command in the WebVPN group policy configuration mode to disable the DTLS support on the SSL VPN.

- Prerequisites for DTLS Support for IOS SSL VPN, page 30
- Restrictions for DTLS Support for IOS SSL VPN, page 30

Prerequisites for DTLS Support for IOS SSL VPN

You must use a valid K9 image to have the DTLS Support for IOS SSL VPN feature.

Restrictions for DTLS Support for IOS SSL VPN

- Cisco IOS gateway supports the DTLS Support for IOS SSL VPN feature only with an AnyConnect clients.
- The DTLS Support for IOS SSL VPN feature is supported on AnyConnect clients with version 2.x.
- The DTLS Support for IOS SSL VPN feature is not supported on SSL VPN Client (SVC) with version 1.x.

Cisco AnyConnect VPN Client Full Tunnel Support

- Remote Client Software from the SSL VPN Gateway, page 30
- Address Pool, page 30
- Manual Entry to the IP Forwarding Table, page 31

Remote Client Software from the SSL VPN Gateway

The Cisco AnyConnect VPN Client software package is pushed from the SSL VPN gateway to remote clients when support is needed. The remote user (PC or device) must have either the Java Runtime Environment for Windows (version 1.4 later), or the browser must support or be configured to permit Active X controls. In either scenario, the remote user must have local administrative privileges.

Address Pool

The address pool is first defined with the **ip local pool** command in global configuration mode. The standard configuration assumes that the IP addresses in the pool are reachable from a directly connected network.

Address Pools for Nondirectly Connected Networks

If you need to configure an address pool for IP addresses from a network that is not directly connected, perform the following steps:

1. Create a local loopback interface and configure it with an IP address and subnet mask from the address pool.

2. Configure the address pool with the **ip local pool** command. The range of addresses must fall under the subnet mask configured in Step 1.
3. Set up the route. If you are using the Routing Information Protocol (RIP), configure the **router rip** command and then the **network** command, as usual, to specify a list of networks for the RIP process. If you are using the Open Shortest Path First (OSPF) protocol, configure the **ip ospf network point-to-point** command in the loopback interface. As a third choice (instead of using the RIP or OSPF protocol), you can set up static routes to the network.
4. Configure the **svc address-pool** command with the name configured in Step 2.

Manual Entry to the IP Forwarding Table

If the SSL VPN software client is unable to update the IP forwarding table on the PC of the remote user, the following error message will be displayed in the router console or syslog:

```
Error : SSL VPN client was unable to Modify the IP forwarding table ......
```

This error can occur if the remote client does not have a default route. You can work around this error by performing the following steps:

1. Open a command prompt (DOS shell) on the remote client.
2. Enter the **route print** command.
3. If a default route is not displayed in the output, enter the **route** command followed by the **add** and **mask** keywords. Include the default gateway IP address at the end of the route statement. See the following example:

```
C:\>route ADD 0.0.0.0 MASK 0.0.0.0 10.1.1.1
```

Other SSL VPN Features

The following table lists the requirements for various SSL VPN features.

Table 4 SSL VPN Remote User System Requirements

Task	Remote User System Requirements	Additional Information
Web Browsing	Usernames and passwords for protected websites	Users should log out on SSL VPN sessions when they are finished. The look and feel of web browsing with SSL VPN might be different from what users are accustomed to. For example, when they are using SSL VPN, the following should be noted: • The SSL VPN title bar appears above each web page. • Websites can be accessed as follows: ◦ Entering the URL in the Enter Web Address field on the SSL VPN home page ◦ Clicking a preconfigured website link on the SSL VPN home page ◦ Clicking a link on a webpage accessed by one of the previous two methods Also, depending on how a particular account was configured, the following might have occurred: • Some websites are blocked. • Only the websites that appear as links on the SSL VPN home page are available.

Task	Remote User System Requirements	Additional Information
Network Browsing and File Management	File permissions configured for shared remote access Server name and passwords are necessary for protected file servers Domain, workgroup, and server names where folders and files reside	Only shared folders and files are accessible through SSL VPN. A user might not be familiar with how to locate files through the network of an organization. **Note** You should not interrupt the Copy File to Server operation or navigate to a different window while the copying is in progress. Interrupting this operation can cause an incomplete file to be saved on the server.
Using e-mail:Thin Client	Same requirements as for Thin Client (see the TCP Port Forwarding and Thin Client, page 24) section. Other Mail Clients **Note** If you use an IMAP client and lose the e-mail server connection or you are unable to make a new connection, you should close the IMAP application and restart SSL VPN.	To use e-mail, users must start Thin Client from the SSL VPN home page. The e-mail client is then available for use. Microsoft Outlook Express versions 5.5 and 6.0 have been tested. SSL VPN should support other SMTPS, POP3S, or IMAP4S e-mail programs, such as Netscape Mail, Lotus Notes, and Eudora, but they have not been verified.

Task	Remote User System Requirements	Additional Information
Using e-mail: Web Access	Web-based e-mail product installed	Supported products are as follows: • OWA 5.5, 2000, and 2003 Netscape, Mozilla, and Internet Explorer are supported with OWA 5.5 and 2000. Internet Explorer 6.0 or a later version is required with OWA 2003. Netscape and Mozilla are supported with OWA 2003. • Lotus Notes Operating system support: **Note** Later versions of the following browsers are also supported. • Microsoft Windows 2000, Windows XP, or Windows Vista • Macintosh OS X 10.4.6 • Linux (Redhat RHEL 3.0 +, FEDORA 5, or FEDORA 6) SSL VPN-supported browser: The following browsers have been verified for SSL VPN. Other browsers might not fully support SSL VPN features. **Note** Later versions of the following software are also supported. • Internet Explorer 6.0 or 7.0 • Firefox 2.0 (Windows and Linux) • Safari 2.0.3 Other web-based e-mail products should also work, but they have not been verified.

Task	Remote User System Requirements	Additional Information
Using the Cisco Tunnel Connection	--	To retrieve Tunnel Connection log messages using the Windows Event Viewer, go to Program Files > Administrative Tools > Event Viewer in Windows.
Using Secure Desktop Manager	A Secure Desktop Manager-supported browser	On Microsoft Windows: • Internet Explorer version 6.0 or 7.0 • Netscape version 7.2 On Linux: • Netscape version 7.2
Using Cache Cleaner or Secure Desktop	A Cisco Secure Desktop-supported browser	Any browser supported for Secure Desktop Manager.

Platform Support

For information about platform support for the SSL VPN feature, see the data sheet Cisco IOS SSL VPN (“Feature Availability” section).

How to Configure SSL VPN Services on a Router

Configuring an SSL VPN Gateway

The SSL VPN gateway acts as a proxy for connections to protected resources. Protected resources are accessed through an SSL-encrypted connection between the gateway and a web-enabled browser on a remote device, such as a personal computer. Entering the **webvpn gateway** command places the router in SSL VPN gateway configuration mode. The following configuration are accomplished in this task:

- The gateway is configured with an IP address.
- A port number is configured to carry HTTPS traffic (443 is default).
- A hostname is configured for the gateway.
- Crypto encryption and trust points are configured.
- The gateway is configured to redirect HTTP traffic (port 80) over HTTPS.
- The gateway is enabled.

The SSL VPN provides remote-access connectivity from almost any Internet-enabled location using only a web browser and its native SSL encryption. The **ssl encryption** command is configured to restrict the encryption algorithms that SSL uses in Cisco IOS software.

Note There is a known compatibility issue with the encryption type and Java. If the Java port-forwarding applet does not download properly and the configuration line **ssl encryption 3des-sha1 aes-sha1** is present, you should remove the line from the WebVPN gateway subconfiguration.

The configuration of the **ssl trustpoint** command is required only if you need to configure a specific certification authority (CA) certificate. A self-signed certificate is automatically generated when an SSL VPN gateway is put in service.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn gateway** *name*
4. **hostname** *name*
5. **ip address** *number* [**port** *number*] [**standby** *name*]
6. **http-redirect** [**port** *number*]
7. **ssl encryption** [**3des-sha1**] [**aes-sha1**] [**rc4-md5**]
8. **ssl trustpoint** *name*
9. **inservice**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn gateway** *name* **Example:** `Router(config)# webvpn gateway GW_1`	Enters WebVPN gateway configuration mode to configure an SSL VPN gateway. • Only one gateway is configured in an SSL VPN-enabled network.
Step 4	**hostname** *name* **Example:** `Router(config-webvpn-gateway)# hostname VPN_1`	(Optional) Configures the hostname for an SSL VPN gateway.
Step 5	**ip address** *number* [**port** *number*] [**standby** *name*] **Example:** `Router(config-webvpn-gateway)# ip address 10.1.1.1`	(Optional) Configures a proxy IP address on an SSL VPN gateway. • **port** --Specifies the port number for proxy traffic. A number from 1 to 65535 can be entered for the *number* argument. • **standby** --Indicates that the gateway is standby. A redundancy group name must be entered for the *name* argument.

	Command or Action	Purpose
Step 6	**http-redirect** [**port** *number*] **Example:** `Router(config-webvpn-gateway)# http-redirect`	(Optional) Configures HTTP traffic to be carried over HTTPS. • When this command is enabled, the SSL VPN gateway listens on port 80 and redirects HTTP traffic over port 443 or the port number specified with the **port** keyword.
Step 7	**ssl encryption** [**3des-sha1**] [**aes-sha1**] [**rc4-md5**] **Example:** `Router(config-webvpn-gateway)# ssl encryption rc4-md5`	(Optional) Specifies the encryption algorithm that the SSL protocol uses for SSL VPN connections. • The ordering of the algorithms specifies the preference.
Step 8	**ssl trustpoint** *name* **Example:** `Router(config-webvpn-gateway)# ssl trustpoint CA_CERT`	(Optional if a self-signed certificate is to be used.) Configures the certificate trust point on an SSL VPN gateway. **Tip** Entering the **no** form of this command configures the SSL VPN gateway to revert to using an autogenerated self-signed certificate.
Step 9	**inservice** **Example:** `Router(config-webvpn-gateway)# inservice`	(Optional) Enables an SSL VPN gateway. • A gateway cannot be enabled or put “in service” until a proxy IP address has been configured.

• What to Do Next, page 38

What to Do Next

SSL VPN context and policy group configurations must be configured before an SSL VPN gateway can be operationally deployed. Proceed to the section “Configuring an SSL VPN Context” to see information on SSL VPN context configuration.

Configuring a Generic SSL VPN Gateway

To configure a generic SSL VPN gateway, perform the following steps in privileged EXEC mode.

Note The advantage of this configuration over the one in the configuration task in the Configuring an SSL VPN Gateway, page 36 is that basic commands and context can be configured quickly using just the **webvpn enable** command.

SUMMARY STEPS

1. **enable**
2. **webvpn enable gateway-addr** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**webvpn enable gateway-addr** *ip-address* **Example:** `Router# webvpn enable gateway-addr 10.1.1.1`	Configures a generic SSL VPN gateway.

Configuring an SSL VPN Context

The SSL VPN context defines the virtual configuration of the SSL VPN. Entering the **webvpn context** command places the router in SSL VPN configuration mode. The following configurations are accomplished in this task:

- A gateway and domain is associated.
- The AAA authentication method is specified.
- A group policy is associated.
- The remote user portal (web page) is customized.
- A limit on the number users sessions is configured.
- The context is enabled.

The **ssl authenticate verify all** command is enabled by default when a context configuration is created. The context cannot be removed from the router configuration while an SSL VPN gateway is in an enabled state (in service).

A virtual hostname is specified when multiple virtual hosts are mapped to the same IP address on the SSL VPN gateway (similar to the operation of a canonical domain name). The virtual hostname differentiates host requests on the gateway. The host header in the HTTP message is modified to direct traffic to the virtual host. The virtual hostname is configured with the **gateway** command in WebVPN context configuration mode.

The SSL VPN gateway configuration has been completed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **aaa authentication** {**domain** *name* | **list** *name*}
5. **policy group** *name*
6. **exit**
7. **default-group-policy** *name*
8. **exit**
9. **gateway** *name* [**domain** *name* | **virtual-host** *name*]
10. **inservice**
11. **login-message** [*message-string*]
12. **logo** [**file** *filename* | **none**]
13. **max-users** *number*
14. **secondary-color** *color*
15. **secondary-text-color** {**black** | **white**}
16. **title** [*title-string*]
17. **title-color** *color*
18. **svc platform** {**lin** | **mac** | **win**} **seq** *sequence-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context. **Tip** The context can be optionally named using the domain or virtual hostname. This is recommended as a best practice. It simplifies the management of multiple context configurations.

	Command or Action	Purpose
Step 4	**aaa authentication** {**domain** *name* | **list** *name*} **Example:** `Router(config-webvpn-context)# aaa authentication domain SERVER_GROUP`	(Optional) Specifies a list or method for SSL VPN remote-user authentication. **Tip** If this command is not configured, the SSL VPN gateway will use global AAA parameters (if configured) for remote-user authentication.
Step 5	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	(Optional) Creates a policy group within the SSL VPN context and enters WebVPN group policy configuration mode. • Used to define a policy that can be applied to the user.
Step 6	**exit** **Example:** `Router(config-webvpn-group)# exit`	(Optional) Exits WebVPN group policy configuration mode.
Step 7	**default-group-policy** *name* **Example:** `Router(config-webvpn-context)# default-group-policy ONE`	(Optional) Associates a group policy with an SSL VPN context configuration. • This command is configured to attach the policy group to the SSL VPN context when multiple group policies are defined under the context. • This policy will be used as default, unless a AAA server pushes an attribute that specifically requests another group policy.
Step 8	**exit** **Example:** `Router(config-webvpn-context)# exit`	(Optional) Exits WebVPN context configuration mode.
Step 9	**gateway** *name* [**domain** *name* | **virtual-host** *name*] **Example:** `Router(config-webvpn-context)# gateway GW_1 domain cisco.com`	(Optional) Associates an SSL VPN gateway with an SSL VPN context.

Command or Action	Purpose
Step 10 **inservice** **Example:** `Router(config-webvpn-gateway)# inservice`	(Optional) Enables an SSL VPN context configuration. • The context is put "in service" by entering this command. However, the context is not operational until it is associated with an enabled SSL VPN gateway.
Step 11 **login-message** [*message-string*] **Example:** `Router(config-webvpn-context)# login-message "Please enter your login credentials"`	(Optional) Configures a message for the user login text box displayed on the login page.
Step 12 **logo** [**file** *filename* | **none**] **Example:** `Router(config-webvpn-context)# logo file flash:/mylogo.gif`	(Optional) Configures a custom logo to be displayed on the login and portal pages of an SSL VPN. • The source image file for the logo is a gif, jpg, or png file that is up to 255 characters in length (filename) and up to 100 KB in size. • The file is referenced from a local file system, such as flash memory. An error message will be displayed if the file is not referenced from a local file system. • No logo will be displayed if the image file is removed from the local file system.
Step 13 **max-users** *number* **Example:** `Router(config-webvpn-context)# max-users 500`	(Optional) Limits the number of connections to an SSL VPN that will be permitted.
Step 14 **secondary-color** *color* **Example:** `Router(config-webvpn-context)# secondary-color darkseagreen`	(Optional) Configures the color of the secondary title bars on the login and portal pages of an SSL VPN. • The value for the *color* argument is entered as a comma-separated red, green, blue (RGB) value, an HTML color value (beginning with a pound sign [#]), or the name of the color that is recognized in HTML (no spaces between words or characters). The value is limited to 32 characters. The value is parsed to ensure that it matches one of the following formats (using Perl regex notation): ◦ \#/x{6} ◦ \d{1,3},\d{1,3},\d{1,3} (and each number is from 1 to 255) ◦ \w+ • The default color is purple.

	Command or Action	Purpose
Step 15	**secondary-text-color** {**black** | **white**} **Example:** `Router(config-webvpn-context)# secondary-text-color white`	(Optional) Configures the color of the text on the secondary bars of an SSL VPN. • The color of the text on the secondary bars must be aligned with the color of the text on the title bar. • The default color is black.
Step 16	**title** [*title-string*] **Example:** `Router(config-webvpn-context)# title “Secure Access: Unauthorized users prohibited”`	(Optional) Configures the HTML title string that is shown in the browser title and on the title bar of an SSL VPN. • The optional form of the **title** command is entered to configure a custom text string. If this command is issued without entering a text string, a title will not be displayed in the browser window. If the **no** form of this command is used, the default title string “WebVPN Service” is displayed.
Step 17	**title-color** *color* **Example:** `Router(config-webvpn-context)# title-color darkseagreen`	(Optional) Specifies the color of the title bars on the login and portal pages of an SSL VPN. • The value for the *color* argument is entered as a comma-separated red, green, blue (RGB) value, an HTML color value (beginning with a pound sign [#]), or the name of the color that is recognized in HTML (no spaces between words or characters). The value is limited to 32 characters. The value is parsed to ensure that it matches one of the following formats (using Perl regex notation): ◦ \#/x{6} ◦ \d{1,3},\d{1,3},\d{1,3} (and each number is from 1 to 255) ◦ \w+ • The default color is purple.
Step 18	**svc platform** {**lin** | **mac** | **win**} **seq** *sequence-number* **Example:** `Router(config-webvpn-context)# svc platform lin seq 1`	(Optional) Configures the platform of an AnyConnect version per context. • If the **svc platform** command is not used, AnyConnect is configured in standalone mode. • The **seq** keyword assigns a priority number to an AnyConnect client in the same platform. The range of sequence-number argument is from 1 to 10.

• What to Do Next, page 43

What to Do Next

An SSL VPN policy group configuration must be defined before an SSL VPN gateway can be operationally deployed. Proceed to the Configuring an SSL VPN Policy Group, page 43 to see information on SSL VPN policy group configuration.

Configuring an SSL VPN Policy Group

The policy group is a container that defines the presentation of the portal and the permissions for resources that are configured for a group of remote users. Entering the **policy group** command places the router in

WebVPN group policy configuration mode. After it is configured, the group policy is attached to the SSL VPN context configuration by configuring the **default-group-policy** command. The following tasks are accomplished in this configuration:

- The presentation of the SSL VPN portal page is configured.
- A NetBIOS server list is referenced.
- A port-forwarding list is referenced.
- The idle and session timers are configured.
- A URL list is referenced.

Outlook Web Access (OWA) 2003 is supported by the SSL VPN gateway upon completion of this task. The Outlook Exchange Server must be reachable by the SSL VPN gateway via TCP/IP.

A URL list can be configured under the SSL VPN context configuration and then separately for each individual policy group configuration. Individual URL list configurations must have unique names.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **policy group** *name*
5. **banner** *string*
6. **hide-url-bar**
7. **nbns-list** *name*
8. **port-forward** *name* [**auto-download**[**http-proxy** [**proxy-url** *homepage-url*]] | **http-proxy** [**proxy-url** *homepage-url*] [**auto-download**]]
9. **timeout** {**idle** *seconds* | **session** *seconds*}
10. **url-list** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 5	**banner** *string* **Example:** `Router(config-webvpn-group)# banner “Login Successful”`	(Optional) Configures a banner to be displayed after a successful login.
Step 6	**hide-url-bar** **Example:** `Router(config-webvpn-group)# hide-url-bar`	(Optional) Prevents the URL bar from being displayed on the SSL VPN portal page.
Step 7	**nbns-list** *name* **Example:** `Router(config-webvpn-group)# nbns-list SERVER_LIST`	(Optional) Attaches a NetBIOS Name Service (NBNS) server list to a policy group configuration. • The NBNS server list is first defined in SSL VPN NBNS list configuration mode.
Step 8	**port-forward** *name* [**auto-download**[**http-proxy** [**proxy-url** *homepage-url*]] | **http-proxy** [**proxy-url** *homepage-url*] [**auto-download**]] **Example:** `Router(config-webvpn-group)# port-forward EMAIL auto-download http-proxy proxy-url "http://www.example.com"`	(Optional) Attaches a port-forwarding list to a policy group configuration. • **auto-download** --(Optional) Allows for automatic download of the port-forwarding Java applet on the portal page of a website. • **http-proxy** --(Optional) Allows the Java applet to act as a proxy for the browser of the user. • **proxy-url** --(Optional) Page at this URL address opens as the portal (home) page of the user. • *homepage-url* --URL of the home page.

	Command or Action	Purpose
Step 9	**timeout** {**idle** *seconds* | **session** *seconds*} **Example:** `Router(config-webvpn-group)# timeout idle 1800`	(Optional) Configures the length of time that a remote user session can remain idle or the total length of time that the session can remain connected. • Upon expiration of either timer, the remote user connection is closed. The remote user must log in (reauthenticate) to access the SSL VPN.
Step 10	**url-list** *name* **Example:** `Router(config-webvpn-group)# url-list ACCESS`	(Optional) Attaches a URL list to policy group configuration.

• What to Do Next, page 46

What to Do Next

At the completion of this task, the SSL VPN gateway and context configurations are operational and enabled (in service), and the policy group has been defined. The SSL VPN gateway is operational for clientless remote access (HTTPS only). Proceed to the Configuring Local AAA Authentication for SSL VPN User Sessions, page 46 to see information about configuring AAA for remote-user connections.

Configuring Local AAA Authentication for SSL VPN User Sessions

The steps in this task show how to configure a local AAA database for remote-user authentication. AAA is configured in global configuration mode. In this task, the **aaa authentication** command is not configured under the SSL VPN context configuration. Omitting this command from the SSL VPN context configuration causes the SSL VPN gateway to use global authentication parameters by default.

SSL VPN gateway and context configurations are enabled and operational.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **username** *name* **secret** {**0** *user-secret* | **5** *secret-string* | *user-secret*}
5. **aaa authentication login default local**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**username** *name* **secret** {**0** *user-secret* | **5** *secret-string* | *user-secret*} **Example:** `Router(config)# username USER1 secret 0 PsW2143`	Establishes a username-based authentication system. • Entering **0** configures the password as clear text. Entering **5** encrypts the password.
Step 5	**aaa authentication login default local** **Example:** `Router(config)# aaa authentication login default local`	Configures local AAA authentication.

• What to Do Next, page 47

What to Do Next

The database that is configured for remote-user authentication on the SSL VPN gateway can be a local database, as shown in this task, or the database can be accessed through any RADIUS or TACACS+ AAA server.

It is recommended that you use a separate AAA server, such as a Cisco ACS. A separate AAA server provides a more robust security solution. It allows you to configure unique passwords for each remote user and accounting and logging for remote-user sessions. Proceed to the Configuring AAA for SSL VPN Users Using a Secure Access Control Server, page 48 to see more information.

Configuring AAA for SSL VPN Users Using a Secure Access Control Server

The steps in this task show how to configure AAA using a separate RADIUS or TACACS+ server. AAA is configured in global configuration mode. The authentication list or method is referenced in the SSL VPN context configuration with the **aaa authentication** command. The steps in this task configure AAA using a RADIUS server.

- SSL VPN gateway and context configurations are enabled and operational.
- A RADIUS or TACACS+ AAA server is operational and reachable from the SSL VPN gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server** {**radius** *group-name* | **tacacs+** *group-name*}
5. **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
6. **exit**
7. **aaa authentication login** {**default** | *list-name*} *method1* [*method2*...]
8. **radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] [**alias** {*hostname* | *ip-address*}]
9. **webvpn context** *name*
10. **aaa authentication** {**domain** *name* | **list** *name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.

	Command or Action	Purpose
Step 4	**aaa group server** {**radius** *group-name* | **tacacs+** *group-name*} **Example:** `Router(config)# aaa group server radius myServer`	Configures a RADIUS or TACACS+ server group and specifies the authentication list or method, and enters server-group configuration mode.
Step 5	**server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Router(config-sg-radius)# server 10.1.1.20 auth-port 1645 acct-port 1646`	Configures the IP address of the AAA group server.
Step 6	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits server-group configuration mode.
Step 7	**aaa authentication login** {**default** | *list-name*} *method1* [*method2*...] **Example:** `Router(config)# aaa authentication login default local group myServer`	Sets AAA login parameters.
Step 8	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] [**alias** {*hostname* | *ip-address*}] **Example:** `Router(config)# radius-server host 10.1.1.20 auth-port 1645 acct-port 1646`	Specifies a host as the group server.
Step 9	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters SSL VPN configuration mode to configure the SSL VPN context.
Step 10	**aaa authentication** {**domain** *name* | **list** *name*} **Example:** `Router(config-webvpn-context)# aaa authentication domain myServer`	Configures AAA authentication for SSL VPN sessions.

What to Do Next

Proceed to the section "Configuring RADIUS Attribute Support for SSL VPN, page 51" to see RADIUS attribute-value pair information introduced to support this feature.

Configuring RADIUS Accounting for SSL VPN User Sessions

Before configuring RADIUS accounting for SSL VPN user sessions, you should first have configured AAA-related commands (in global configuration mode) and have set the accounting list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **webvpn context** *context-name*
5. **aaa accounting list** *aaa-list*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**webvpn context** *context-name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.

	Command or Action	Purpose
Step 5	**aaa accounting list** *aaa-list* **Example:** `Router(config-webvpn-context)# aaa accounting list list1`	Enables AAA accounting when you are using RADIUS for SSL VPN sessions.

Monitoring and Maintaining RADIUS Accounting for an SSL VPN Session

To monitor and maintain your RADIUS accounting configuration, perform the following steps (the **debug** commands can be used together or individually).

SUMMARY STEPS

1. **enable**
2. **debug webvpn aaa**
3. **debug aaa accounting**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug webvpn aaa** **Example:** `Router# debug webvpn aaa`	Enables SSL VPN session monitoring for AAA.
Step 3	**debug aaa accounting** **Example:** `Router# debug aaa accounting`	Displays information on accountable events as they occur.

Configuring RADIUS Attribute Support for SSL VPN

This section lists RADIUS attribute-value (AV) pair information introduced to support SSL VPN. For information on using RADIUS AV pairs with Cisco IOS software, see the Configuring RADIUS module in the *RADIUS Configuration Guide*.

The following table shows information about SSL VPN RADIUS attribute-value pairs. All SSL VPN attributes (except for the standard IETF RADIUS attributes) start with **webvpn:** as follows:

webvpn:urllist-name=cisco webvpn:nbnslist-name=cifs webvpn:default-domain=cisco.com

Table 5 SSL VPN RADIUS Attribute-Value Pairs

Attribute	Type of Value	Values	Default
addr (Framed-IP-Address[1])	ipaddr	*IP_address*	--
addr-pool	string	*name*	--
auto-applet-download	integer	0 (disable) 1 (enable)[2]	0
banner	string		--
citrix-enabled	integer	0 (disable) 1 (enable)[3]	0
default-domain	string	--	--
dns-servers	ipaddr	*IP_address*	--
dpd-client-timeout	integer (seconds)	0 (disabled)-3600	300
dpd-gateway-timeout	integer (seconds)	0 (disabled)-3600	300
file-access	integer	0 (disable) 1 (enable) Configuring RADIUS Attribute Support for SSL VPN, page 51	0
file-browse	integer	0 (disable) 1 (enable) Configuring RADIUS Attribute Support for SSL VPN, page 51	0
file-entry	integer	0 (disable) 1 (enable) Configuring RADIUS Attribute Support for SSL VPN, page 51	0
hide-urlbar	integer	0 (disable) 1 (enable) Configuring RADIUS Attribute Support for SSL VPN, page 51	0
home-page	string	--	--

1 Standard IETF RADIUS attributes.

2 Any integer other than 0 enables this feature.

3 Any integer other than 0 enables this feature.

Attribute	Type of Value	Values	Default
idletime (Idle-Timeout Configuring RADIUS Attribute Support for SSL VPN, page 51)	integer (seconds)	0-3600	2100
ie-proxy-exception	string	*DNS_name*	--
	ipaddr	*IP_address*	--
ie-proxy-server	ipaddr	*IP_address*	--
inacl	integer	1-199, 1300-2699	--
	string	*name*	--
keep-svc-installed	integer	0 (disable) 1 (enable) Configuring RADIUS Attribute Support for SSL VPN, page 51	1
nbnslist-name	string	*name*	--
netmask (Framed-IP-Netmask Configuring RADIUS Attribute Support for SSL VPN, page 51)	ipaddr	*IP_address_mask*	--
port-forward-auto	integer	0 (disable) 1 (enable)	If this AV pair is not configured, the default is whatever was configured for the group policy. If this AV pair is configured with an integer of 1, the 1 will override a group policy value of 0.
port-forward-http-proxy	integer	0 (disable) 1 (enable)	HTTP proxy is not enabled. If this AV pair is configured with an integer of 1, the 1 will override a group policy value of 0.
port-forward-http-proxy-url	string	URL address (for example, http://example.com)	--
port-forward-name	string	*name*	--

Attribute	Type of Value	Values	Default
primary-dns	ipaddr	*IP_address*	--
rekey-interval	integer (seconds)	0-43200	21600
secondary-dns	ipaddr	*IP_address*	--
split-dns	string	--	--
split-exclude[4]	ipaddr ipaddr	*IP_address IP_address_mask*	--
	word	local-lans	--
split-include Configuring RADIUS Attribute Support for SSL VPN, page 51	ipaddr ipaddr	*IP_address IP_address_mask*	--
sso-server-name	string	*name*	--
svc-enabled[5]	integer	0 (disable) 1 (enable) Configuring RADIUS Attribute Support for SSL VPN, page 51	0
svc-ie-proxy-policy	word	none, auto, bypass-local	--
svc-required Configuring RADIUS Attribute Support for SSL VPN, page 51	integer	0 (disable) 1 (enable) Configuring RADIUS Attribute Support for SSL VPN, page 51	0
timeout (Session-Timeout Configuring RADIUS Attribute Support for SSL VPN, page 51)	integer (seconds)	1-1209600	43200
urllist-name	string	*name*	--
user-vpn-group	string	*name*	--
wins-server-primary	ipaddr	*IP_address*	--
wins-servers	ipaddr	*IP_address*	--
wins-server-secondary	ipaddr	*IP_address*	--

- What to Do Next, page 55

4 You can specify either split-include or split-exclude, but you cannot specify both options.

5 You can specify either svc-enable or svc-required, but you cannot specify both options.

What to Do Next

See the Configuring a URL List for Clientless Remote Access, page 55 for information about customizing the URL list configured in Step 10 of the section Configuring an SSL VPN Policy Group, page 43.

Configuring a URL List for Clientless Remote Access

The steps in this configuration task show how to configure a URL list. The URL list, as the name implies, is a list of HTTP URLs that are displayed on the portal page after a successful login. The URL list is configured in WebVPN context configuration and WebVPN group policy configuration modes.

SSL VPN gateway and context configurations are enabled and operational.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **url-list** *name*
5. **heading** *text-string*
6. **url-text** *name* **url-value** *url*
7. **exit**
8. **policy group** *name*
9. **url-list** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.

	Command or Action	Purpose
Step 4	**url-list** *name* **Example:** `Router(config-webvpn-context)# url-list ACCESS`	Enters WebVPN URL list configuration mode to configure the list of URLs to which a user has access on the portal page of an SSL VPN.
Step 5	**heading** *text-string* **Example:** `Router(config-webvpn-url)# heading “Quick Links”`	Configures the heading that is displayed above URLs listed on the portal page of an SSL VPN. • The heading for the URL list is entered as a text string. The heading must be entered inside of quotation marks if it contains spaces.
Step 6	**url-text** *name* **url-value** *url* **Example:** `Router(config-webvpn-url)# url-text “Human Resources” url-value example.com`	Adds an entry to a URL list.
Step 7	**exit** **Example:** `Router(config-webvpn-url)# exit`	Exits WebVPN URL list configuration mode, and enters SSL VPN context configuration mode.
Step 8	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 9	**url-list** *name* **Example:** `Router(config-webvpn-group)# url-list ACCESS`	Attaches the URL list to the policy group configuration.

- What to Do Next, page 56

What to Do Next

See the Configuring Microsoft File Shares for Clientless Remote Access, page 57 for information about configuring clientless remote access to file shares.

Configuring Microsoft File Shares for Clientless Remote Access

In clientless remote access mode, files and directories created on Microsoft Windows servers can be accessed by the remote client through the HTTPS-enabled browser. When clientless remote access is enabled, a list of file server and directory links is displayed on the portal page after login. The administrator can customize permissions on the SSL VPN gateway to provide limited read-only access for a single file or full-write access and network browsing capabilities. The following access capabilities can be configured:

- Network browse (listing of domains)
- Domain browse (listing of servers)
- Server browse (listing of shares)
- Listing files in a share
- Downloading files
- Modifying files
- Creating new directories
- Creating new files
- Deleting files

Common Internet File System Support--CIFS is the protocol that provides access to Microsoft file shares and support for common operations that allow shared files to be accessed or modified.

NetBIOS Name Service Resolution--Windows Internet Name Service (WINS) uses NetBIOS name resolution to map and establish connections between Microsoft servers. A single server must be identified by its IP address in this configuration. Up to three servers can be added to the configuration. If multiple servers are added, one server should be configured as the master browser.

Samba Support--Microsoft file shares can be accessed through the browser on a Linux system that is configured to run Samba.

- SSL VPN gateway and context configurations are enabled and operational.
- A Microsoft file server is operational and reachable from the SSL VPN gateway over TCP/IP.

Note File shares configured on Windows 2008 is not supported. Only file shares configured on Microsoft Windows 2000, Windows 2003, Windows XP, and Red Hat Linux servers are supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **nbns-list** *name*
5. **nbns-server** *ip-address* [**master**] [**timeout** *seconds*] [**retries** *number*]
6. **exit**
7. **policy group** *name*
8. **nbns-list** *name*
9. **functions** {**file-access** | **file-browse** | **file-entry** | **svc-enabled** | **svc-required**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**nbns-list** *name* **Example:** `Router(config-webvpn-context)# nbns-list SERVER_LIST`	Enters WebVPN NBNS list configuration mode to configure an NBNS server list for CIFS name resolution.
Step 5	**nbns-server** *ip-address* [**master**] [**timeout** *seconds*] [**retries** *number*] **Example:** `Router(config-webvpn-nbnslist)# nbns-server 172.16.1.1 master`	Adds a server to an NBNS server list and enters WebVPN NBNS list configuration mode. • The server specified with the *ip-address* argument can be a primary domain controller (PDC) in a Microsoft network. • When multiple NBNS servers are specified, a single server is configured as master browser. • Up to three NBNS server statements can be configured.
Step 6	**exit** **Example:** `Router(config-webvpn-nbnslist)# exit`	Exits WebVPN NBNS list configuration mode and enters WebVPN context configuration mode.
Step 7	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Enters WebVPN group policy configuration mode to configure a group policy.

	Command or Action	Purpose
Step 8	**nbns-list** *name* **Example:** `Router(config-webvpn-group)# nbns-list SERVER_LIST`	Attaches an NBNS server list to a policy group configuration.
Step 9	**functions** {**file-access** | **file-browse** | **file-entry** | **svc-enabled** | **svc-required**} **Example:** `Router(config-webvpn-group)# functions file-access`	Configures access for Microsoft file shares. • Entering the **file-access** keyword enables network file share access. File servers in the server list are listed on the SSL VPN portal page when this keyword is enabled. • Entering the **file-browse** keyword enables browse permissions for server and file shares. The file-access function must be enabled in order to also use this function. • Entering the **file-entry** keyword enables “modify” permissions for files in the shares listed on the SSL VPN portal page.

- What to Do Next, page 59

What to Do Next

See the Configuring Citrix Application Support for Clientless Remote Access, page 59 for information about configuring clientless remote access for Citrix- enabled applications.

Configuring Citrix Application Support for Clientless Remote Access

Clientless Citrix support allows the remote user to run Citrix-enabled applications through the SSL VPN as if the application were locally installed (similar to traditional thin-client computing). Citrix applications run on a MetaFrame XP server (or server farm). The SSL VPN gateway provides access to the remote user. The applications run in real time over the SSL VPN. This task shows how to enable Citrix support for policy group remote users.

The Independent Computing Architecture (ICA) client carries keystrokes and mouse clicks from the remote user to the MetaFrame XP server. ICA traffic is carried over TCP port number 1494. This port is opened when a Citrix application is accessed. If multiple application are accessed, the traffic is carried over a single TCP session.

- A Citrix MetaFrame XP server is operational and reachable from the SSL VPN gateway over TCP/IP.
- SSL VPN gateway and context configurations are enabled and operational.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**permit** | **deny**} *protocol source destination*
4. **webvpn context** *name*
5. **policy group** *name*
6. **citrix enabled**
7. **filter citrix** *extended-acl*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**permit** | **deny**} *protocol source destination* **Example:** `Router(config)# access-list 100 permit ip 192.168.1.0 0.255.255.255`	Configures the access list mechanism for filtering frames by protocol type or vendor code.
Step 4	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 5	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Enters WebVPN group policy configuration mode to configure a group policy.

	Command or Action	Purpose
Step 6	**citrix enabled** **Example:** `Router(config-webvpn-group)# citrix enabled`	Enables Citrix application support for remote users in a policy group.
Step 7	**filter citrix** *extended-acl* **Example:** `Router(config-webvpn-group)# filter citrix 100`	Configures a Citrix Thin Client filter. • An extended access list is configured to define the Thin Client filter. This filter is used to control remote user access to Citrix applications.

- What to Do Next, page 61

What to Do Next

Support for standard applications that use well-known port numbers, such as e-mail and Telnet, can be configured using the port forwarding feature. See the Configuring Application Port Forwarding, page 61 for more information.

Configuring Application Port Forwarding

Application port forwarding is configured for thin-client mode SSL VPN. Port forwarding extends the cryptographic functions of the SSL-protected browser to provide remote access to TCP and UDP-based applications that use well-known port numbers, such as POP3, SMTP, IMAP, Telnet, and SSH.

When port forwarding is enabled, the hosts file on the SSL VPN client is modified to map the application to the port number configured in the forwarding list. The application port mapping is restored to default when the user terminates the SSL VPN session.

When you are enabling port forwarding, the SSL VPN gateway will modify the hosts file on the PC of the remote user. Some software configurations and software security applications will detect this modification and prompt the remote user to choose “Yes” to permit. To permit the modification, the remote user must have local administrative privileges.

There is a known compatibility issue with the encryption type and Java. If the Java port-forwarding applet does not download properly and the configuration line **ssl encryption 3des-sha1 aes-sha1** is present, you should remove the line from the WebVPN gateway subconfiguration.

SSL VPN gateway and SSL VPN context configurations are enabled and operational.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **port-forward** *name*
5. **local-port** *number* **remote-server** *name* **remote-port** *number* **description** *text-string*
6. **exit**
7. **policy group** *name*
8. **port-forward** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**port-forward** *name* **Example:** `Router(config-webvpn-context)# port-forward EMAIL`	Enters WebVPN port-forward list configuration mode to configure a port forwarding list.
Step 5	**local-port** *number* **remote-server** *name* **remote-port** *number* **description** *text-string* **Example:** `Router(config-webvpn-port-fwd)# local-port 30016 remote-server example.com remote-port 110 description POP3`	Remaps (forwards) an application port number in a port forwarding list. • The remote port number is the well-known port to which the application listens. The local port number is the entry configured in the port forwarding list. A local port number can be configured only once in a given port forwarding list.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-webvpn-port-fwd)# exit`	Exits WebVPN port-forward list configuration mode, and enters WebVPN context configuration mode.
Step 7	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 8	**port-forward** *name* **Example:** `Router(config-webvpn-group)# port-forward EMAIL`	Attaches a port forwarding list to a policy group configuration.

Configuring the SSL VPN Gateway to Distribute CSD and Cisco AnyConnect VPN Client Package Files

The SSL VPN gateway is preconfigured to distribute Cisco Secure Desktop (CSD) or Cisco AnyConnect VPN Client software package files to remote users. The files are distributed only when CSD or Cisco AnyConnect VPN Client support is needed. The administrator performs the following tasks to prepare the gateway:

- The current software package is downloaded from www.cisco.com.
- The package file is copied to a local file system.
- The package file is installed for distribution by configuring the **webvpn install** command.

The remote user must have administrative privileges, and the JRE for Windows version 1.4 or later must be installed before the CSD client package can be installed.

For Cisco AnyConnect VPN Client software installation, the remote user must have either the Java Runtime Environment for Windows (version 1.4 or later), or the browser must support or be configured to permit Active X controls.

CSD and Cisco AnyConnect VPN Client software packages should be installed for distribution on the SSL VPN gateway. Download the latest version that supports your device and the image you are using (consult a compatibility matrix for your particular setup).

The CSD software package can be downloaded at the following URL:

- http://www.cisco.com/cgi-bin/tablebuild.pl/securedesktop

The Cisco AnyConnect VPN Client software package can be downloaded at the following URL:

- http://www.cisco.com/cgi-bin/tablebuild.pl/anyconnect

The Cisco SSL VPN Client software package can be downloaded at the following URL:

- http://www.cisco.com/cgi-bin/tablebuild.pl/sslvpnclient

You will be prompted to enter your login name and password to download these files from cisco.com.

- SSL VPN gateway and context configurations are enabled and operational.
- Software installation packages are copied to a local files system, such as flash memory.

Note Effective with Cisco IOS Release 12.4(20)T, multiple packages can be downloaded to a gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn install** [**csd** *location-name* | **svc** *location-name* [**sequence** *sequence-number*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**webvpn install** [**csd** *location-name* | **svc** *location-name* [**sequence** *sequence-number*]]
	
	Example:
	`Router(config)# webvpn install svc flash:/webvpn/svc.pkg`
	or
	`Router(config)# webvpn install svc vpn-2_i386-Release-2.0.0077-k9.pkg sequence 6`
		Installs a CSD or Cisco AnyConnect VPN Client package file to an SSL VPN gateway for distribution to remote users.
		• The CSD and Cisco AnyConnect VPN Client software packages are pushed to remote users as access is needed.
		• The **sequence** keyword and *sequence-number* argument are used to install multiple packages to a gateway.

- Examples, page 64
- What to Do Next, page 65

Examples

The following example, starting in global configuration mode, installs the Cisco AnyConnect VPN Client package to an SSL VPN gateway:

```
Router(config)# webvpn install svc flash:/webvpn/svc.pkg
```

```
SSL VPN Package SSL-VPN-Client : installed successfully
```

The following example, starting in global configuration mode, installs the CSD package to an SSL VPN gateway:

```
Router(config)# webvpn install csd flash:/securedesktop_10_1_0_9.pkg
SSL VPN Package Cisco-Secure-Desktop : installed successfully
```

The following example shows that Package B is being installed to an SSL VPN gateway:

```
Router(config)# webvpn install svc flash:/webvpn/packageB sequence 2
```

What to Do Next

Support for CSD and Cisco AnyConnect VPN Client can be enabled for remote users after the gateway has been prepared to distribute CSD or Cisco AnyConnect VPN Client software.

Configuring Cisco Secure Desktop Support

CSD provides a session-based interface where sensitive data can be shared for the duration of an SSL VPN session. All session information is encrypted. All traces of the session data are removed from the remote client when the session is terminated, even if the connection is terminated abruptly. CSD support for remote clients is enabled in this task.

The remote user (PC or device) must have administrative privileges, and the JRE for Windows version 1.4 or later must be installed before the CSD client packages can be installed.

- SSL VPN gateway and context configurations are enabled and operational.
- The CSD software package is installed for distribution on the SSL VPN gateway.

See the Configuring the SSL VPN Gateway to Distribute CSD and Cisco AnyConnect VPN Client Package Files, page 63 section if you have not already prepared the SSL VPN gateway to distribute CSD software.

Note Only Microsoft Windows 2000, Windows XP, Windows Vista, Apple-Mac, and Linux are supported on the remote client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **csd enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**csd enable** **Example:** `Router(config-webvpn-context)# csd enable`	Enables CSD support for SSL VPN sessions.

• What to Do Next, page 66

What to Do Next

Upon completion of this task, the SSL VPN gateway has been configured to provide clientless and thin-client support for remote users. The SSL VPN feature also has the capability to provide full VPN access (similar to IPsec). Proceed to the Configuring Cisco AnyConnect VPN Client Full Tunnel Support, page 66 to see more information.

Configuring Cisco AnyConnect VPN Client Full Tunnel Support

The Cisco AnyConnect VPN Client is an application that allows a remote user to establish a full VPN connection similar to the type of connection that is established with an IPsec VPN. Cisco AnyConnect VPN Client software is pushed (downloaded) and installed automatically on the PC of the remote user. The Cisco AnyConnect VPN Client uses SSL to provide the security of an IPsec VPN without the complexity required to install IPsec in your network and on remote devices. The following tasks are completed in this configuration:

- An access list is applied to the tunnel to restrict VPN access.
- Cisco AnyConnect VPN Client tunnel support is enabled.
- An address pool is configured for assignment to remote clients.

- The default domain is configured.
- DNS is configured for Cisco AnyConnect VPN Client tunnel clients.
- Dead peer timers are configured for the SSL VPN gateway and remote users.
- The login home page is configured.
- The Cisco AnyConnect VPN Client software package is configured to remain installed on the remote client.
- Tunnel key refresh parameters are defined.

- SSL VPN gateway and context configurations are enabled and operational.
- The Cisco AnyConnect VPN Client software package is installed for distribution on the SSL VPN gateway.
- The remote client has administrative privileges. Administrative privileges are required to download the SSL VPN software client.

See the Configuring the SSL VPN Gateway to Distribute CSD and Cisco AnyConnect VPN Client Package Files, page 63 section if you have not already prepared the SSL VPN gateway to distribute SSL VPN software.

Note Only Microsoft Windows 2000, Windows XP, Windows Vista, Apple-Mac, and Linux are supported on the remote client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **policy group** *name*
5. **filter tunnel** *extended-acl*
6. **functions** {**file-access** | **file-browse** | **file-entry** | **svc-enabled** | **svc-required**}
7. **svc address-pool** *name*
8. **svc default-domain** *name*
9. **svc dns-server** {**primary** | **secondary**} *ip-address*
10. **svc dpd-interval** {**client** | **gateway**} *seconds*
11. **svc keepalive** *seconds*
12. **svc homepage** *string*
13. **svc keep-client-installed**
14. **svc rekey** {**method** {**new-tunnel** | **ssl**} | **time** *seconds*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 5	**filter tunnel** *extended-acl* **Example:** `Router(config-webvpn-group)# filter tunnel 101`	Configures an SSL VPN tunnel access filter. • The tunnel access filter is used to control network and application level access. The tunnel filter is also defined in an extended access list.
Step 6	**functions** {**file-access** | **file-browse** | **file-entry** | **svc-enabled** | **svc-required**} **Example:** `Router(config-webvpn-group)# functions svc-enabled`	Configures Cisco AnyConnect VPN Client tunnel mode support. • Entering the **svc-enabled** keyword enables tunnel support for the remote user. If the Cisco AnyConnect VPN Client software package fails to install, the remote user can continue to use clientless mode or thin-client mode. • Entering the **svc-required** keyword enables only tunnel support for the remote user. If the Cisco AnyConnect VPN Client software package fails to install (on the PC of the remote user), the other access modes cannot be used.

	Command or Action	Purpose
Step 7	**svc address-pool** *name* **Example:** `Router(config-webvpn-group)# svc address-pool ADDRESSES`	Configures a pool of IP addresses to assign to remote users in a policy group. • The address pool is first defined with the **ip local pool** command in global configuration mode. • If you are configuring an address pool for a network that is not direc tly connected, an address from the pool must be configured on a locally loopback interface. See the third example at the end of this section.
Step 8	**svc default-domain** *name* **Example:** `Router(config-webvpn-group)# svc default-domain cisco.com`	Configures the default domain for a policy group.
Step 9	**svc dns-server** {**primary** | **secondary**} *ip-address* **Example:** `Router(config-webvpn-group)# svc dns-server primary 192.168.3.1`	Configures DNS servers for policy group remote users.
Step 10	**svc dpd-interval** {**client** | **gateway**} *seconds* **Example:** `Router(config-webvpn-group)# svc dpd-interval gateway 30`	Configures the dead peer detection (DPD) timer value for the gateway or client. • The DPD timer is reset every time a packet is received over the SSL VPN tunnel from the gateway or remote user.
Step 11	**svc keepalive** *seconds* **Example:** `Router(config-webvpn-group)# svc keepalive 300`	(Optional) Enables the SVC to send keepalive messages by default with a frequency of 30 seconds. • Use this command to adjust the frequency of keepalive messages to ensure that an SVC connection through a proxy, Cisco IOS firewall, or NAT device remains active, even if the device limits the time that the connection can be idle. Adjusting the frequency also ensures that the SVC does not disconnect and reconnect when the remote user is not actively running a socket-based application, such as Microsoft Outlook or Microsoft Internet Explorer. • If the **svc keepalive** command is configured with a value of **0** seconds, then the keepalive function is disabled.

Command or Action	Purpose
Step 12 **svc homepage** *string* **Example:** `Router(config-webvpn-group)# svc homepage www.cisco.com`	Configures the URL of the web page that is displayed upon successful user login. • The *string* argument is entered as an HTTP URL. The URL can be up to 255 characters in length.
Step 13 **svc keep-client-installed** **Example:** `Router(config-webvpn-group)# svc keep-client-installed`	Configures the remote user to keep Cisco AnyConnect VPN Client software installed when the SSL VPN connection is not enabled.
Step 14 **svc rekey** {**method** {**new-tunnel** | **ssl**} | **time** *seconds*} **Example:** `Router(config-webvpn-group)# svc rekey method new-tunnel`	Configures the time and method that a tunnel key is refreshed for policy group remote users. • The tunnel key is refreshed by renegotiating the SSL connection or initiating a new tunnel connection. • The time interval between tunnel refresh cycles is configured in seconds.

- Examples, page 70
- What to Do Next, page 71

Examples

Tunnel Filter Configuration

The following example, starting in global configuration mode, configures a deny access filter for any host from the 172.16.2/24 network:

```
Router(config)# access-list 101 deny ip 172.16.2.0 0.0.0.255 any
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# filter tunnel 101
Router(config-webvpn-group)# end
```

Address Pool (Directly Connected Network) Configuration

The following example, starting in global configuration mode, configures the 192.168.1/24 network as an address pool:

```
Router(config)# ip local pool ADDRESSES 192.168.1.1 192.168.1.254
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# svc address-pool ADDRESSES
Router(config-webvpn-group)# end
```

Address Pool (Nondirectly Connected Network) Configuration

The following example, starting in global configuration mode, configures the 172.16.1/24 network as an address pool. Because the network is not directly connected, a local loopback interface is configured.

```
Router(config)# interface loopback 0
Router(config-int)# ip address 172.16.1.126 255.255.255.0
Router(config-int)# no shutdown
Router(config-int)# exit
Router(config)# ip local pool ADDRESSES 172.16.1.1 172.16.1.254
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# svc address-pool ADDRESSES
Router(config-webvpn-group)# end
```

Full Tunnel Configuration

The following example, starting in global configuration mode, configures full Cisco AnyConnect VPN Client tunnel support on an SSL VPN gateway:

```
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# functions svc-enabled
Router(config-webvpn-group)# functions svc-required
Router(config-webvpn-group)# svc default-domain cisco.com
Router(config-webvpn-group)# svc dns-server primary 192.168.3.1
Router(config-webvpn-group)# svc dns-server secondary 192.168.4.1
Router(config-webvpn-group)# svc dpd-interval gateway 30
Router(config-webvpn-group)# svc dpd-interval client 300
Router(config-webvpn-group)# svc homepage www.cisco.com
Router(config-webvpn-group)# svc keep-client-installed
Router(config-webvpn-group)# svc rekey method new-tunnel
Router(config-webvpn-group)# svc rekey time 3600
Router(config-webvpn-group)# end
```

What to Do Next

Proceed to the Configuring Advanced SSL VPN Tunnel Features, page 71 to see advanced Cisco AnyConnect VPN Client tunnel configuration information.

Configuring Advanced SSL VPN Tunnel Features

This section describes advanced Cisco AnyConnect VPN Client tunnel configurations. The following configuration steps are completed in this task:

- Split tunnel support and split DNS resolution are enabled on the SSL VPN gateway.
- SSL VPN gateway support for Microsoft Internet Explorer proxy settings is configured.
- WINS resolution is configured for Cisco AnyConnect VPN Client tunnel clients.

Microsoft Internet Explorer Proxy Configuration--The SSL VPN gateway can be configured to pass or bypass Microsoft Internet Explorer (MSIE) proxy settings. Only HTTP proxy settings are supported by the SSL VPN gateway. MSIE proxy settings have no effect on any other supported browser.

Split Tunneling--Split tunnel support allows you to configure a policy that permits specific traffic to be carried outside of the Cisco AnyConnect VPN Client tunnel. Traffic is either included (resolved in tunnel) or excluded (resolved through the Internet service provider [ISP] or WAN connection). Tunnel resolution configuration is mutually exclusive. An IP address cannot be both included and excluded at the same time. Entering the **local-lans** keyword permits the remote user to access resources on a local LAN, such as network printer.

- SSL VPN gateway and context configurations are enabled and operational.

- The Cisco AnyConnect VPN Client software package is installed for distribution on the SSL VPN gateway.

Note Only Microsoft Windows 2000, Windows XP, Windows Vista, Apple-Mac, and Linux are supported on the remote client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **policy group** *name*
5. **svc split exclude** {{*ip-address mask* | **local-lans**} | **include** *ip-address mask*}
6. **svc split dns** *name*
7. **svc msie-proxy** {**exception** *host* | **option** {**auto** | **bypass-local** | **none**}}
8. **svc msie-proxy server** *host*
9. **svc wins-server** {**primary** | **secondary**} *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Enters WebVPN group policy configuration mode to configure a group policy.

Command or Action	Purpose
Step 5 **svc split exclude** {{*ip-address mask* | **local-lans**} | **include** *ip-address mask*} **Example:** `Router(config-webvpn-group)# svc split exclude 192.168.1.1 0.0.0.255`	Configures split tunneling for policy group remote users. • Split tunneling is configured to include or exclude traffic in the Cisco AnyConnect VPN Client tunnel. Traffic that is included is sent over the SSL VPN tunnel. Excluded traffic is resolved outside of the tunnel. • Exclude and include statements are configured with IP address/wildcard mask pairs.
Step 6 **svc split dns** *name* **Example:** `Router(config-webvpn-group)# svc split dns www.examplecompany.com`	Configures the SSL VPN gateway to resolve the specified fully qualified DNS names through the Cisco AnyConnect VPN Client tunnel. • A default domain was configured in the previous task with the **svc default-domain** command. DNS names configured with the **svc split dns** command are configured in addition. • Up to 10 split DNS statements can be configured.
Step 7 **svc msie-proxy** {**exception** *host* | **option** {**auto** | **bypass-local** | **none**}} **Example:** `Router(config-webvpn-group)# svc msie-proxy option auto`	Configures MSIE browser proxy settings for policy group remote users. • Entering the **option auto** keywords configures the browser of the remote user to autodetect proxy settings. • Entering the **option bypass-local** keywords configures local addresses to bypass the proxy. • Entering the **option none** keywords configures the browser on the remote client to not use a proxy.
Step 8 **svc msie-proxy server** *host* **Example:** `Router(config-webvpn-group)# svc msie-proxy server 10.10.10.1:80`	Specifies an MSIE proxy server for policy group remote users. • The proxy server is specified by entering an IP address or a fully qualified domain name.
Step 9 **svc wins-server** {**primary** | **secondary**} *ip-address* **Example:** `Router(config-webvpn-group)# svc wins-server primary 172.31.1.1`	Configures WINS servers for policy group remote users.

• Examples, page 73

Examples

Split DNS Configuration

The following example, starting in global configuration mode, configures the following DNS names to be resolved in the Cisco AnyConnect VPN Client tunnel:

```
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# svc split dns www.example.com
Router(config-webvpn-group)# svc split dns myexample.com
```

Including and Excluding IP Prefixes

The following example configures a list of IP addresses to be resolved over the tunnel (included) and a list to be resolved outside of the tunnel (excluded):

```
Router(config-webvpn-group)# svc split exclude 192.168.1.0 255.255.255.0
Router(config-webvpn-group)# svc split include 172.16.1.0 255.255.255.0
```

MSIE Proxy Configuration

The following example configures MSIE proxy settings:

```
Router(config-webvpn-group)# svc msie-proxy option auto
Router(config-webvpn-group)# svc msie-proxy exception www.example.com
Router(config-webvpn-group)# svc msie-proxy exception 10.20.20.1
Router(config-webvpn-group)# svc msie-proxy server 10.10.10.1:80
```

WINS Server Configuration

The following example configures primary and secondary WINS servers for the policy group:

```
Router(config-webvpn-group)# svc wins-server primary 172.31.1.1
Router(config-webvpn-group)# svc wins-server secondary 172.31.2.1
Router(config-webvpn-group)# svc wins-server secondary 172.31.3.1
Router(config-webvpn-group)# end
```

Configuring VRF Virtualization

VRF Virtualization allows you to associate a traditional VRF with an SSL VPN context configuration. This feature allows you to apply different configurations and reuse address space for different groups of users in your organization.

- A VRF has been configured in global configuration mode.
- SSL VPN gateway and context configurations are enabled and operational.
- A policy group has been configured and associated with the WebVPN context.

Note Only a single VRF can be configured for each SSL VPN context configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **vrf-name** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.

Step 4	**vrf-name** *name*
	
	Example:
	`Router(config-webvpn-context)# vrf-name vrf1`
		Associates a VRF with an SSL VPN context.
		Note When you configure the VRF Virtualization feature in Cisco IOS Release 12.4(24)T1 and later releases, the following message is displayed:
		`% IP VRF vrf1 configuration applied.`
		`% But please use Virtual-Template to configure VRF.`
		See the Configuring SSLVPN DVTI Support, page 104 section for the procedure to configure IP features using virtual template.

Configuring ACL Rules

The ACL rules can be overridden for an individual user when the user logs in to the gateway (using AAA policy attributes). If a user session has no ACL attribute configured, all application requests from that user session are permitted by default.

Before configuring the ACL rules, you must have first configured the time range using the **time-range** command (this prerequisite is in addition to optionally configuring the time range, in the task table, as part of the **permit** or **deny** entries).

Note There is no limitation on the maximum number of filtering rules that can be configured for each ACL entry, but keeping the number below 50 should have no significant impact on router performance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **acl** *acl-name*
5. Do one of the following:
   - **permit** [**url** [**any** | *url-string*]] [**ip** | **tcp** | **udp** | **http** | **https** | **cifs**] [**any** | *source-ip source-mask*] [**any** | *destination-ip destination-mask*] [**time-range** *time-range-name*] [**syslog**]
   - **deny** [**url** [**any** | *url-string*]] [**ip** | **tcp** | **udp** | **http** | **https** | **cifs**] [**any** | *source-ip source-mask*] [**any** | *destination-ip destination-mask*] [**time-range** *time-range-name*] [**syslog**]
6. **add** *position acl-entry*
7. **error-url** *access-deny-page-url*
8. **error-msg** *message-string*
9. **list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**acl** *acl-name* **Example:** `Router(config-webvpn-context)# acl acl1`	Defines the ACL and enters WebVPN ACL configuration mode.

Command or Action	Purpose
Step 5 Do one of the following: • **permit** [**url** [**any** | *url-string*]] [**ip** | **tcp** | **udp** | **http** | **https** | **cifs**] [**any** | *source-ip source-mask*] [**any** | *destination-ip destination-mask*] [**time-range** *time-range-name*] [**syslog**] • **deny** [**url** [**any** | *url-string*]] [**ip** | **tcp** | **udp** | **http** | **https** | **cifs**] [**any** | *source-ip source-mask*] [**any** | *destination-ip destination-mask*] [**time-range** *time-range-name*] [**syslog**] **Example:** `Router(config-webvpn-acl)# permit url any`	Sets conditions in a named SSL VPN access list that will permit or deny packets.
Step 6 **add** *position acl-entry* **Example:** `Router(config-webvpn-acl)# add 3 permit url any`	(Optional) Adds an ACL entry at a specified position.
Step 7 **error-url** *access-deny-page-url* **Example:** `Router(config-webvpn-acl)# error-url “http://www.example.com”`	(Optional) Defines a URL as an ACL violation page. • If the **error-url** command is configured, the user is redirected to a predefined URL for every request that is not allowed. If the **error-url** command is not configured, the user gets a standard, gateway-generated error page.
Step 8 **error-msg** *message-string* **Example:** `Router(config-webvpn-acl)# error-msg “If you have any questions, please contact <a href+mailto:employee1@example.com>Employee1</a>.”`	(Optional) Displays a specific error message when a user logs in and his or her request is denied.
Step 9 **list** **Example:** `Router(config-webvpn-acl)# list`	(Optional) Lists the currently configured ACL entries sequentially and assigns a position number.

Associating an ACL Attribute with a Policy Group

Note Associating an ACL attribute for an individual user must be performed as part of a AAA operation.

- The ACL rules can be overridden for an individual user when the user logs in to the gateway (using AAA policy attributes).
- If a user session has no ACL attribute configured, all application requests from that user session are permitted by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **policy group** *name*
5. **exit**
6. **acl** *acl-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Configures the SSL VPN context and enters WebVPN context configuration mode.
Step 4	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group group1`	Defines a policy that can be applied to the user and enters WebVPN policy group configuration mode.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-webvpn-group)# exit`	Exits WebVPN policy group configuration mode.
Step 6	**acl** *acl-name* **Example:** `Router(config-webvpn-context)# acl acl1`	Defines the ACL and enters WebVPN ACL configuration mode.

- Monitoring and Maintaining ACLs, page 79

Monitoring and Maintaining ACLs

SUMMARY STEPS

1. **enable**
2. **debug webvpn acl**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug webvpn acl** **Example:** `Router# debug webvpn acl`	Displays information about ACLs.

Configuring SSO Netegrity Cookie Support for a Virtual Context

To configure SSO Netegrity cookie support for a virtual context, perform the following steps.

Note A Cisco plug-in must first be installed on a Netegrity server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **sso-server** *name*
5. **web-agent-url** *url*
6. **secret-key** *key-name*
7. **max-retry-attempts** *number-of-retries*
8. **request-timeout** *number-of-seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**sso-server** *name* **Example:** `Router(config-webvpn-context)# sso-server “test-sso-server”`	Creates an SSO server name under an SSL VPN context and enters WebVPN SSSO server configuration mode.
Step 5	**web-agent-url** *url* **Example:** `Router(config-webvpn-sso-server)# web-agent-url http://www.example.comwebvpn/`	Configures the Netegrity agent URL to which SSO authentication requests will be dispatched.

	Command or Action	Purpose
Step 6	**secret-key** *key-name* **Example:** `Router(config-webvpn-sso-server)# secret-key “12345”`	Configures the policy server secret key that is used to secure authentication requests.
Step 7	**max-retry-attempts** *number-of-retries* **Example:** `Router(config-webvpn-sso-server)# max-retry-attempts 3`	Sets the maximum number of retries before SSO authentication fails.
Step 8	**request-timeout** *number-of-seconds* **Example:** `Router(config-webvpn-sso-server)# request-timeout 15`	Sets the number of seconds before an authentication request times out.

Associating an SSO Server with a Policy Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **policy group** *name*
5. **sso-server** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Configures the SSL VPN context and enters WebVPN context configuration mode.
Step 4	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Configures a group policy and enters WebVPN group policy configuration mode.
Step 5	**sso-server** *name* **Example:** `Router(config-group-webvpn)# sso-server “test-sso-server”`	Attaches an SSO server to a policy group.

Configuring URL Obfuscation (Masking)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **policy group** *name*
5. **mask-urls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Configures the SSL VPN context and enters WebVPN context configuration mode.
Step 4	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Configures a group policy and enters group policy configuration mode.
Step 5	**mask-urls** **Example:** `Router(config-webvpn-group)# mask-urls`	Obfuscates, or masks, sensitive portions of an enterprise URL, such as IP addresses, hostnames, or port numbers.

Adding a CIFS Server URL List to an SSL VPN Context and Attaching It to a Policy Group

Before adding a CIFS server URL list to an SSL VPN context, you must have already set up the Web VPN context using the **webvpn context** command, and you must be in WebVPN context configuration mode.

SUMMARY STEPS

1. **cifs-url-list** *name*
2. **heading** *text-string*
3. **url-text** *name*
4. **exit**
5. **policy group** *name*
6. **cifs-url-list** *name*
7. **exit**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**cifs-url-list** *name* **Example:** `Router(config-webvpn-context)# cifs-url-list c1`	Enters WebVPN URL list configuration mode to configure a list of CIFS server URLs to which a user has access on the portal page of an SSL VPN.

	Command or Action	Purpose
Step 2	**heading** *text-string* **Example:** `Router(config-webvpn-url)# heading “cifs-url”`	Configures the heading that is displayed above URLs listed on the portal page of an SSL VPN.
Step 3	**url-text** *name* **Example:** `Router(config-webvpn-url)# url-text “SSLVPN-SERVER2” url-value “\\SLVPN-SERVER2”`	Adds an entry to a URL list. • More than one entry can be added by reentering the **url-text** command for each subsequent entry.
Step 4	**exit** **Example:** `Router(config-webvpn-url)# exit`	Exits WebVPN URL list configuration mode and returns to WebVPN context configuration mode.
Step 5	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group ONE`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 6	**cifs-url-list** *name* **Example:** `Router(config-webvpn-group)# cifs-url-list “c1”`	Attaches a URL list to a policy group.
Step 7	**exit** **Example:** `Router(config-webvpn-group)# exit`	Exits WebVPN group policy configuration mode.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring User-Level Bookmarks

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **user-profile location flash:** *directory*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Configures the SSL VPN context and enters WebVPN context configuration mode.
Step 4	**user-profile location flash:** *directory* **Example:** `Router(config-webvpn-context)# user-profile location flash:webvpn/sslvpn/vpn_context/`	Stores bookmarks on a directory.

Configuring FVRF

To configure FVRF so that the SSL VPN gateway is fully integrated into an MPLS network, perform the following steps.

As the following configuration task shows, IP VRF must be configured before the FVRF can be associated with the SSL VPN gateway. For more information about configuring IP VRF, see the Configuring IP VRF (**ip vrf** command) in the Additional References, page 141 section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **exit**
5. **webvpn gateway** *name*
6. **vrfname** *name*
7. **exit**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf_1`	Defines a VPN VRF instance and enters VRF configuration mode. **Note** The *vrf-name* argument specified here must be the same as the *name* argument in Step 6.
Step 4	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode.
Step 5	**webvpn gateway** *name* **Example:** `Router(config)# webvpn gateway mygateway`	Enters WebVPN gateway configuration mode to configure an SSL VPN gateway.
Step 6	**vrfname** *name* **Example:** `Router(config-webvpn-gateway)# vrfname vrf_1`	Associates a VPN FVRF with an SSL VPN gateway. **Note** The value for the *name* argument here must be the same as the value for the *vrf-name* argument in Step 3.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-webvpn-gateway)# exit`	Exits WebVPN gateway configuration mode.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Disabling Full-Tunnel Cisco Express Forwarding

Note The **no webvpn cef** command disables all Web VPN Cisco Express Forwarding support, not just full-tunnel Cisco Express Forwarding support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no webvpn cef**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no webvpn cef** **Example:** `Router(config)# no webvpn cef`	Disables full-tunnel Cisco Express Forwarding support. **Note** The **webvpn cef** command is enabled by default.

Configuring Automatic Authentication and Authorization

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **aaa authentication auto**
5. **aaa authorization list** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**aaa authentication auto** **Example:** `Router(config-webvpn-context)# aaa authentication auto`	Allows automatic authentication for users. • Users provide their usernames and passwords via the gateway page URL and do not have to again enter their usernames and passwords from the login page.
Step 5	**aaa authorization list** *name* **Example:** `Router(config-webvpn-context)# aaa authorization list 11`	Allows user attributes to get “pushed” during authentication. • *name* --Name of the list to be automatically authorized.

Configuring SSL VPN Client-Side Certificate-Based Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn import svc profile** *profile-name device-name*
4. **webvpn context** *context-name*
5. **authentication certificate aaa**
6. **username-prefill**
7. **ca trustpoint** *trustpoint-name*
8. **match-certificate** *certificate-name*
9. **policy group** *policy-name*
10. **svc profile** *profile-name*
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn import svc profile** *profile-name device-name* **Example:** `Router(config)# webvpn import svc profile profile1 flash:AnyconnectProfile.tmpl`	Imports an AnyConnect profile.
Step 4	**webvpn context** *context-name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.

	Command or Action	Purpose
Step 5	**authentication certificate aaa** **Example:** `Router(config-webvpn-context)# authentication certificate aaa`	Enables certificate-based AAA authentication.
Step 6	**username-prefill** **Example:** `Router(config-webvpn-context)# username-prefill`	Enables trustpoint configuration to prefill the username field from an authentication certificate.
Step 7	**ca trustpoint** *trustpoint-name* **Example:** `Router(config-webvpn-context)# ca trustpoint trustpoint1`	Enables the trustpoint to authenticate users using the specified trust point name.
Step 8	**match-certificate** *certificate-name* **Example:** `Router(config-webvpn-context)# match-certificate certificate1`	Enables certificate map matching.
Step 9	**policy group** *policy-name* **Example:** `Router(config-webvpn-context)# policy group policy3`	Enters WebVPN group policy configuration mode to configure a WebVPN group policy.
Step 10	**svc profile** *profile-name* **Example:** `Router(config-webvpn-group)# svc profile profile1`	Enables a WebVPN group policy with an AnyConnect profile.
Step 11	**exit** **Example:** `Router(config-webvpn-group)# exit`	Exits WebVPN group policy mode.

Configuring a URL Rewrite Splitter

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **url rewrite**
5. **host** *host-name*
6. **ip** *ip-address*
7. **unmatched-action** [**direct-access** | **redirect**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**url rewrite** **Example:** `Router(config-webvpn-context)# url rewrite`	Allows you to mangle selective URL requests and enters URL rewrite mode. **Note** You must enter either the **host** command (Step 5) or the **ip** command (Step 6).
Step 5	**host** *host-name* **Example:** `Router(config-webvpn-url-rewrite)# host www.examplecompany.com`	Hostname of the site to be mangled. **Note** You must enter either the **host** command (Step 5) or the **ip** command (Step 6).

	Command or Action	Purpose
Step 6	**ip** *ip-address* **Example:** `Router(config-webvpn-url-rewrite)# ip 10.1.1.0 255.255.0.0`	IP address of the site to be mangled. **Note** You must enter either the **host** command (Step 5) or the **ip** command (Step 6).
Step 7	**unmatched-action** [**direct-access** | **redirect**] **Example:** `Router(config-webvpn-url-rewrite)# unmatched-action direct-access`	(Optional) Defines the action for the request to the public website. • **direct-access** --Provides the user with direct access to the URL. In addition, the user receives an information page stating that he or she can access the URL directly. • **redirect** --Provides the user with direct access to the URL, but the user does not receive the information page.

Configuring a Backend HTTP Proxy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **policy group** *name*
5. **http proxy-server** {*ip-address* | *dns-name*} **port** *port-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.

	Command or Action	Purpose
Step 4	**policy group** *name* **Example:** `Router(config-webvpn-context)# policy group g1`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 5	**http proxy-server** {*ip-address* | *dns-name*} **port** *port-number* **Example:** `Router(config-webvpn-context)# http proxy-server 10.1.1.1 port 2034`	Allows user requests to go through a backend HTTP proxy. • *ip-address* --IP address of the proxy server. • *dns-name* --DNS of the proxy server. • **port** *port-number* --Proxy port number.

Configuring Stateless High Availability with HSRP for SSL VPN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **standby** *number* **ip** *ip-address*
5. **standby** *number* **name** *standby-name*
6. **exit**
7. **webvpn gateway** *name*
8. **ip address** *number* **port** *port-number* **standby** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type slot/port* **Example:** `Router(config)# interface gateway 0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**standby** *number* **ip** *ip-address* **Example:** `Router(config-if)# standby 0 ip 10.1.1.1`	Configures a standby IP address.
Step 5	**standby** *number* **name** *standby-name* **Example:** `Router(config-if)# standby 0 name SSLVPN`	Configures a standby name.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**webvpn gateway** *name* **Example:** `Router(config)# webvpn gateway Gateway1`	Enters WebVPN gateway configuration mode to configure an SSL VPN gateway.
Step 8	**ip address** *number* **port** *port-number* **standby** *name* **Example:** `Router(config)# ip address 10.1.1.1 port 443 standby SSLVPN`	Configures a standby IP address as the proxy IP address on an SSL VPN gateway. **Note** The IP address configured here must be the same as the IP address that was configured as the standby IP address (**standby** *number* **ip** *ip-address*).

Configuring Internationalization

Generating the Template Browser Attribute File

SUMMARY STEPS

1. **enable**
2. **webvpn create template browser-attribute** *device*:
3. Copy the browser attribute file to another device on which you can edit the language being configured.
4. Copy the edited file back to the storage device.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**webvpn create template browser-attribute** *device*: **Example:** `Router# webvpn create template browser-attribute flash:`	Generates the browser attribute template XML file (battr_tpl.xml).
Step 3	Copy the browser attribute file to another device on which you can edit the language being configured.	For an example of how to copy the file to your PC, see the Example: Copying the Browser Attribute File to Another PC for Editing, page 128.
Step 4	Copy the edited file back to the storage device.	For an example of how to copy the edited file to a storage device, see the Example: Copying the Edited File to flash, page 128.

- What to Do Next, page 95

What to Do Next

Proceed to the Importing the Browser Attribute File, page 95.

Importing the Browser Attribute File

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **browser-attribute import** *device*:*file-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**browser-attribute import** *device*:*file-name* **Example:** `Router(config-webvpn-context)# browser-attribute import flash:battr_tpl.xml`	Imports the edited browser attribute file from the storage device.

• What to Do Next, page 96

What to Do Next

Proceed to the Verifying That the Browser Attribute File Was Imported Correctly, page 96.

Verifying That the Browser Attribute File Was Imported Correctly

SUMMARY STEPS

1. **enable**
2. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show running-config** **Example:** `Router# show running-config`	Verifies that the browser attribute file was imported correctly.

- What to Do Next, page 97

What to Do Next

Proceed to the Creating the Language File, page 97.

Creating the Language File

SUMMARY STEPS

1. **enable**
2. **webvpn create template language** *device*:
3. Copy the language lang.js file to a PC for editing.
4. Copy the edited language lang.js file to the storage device.
5. **webvpn create template language** {**japanese** | **customize** *language-name device*:*file*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**webvpn create template language** *device*: **Example:** `Router# webvpn create template language flash:`	Creates the language template file lang.js. **Note** A lang.js file does not have to be created if the language is English or Japanese.
Step 3	Copy the language lang.js file to a PC for editing.	For an example of how to copy the language file to another PC, see the Example: Copying the Language File to Another PC for Editing, page 129.
Step 4	Copy the edited language lang.js file to the storage device.	For an example of how to copy the edited file to the storage device, see the Example: Copying the Edited Language File to the Storage Device, page 129.

	Command or Action	Purpose
Step 5	**webvpn create template language** {**japanese** | **customize** *language-name device***:***file*} **Example:** `Router# webvpn create template language japanese`	Creates templates for multilanguage support for messages initiated by the headend in an SSL VPN.

- What to Do Next, page 98

What to Do Next

Proceed to the Importing the Language File, page 98.

Importing the Language File

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **language** {**japanese** | **customize** *language-name device***:***file*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.

	Command or Action	Purpose
Step 4	**language** {**japanese** | **customize** *language-name device*:*file*} **Example:** `Router(config-webvpn-context)# language Japanese`	Imports the language file.

- What to Do Next, page 99

What to Do Next

Proceed to the Verifying That the Language File Was Imported Correctly, page 99.

Verifying That the Language File Was Imported Correctly

SUMMARY STEPS

1. **enable**
2. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Verifies that the language file was imported correctly.

- What to Do Next, page 99

What to Do Next

Proceed to the Creating the URL List, page 99.

Creating the URL List

SUMMARY STEPS

1. **enable**
2. **webvpn create template url-list** *device*:
3. Copy the XML file to a PC for editing.
4. Copy the edited url-list XML file back to the storage device.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**webvpn create template url-list** *device*: **Example:** `Router# webvpn create template url-list flash:`	Creates the url-list template.
Step 3	Copy the XML file to a PC for editing.	For an example of how to copy an XML file to a PC for editing, see the Example: URL List, page 129.
Step 4	Copy the edited url-list XML file back to the storage device.	For an example of how to copy the edited url-list XML file back to a storage device, see the Example: URL List, page 129.

- What to Do Next, page 100

What to Do Next

Proceed to the Importing the File into the URL List and Binding It to a Policy Group, page 100.

Importing the File into the URL List and Binding It to a Policy Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *name*
4. **url-list** *name*
5. **import** *device*:*file*
6. **exit**
7. **policy group group** *name*
8. **url-list** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**url-list** *name* **Example:** `Router(config-webvpn-context)# url-list testlist`	Enters WebVPN URL list configuration mode to configure a list of URLs to which a user has access on the portal page of an SSL VPN and attaches the URL list to a policy group.
Step 5	**import** *device*:*file* **Example:** `Router(config-webvpn-url)# import flash:testlist`	Imports the user-defined URL list.
Step 6	**exit** **Example:** `Router(config-webvpn-url)# exit`	Exits WebVPN URL list configuration mode.
Step 7	**policy group group** *name* **Example:** `Router(config-webvpn-context)# policy group policygroup1`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 8	**url-list** *name* **Example:** `Router(config-webvpn-group)# url-list testlist`	Binds the URL list to the policy group.

• What to Do Next, page 102

What to Do Next

Proceed to the Verifying That the URL List File Was Bound Correctly to the Policy Group, page 102.

Verifying That the URL List File Was Bound Correctly to the Policy Group

SUMMARY STEPS

1. **enable**
2. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Verifies that the url-list file was bound correctly to the policy group.

Configuring a Virtual Template

A virtual template enables SSL VPN to interoperate with IP features such as NAT, firewall, and policy-based routing.

- SSL VPN gateway and context configurations are enabled and operational.
- If a VRF is needed, configure it before creating the virtual template.
- If the virtual template is to be associated with a firewall security zone, create the security zone before creating the virtual template.

Note In order for a virtual template to work with SSL VPN, you must configure the**ip unnumbered** command on the virtual template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **ip unnumbered** *type number*
5. **exit**
6. **webvpn context** *name*
7. **virtual-template** *number*
8. **show webvpn context** [*name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 200`	Creates an interface for the virtual template and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered GigabitEthernet 0/0`	Enables IP processing on an interface without assigning an explicit IP address to the interface. • The *type* and *number* arguments specify another interface on which the switch has an assigned IP address. The interface specified cannot be another unnumbered interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 6	**webvpn context** *name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.

	Command or Action	Purpose
Step 7	**virtual-template** *number* **Example:** `Router(config-webvpn-context)# virtual-template 200`	Associates a virtual template with an SSL VPN context.
Step 8	**show webvpn context** [*name*] **Example:** `Router# show webvpn context context1`	Verifies that the virtual template is configured correctly.

Configuring SSLVPN DVTI Support

Configuring per-Tunnel Virtual Templates

Perform this task to configure per-tunnel virtual templates. This task describes how to provide DVTI support for an SSL VPN.

A virtual template is configured with the desired IP features. This virtual template is configured in a WebVPN context on a per-tunnel or per-user basis (because a user will have only one tunnel established at a time). Hence the virtual template configuration is applied on a per-tunnel basis for each SSL VPN full tunnel established in the WebVPN context. This configuration also helps you apply a distinct configuration to each user connecting to the WebVPN context using a AAA server.

The distinct per-user policy configuration is downloaded from the AAA server. This configuration includes group policy attributes and ACLs, and is applied to every user connecting to the WebVPN context on a per-user basis.

If a per-user attribute such as ACL is configured both on the AAA server and the virtual template, then the attribute configured on the AAA server takes precedence. The users logged in to the client computer will have the ACL configuration from the AAA server but will have other configurations, such as firewalls and VRF, from the virtual template. That is, the configuration applied to the users will be a combination of the virtual template configuration and the configuration available on the AAA server.

For example, if IP features such as firewalls, ACLs, and VRF are configured in a virtual template and user attributes such as ACLs are configured on the AAA server, the attributes configured on the AAA server take precedence. The users logged in to the client computer will have the ACL configuration from the AAA server but will have firewall and VRF configurations from the virtual template. That is, the configuration applied to the users will be a combination of virtual templates and AAA, where AAA attributes have a higher priority when there is a configuration conflict.

See the Configuring RADIUS Attribute Support for SSL VPN, page 51 for a list of AAA attributes that support SSL VPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *context-name*
4. **virtual-template** *interface-number* **tunnel**
5. **inservice**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *context-name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**virtual-template** *interface-number* **tunnel** **Example:** `Router(config-webvpn-context)# virtual-template 1 tunnel`	Associates virtual templates for each full tunnel session.
Step 5	**inservice** **Example:** `Router(config-webvpn-context)# inservice`	Enables an SSL VPN context. **Note** If a context is already configured and enabled, then you must disable the context using the **no inservice** command, specify the virtual template using the **virtual-template** *interface-number* command, and then enable the SSL VPN context using the **inservice** command.

Command or Action	Purpose
Step 6 **end** **Example:** `Router(config-webvpn-context)# end`	Exits WebVPN context configuration mode.

- Troubleshooting Tips, page 106

Troubleshooting Tips

Use the following commands to debug any errors that you may encounter when you configure the per-Tunnel Virtual Templates:

- **debug vtemplate** {**cloning** | **error** | **event**}
- **debug webvpn tunnel**

Configuring per-Context Virtual Templates

This task describes how to configure virtual tunnel interface support on a per-context basis.

A virtual template is configured with IP features such as NAT, firewalls, and PBR. This virtual template is configured in a WebVPN context, and enables SSL VPN to interoperate with the IP features configured. This configuration is applied to all users connecting to that WebVPN context.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *context-name*
4. **virtual-template** *interface-number*
5. **inservice**
6. **end**

DETAILED STEPS

Command or Action	Purpose
Step 1 **enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *context-name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**virtual-template** *interface-number* **Example:** `Router(config-webvpn-context)# virtual-template 1`	Associates a virtual template with an SSL VPN context.
Step 5	**inservice** **Example:** `Router(config-webvpn-context)# inservice`	Enables an SSL VPN context. **Note** If a context is already configured and enabled, then you must disable the context using the **no inservice** command, specify the virtual template using the **virtual-template** *interface-number* command, and then enable the SSL VPN context using the **inservice** command.
Step 6	**end** **Example:** `Router(config-webvpn-context)# end`	Exits WebVPN context configuration mode.

- Troubleshooting Tips, page 107

Troubleshooting Tips

Use the following commands to debug any errors that you may encounter when you configure the per-Context Virtual Templates:

- **debug vtemplate** {**cloning** | **error** | **event**}
- **debug webvpn tunnel**

Configuring SSL VPN Phase-4 Features

- Configuring the Start Before Logon Functionality, page 108
- Configuring Split ACL Support, page 110

- Configuring IP NetMask Functionality, page 112

Configuring the Start Before Logon Functionality

In order to import the AnyConnect profile to the Cisco IOS headend, the administrator must download the AnyConnect profile from an AnyConnect client (this profile comes by default with AnyConnect), update the UseStartBeforeLogin XML tag available in the profile file to inform AnyConnect to support SBL, and then import the modified profile into the Cisco IOS software.

The secure gateway administrator maintains the AnyConnect profile file and distributes it to the clients.

Following is an extract of the Cisco IOS AnyConnect VPN client profile XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AnyConnectProfile xmlns="http://schemas.xmlsoap.org/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/encoding/
AnyConnectProfile.xsd">
<ClientInitialization>
<UseStartBeforeLogon UserControllable="false">true</UseStartBeforeLogon>
</ClientInitialization>
```

You can select the hosts from the above list.

```
<ServerList>
  <HostEntry>
    <HostName>abc</HostName>
    <HostAddress>abc.cisco.com</HostAddress>
  </HostEntry> </ServerList>
</AnyConnectProfile>
```

Data is required to connect to a specific host.

The SBL functionality connects the client PC to the enterprise network even before the users log into the PC. This functionality allows the administrator to run the logon scripts even if the user is not connected to the enterprise network. This is useful for a number of deployment scenarios where the user is outside the physical corporate network and cannot access the resources until his system is connected to the corporate network.

Only an administrator can enable or disable SBL. The end users accessing the client PC are not allowed to enable or disable this functionality.

SSL VPN must have the ability to import profiles on the Cisco IOS software and must be able to send the AnyConnect profile to the client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn import svc profile** *profile-name device-name*
4. **webvpn context** *context-name*
5. **policy group** *group-name*
6. **svc profile** *profile-name*
7. **svc module** *module-name*
8. **end**
9. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn import svc profile** *profile-name device-name* **Example:** `Router(config)# webvpn import svc profile profile1 flash:newName`	Imports the AnyConnect profile to the Cisco IOS headend.
Step 4	**webvpn context** *context-name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 5	**policy group** *group-name* **Example:** `Router(config-webvpn-context)# policy group group1`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 6	**svc profile** *profile-name* **Example:** `Router(config-webvpn-group)# svc profile profile1`	Applies the concerned profile to the respective WebVPN group policy.
Step 7	**svc module** *module-name* **Example:** `Router(config-webvpn-group)# svc module vpngina`	Enables the SBL functionality support for the Cisco IOS SSL VPN headend. **Note** Only the vpngina SVC module is supported.

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(config-webvpn-group)# end`	Exits WebVPN group policy configuration mode. **Note** You must restart your system for the SBL functionality to take effect.
Step 9	**show running-config** **Example:** `Router# show running-config`	(Optional) Displays the contents of the current running configuration file or the configuration for a specific module, Layer 2 VLAN, class map, interface, map class, policy map, or virtual circuit (VC) class.

- Troubleshooting Tips, page 110

Troubleshooting Tips

Use the **debug webvpn cookie** command to debug any errors that you may encounter when you configure the SBL functionality.

Configuring Split ACL Support

Perform this task to configure split ACL support.

When the tunnel is active, Cisco IOS SSL VPN supports the **split include** and **split exclude** commands to filter and classify the traffic based on IP. Because the Cisco IOS software supports ACLs to classify the traffic, standard ACL support is provided to filter the traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list standard** {*access-list-number* | *access-list-name*}
4. **permit** *ip-address*
5. **deny** *ip-address*
6. **exit**
7. **webvpn context** *context-name*
8. **policy group** *policy-name*
9. **svc split** {**include** | **exclude**} **acl** *acl-list-name*
10. **end**
11. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list standard** {*access-list-number* | *access-list-name*} **Example:** `Router(config)# ip access-list standard 1`	Defines an IP access list or object group access control list (OGACL) by name or number and enters the standard ACL configuration mode.
Step 4	**permit** *ip-address* **Example:** `Router(config-std-nacl)# permit 10.0.0.1`	Sets conditions to allow packets to pass a named SSL VPN access list. **Note** You can use the **permit** and **deny** commands in any combination, as required.
Step 5	**deny** *ip-address* **Example:** `Router(config-std-nacl)# deny 10.0.0.2`	Sets conditions in a named SSL VPN access list that will deny packets. **Note** You can use the **permit** and **deny** commands in any combination, as required.
Step 6	**exit** **Example:** `Router(config-std-nacl)# exit`	Exits standard ACL configuration mode.
Step 7	**webvpn context** *context-name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 8	**policy group** *policy-name* **Example:** `Router(config-webvpn-context)# policy group default`	Enters WebVPN group policy configuration mode to configure a group policy.

	Command or Action	Purpose
Step 9	**svc split** {**include** | **exclude**} **acl** *acl-list-name* **Example:** `Router(config-webvpn-group)# svc split include acl 1`	Enables split tunneling for Cisco AnyConnect VPN Client tunnel clients.
Step 10	**end** **Example:** `Router(config-webvpn-group)# end`	Exits WebVPN group policy configuration mode.
Step 11	**show running-config** **Example:** `Router# show running-config`	(Optional) Displays the contents of the current running configuration file or the configuration for a specific module, Layer 2 VLAN, class map, interface, map class, policy map, or virtual circuit (VC) class.

Configuring IP NetMask Functionality

The IP NetMask functionality provides SVC or AnyConnect client provision to configure the network mask when the **ip local pool** command is configured on the router. This mask must be a classless mask.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn context** *context-name*
4. **policy group** *group-name*
5. **svc address-pool** *pool-name* [**netmask** *ip-mask*]
6. **end**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn context** *context-name* **Example:** `Router(config)# webvpn context context1`	Enters WebVPN context configuration mode to configure the SSL VPN context.
Step 4	**policy group** *group-name* **Example:** `Router(config-webvpn-context)# policy group default`	Enters WebVPN group policy configuration mode to configure a group policy.
Step 5	**svc address-pool** *pool-name* [**netmask** *ip-mask*] **Example:** `Router(config-webvpn-group)# svc address-pool pool1 netmask 255.255.0.0`	Configures the desired netmask on the router.
Step 6	**end** **Example:** `Router(config-webvpn-group)# end`	Exits WebVPN group policy configuration mode.
Step 7	**show running-config** **Example:** `Router# show running-config`	(Optional) Displays the contents of the current running configuration file or the configuration for a specific module, Layer 2 VLAN, class map, interface, map class, policy map, or virtual circuit (VC) class.

Configuring the DTLS Port

DTLS listens on port 443 by default. Perform this task to configure the desired DTLS port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn gateway** *gateway-name*
4. **dtls port** *port-number*
5. **end**
6. **show webvpn session** [**user** *user-name*] **context** {*context-name* | **all**} [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn gateway** *gateway-name* **Example:** `Router(config)# webvpn gateway gateway1`	Enters WebVPN gateway configuration mode to configure a SSL VPN gateway.
Step 4	**dtls port** *port-number* **Example:** `Router(config-webvpn-gateway)# dtls port 1045`	Configures a DTLS port.
Step 5	**end** **Example:** `Router(config-webvpn-gateway)# end`	Exits WebVPN gateway configuration mode.

	Command or Action	Purpose
Step 6	**show webvpn session** [**user** *user-name*] **context** {*context-name* | **all**} [**detail**] **Example:** `Router# show webvpn session context all`	(Optional) Displays SSL VPN user session information.

- Troubleshooting Tips, page 115

Troubleshooting Tips

The **debug webvpn dtls** [**errors** | **events** | **packets**] command can help troubleshoot IOS SSL VPN DTLS support.

Using SSL VPN clear Commands

This section describes **clear** commands that are used to perform the following tasks:

- Clear NBNS cache information
- Clear remote user sessions
- Clear (or reset) SSL VPN application and access counters

SUMMARY STEPS

1. **enable**
2. **clear webvpn nbns** [**context** {*name* | **all**}]
3. **clear webvpn session** [**user** *name*] **context** {*name* | **all**}
4. **clear webvpn stats** [**cifs** | **citrix** | **mangle** | **port-forward** | **sso** | **tunnel**] [**context** {*name* | **all**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear webvpn nbns** [**context** {*name* | **all**}] **Example:** `Router# clear webvpn nbns context all`	Clears the NBNS cache on an SSL VPN gateway.

	Command or Action	Purpose
Step 3	**clear webvpn session** [**user** *name*] **context** {*name* | **all**} **Example:** `Router# clear webvpn session context all`	Clears SSL VPN remote user sessions.
Step 4	**clear webvpn stats** [**cifs** | **citrix** | **mangle** | **port-forward** | **sso** | **tunnel**] [**context** {*name* | **all**}] **Example:** `Router# clear webvpn stats`	Clears SSL VPN application and access counters.

Verifying SSL VPN Configurations

This section describes how to use **show** commands to verify the following:

- SSL VPN gateway configuration
- SSL VPN context configuration
- CSD and Cisco AnyConnect VPN Client installation status
- NetBIOS name services information
- SSL VPN group policy configuration
- SSL VPN user session information
- SSL VPN application statistics
- SSLVPN DVTI Support configuration

SUMMARY STEPS

1. **enable**
2. **show webvpn context** [*name*]
3. **show webvpn gateway** [*name*]
4. **show webvpn install** {**file** *name* | **package** {**csd** | **svc**} | **status** {**csd** | **svc**}}
5. **show webvpn nbns context** {**all** | *name*}
6. **show webvpn policy group** *name* **context** {**all** | *name*}
7. **show webvpn session** [**user** *name*] **context** {**all** | *name*}
8. **show webvpn stats** [**cifs** | **citrix** | **mangle** | **port-forward** | **sso** | **tunnel**] [**detail**] [**context** {**all** | *name*}]
9. **show webvpn context** [*context-name* | **brief**]
10. **show interface virtual-access** *interface-number*
11. **show webvpn session** [**user** *user-name*] **context** {*context-name* | **all**} [**detail**]
12. **show running-config interface virtual-access** *interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show webvpn context** [*name*] **Example:** `Router# show webvpn context`	Displays the operational status and configuration parameters for SSL VPN context configurations.
Step 3	**show webvpn gateway** [*name*] **Example:** `Router# show webvpn gateway`	Displays the status of the SSL VPN gateway.
Step 4	**show webvpn install** {**file** *name* | **package** {**csd** | **svc**} | **status** {**csd** | **svc**}} **Example:** `Router# show webvpn install status csd`	Displays the installation status of Cisco AnyConnect VPN Client or CSD client software packages.
Step 5	**show webvpn nbns context** {**all** | *name*} **Example:** `Router# show webvpn nbns context all`	Displays information in the NBNS cache.
Step 6	**show webvpn policy group** *name* **context** {**all** | *name*} **Example:** `Router# show webvpn policy group ONE context all`	Displays the context configuration associated with a policy group.
Step 7	**show webvpn session** [**user** *name*] **context** {**all** | *name*} **Example:** `Router# show webvpn session context all`	Displays SSL VPN user session information.

	Command or Action	Purpose
Step 8	**show webvpn stats** [**cifs** | **citrix** | **mangle** | **port-forward** | **sso** | **tunnel**] [**detail**] [**context** {**all** | *name*}] **Example:** `Router# show webvpn stats tunnel detail context all`	Displays SSL VPN application and network statistics.
Step 9	**show webvpn context** [*context-name* | **brief**] **Example:** `Router# show webvpn context brief`	(Optional) Displays the operational status and configuration parameters for SSL VPN context configurations.
Step 10	**show interface virtual-access** *interface-number* **Example:** `Router# show interface virtual-access 1`	(Optional) Displays detailed information about the virtual access interface.
Step 11	**show webvpn session** [**user** *user-name*] **context** {*context-name* | **all**} [**detail**] **Example:** `Router# show webvpn session user user1 context all`	(Optional) Displays SSL VPN user session information.
Step 12	**show running-config interface virtual-access** *interface-number* **Example:** `Router# show running-config interface virtual-access 1`	(Optional) Displays the configuration applied on the virtual access interface.

Using SSL VPN Debug Commands

To monitor and manage your SSL VPN configurations, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug webvpn** [**verbose**] [**aaa** | **acl** | **cifs** | **citrix** [**verbose**] | **cookie** [**verbose**] | **count** | **csd** | **data** | **dns** | **emweb** [**state**] | **entry** *context-name* [**source** *ip*[*network-mask*] | **user** *username*] | **http** [**authentication** | **trace** | **verbose**] | **package** | **sdps** [**level** *number*] | **sock** [**flow**] | **sso**| **timer** | **trie** | **tunnel** [**traffic** *acl-number* | **verbose**] | **url-disp** | **webservice** [**verbose**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug webvpn** [**verbose**] [**aaa** | **acl** | **cifs** | **citrix** [**verbose**] | **cookie** [**verbose**] | **count** | **csd** | **data** | **dns** | **emweb** [**state**] | **entry** *context-name* [**source** *ip*[*network-mask*] | **user** *username*] | **http** [**authentication** | **trace** | **verbose**] | **package** | **sdps** [**level** *number*] | **sock** [**flow**] | **sso**| **timer** | **trie** | **tunnel** [**traffic** *acl-number* | **verbose**] | **url-disp** | **webservice** [**verbose**]] **Example:** `Router# debug webvpn`	Enables the display of debug information for SSL VPN applications and network activity.

Configuration Examples for SSL VPN

Example: Configuring a Generic SSL VPN Gateway

The following output example shows that a generic SSL VPN gateway has been configured in privileged EXEC mode:

```
webvpn gateway SSL_gateway2
 ip address 10.1.1.1. port 442
 ssl trustpoint TP_self_signed _4138349635
 inservice
!
webvpn context SSL_gateway2
 ssl authenticate verify all
!
!
policy group default
default-group-policy default
 gateway SSL_gateway2
inservice
```

Example: Configuring an ACL

The following output example shows the ACL is “acl1.” It has been associated with policy group “default.”

```
webvpn context context1
 ssl authenticate verify all
 !
 acl "acl1"
   error-msg "warning!!!..."
   permit url "http://www.example1.com"
   deny url "http://www.example2.com"
   permit http any any
 !
 nbns-list l1
   nbns-server 10.1.1.20
 !
 cifs-url-list "c1"
   heading "cifs-url"
   url-text "SSL VPN-SERVER2" url-value "\\SSL VPN-SERVER2"
   url-text "SSL-SERVER2" url-value "\\SSL-SERVER2"
 !
 policy group default
   acl "acl1"
   cifs-url-list "c1"
   nbns-list "l1"
   functions file-access
   functions file-browse
   functions file-entry
 default-group-policy default
 gateway public
 inservice
!
```

Example: Configuring HTTP Proxy

The following output example shows that HTTP proxy has been configured and that the portal (home) page from URL "http://www.example.com" will automatically download the home page of the user:

```
webvpn context myContext
 ssl authenticate verify all
 !
 !
 port-forward "email"
   local-port 20016 remote-server "ssl-server1.SSL example1.com" remote-port 110
description "POP-ssl-server1"
```

```
!
policy group myPolicy
  port-forward "email" auto-download http-proxy proxy-url "http://www.example.com"
inservice
```

Example: Configuring Microsoft File Shares for Clientless Remote Access

NBNS Server List Example

The following example, starting in global configuration mode, configures a server list for NBNS resolution:

```
Router(config)# webvpn context context1
Router(config-webvpn-context)# nbns-list SERVER_LIST
Router(config-webvpn-nbnslist)# nbns-server 172.16.1.1 master
Router(config-webvpn-nbnslist)# nbns-server 172.16.2.2 timeout 10 retries 5
Router(config-webvpn-nbnslist)# nbns-server 172.16.3.3 timeout 10 retries 5
Router(config-webvpn-nbnslist)# exit
```

File Share Permissions Example

The following example attaches the server list to and enables full file and network access permissions for policy group ONE:

```
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# nbns-list SERVER_LIST
Router(config-webvpn-group)# functions file-access
Router(config-webvpn-group)#  functions file-browse
Router(config-webvpn-group)# functions file-entry
Router(config-webvpn-group)# end
```

Example: Configuring Citrix Application Support for Clientless Remote Access

The following example, starting in global configuration mode, enables Citrix application support for remote users with a source IP address in the 192.168.1.0/24 network:

```
Router(config)# access-list 100 permit ip 192.168.1.0 0.255.255.255 any
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# citrix enabled
Router(config-webvpn-group)# filter citrix 100
```

Example: Configuring Application Port Forwarding

The following example, starting in global configuration mode, configures port forwarding for well-known e-mail application port numbers:

```
Router(config)# webvpn context context1
Router(config-webvpn-context)# port-forward EMAIL
Router(config-webvpn-port-fwd)# local-port 30016 remote-server mail1.company.com remote-
port 110 description POP3
Router(config-webvpn-port-fwd)# local-port 30017 remote-server mail2.company.com remote-
port 25 description SMTP
Router(config-webvpn-port-fwd)# local-port 30018 remote-server mail3.company.com remote-
port 143 description IMAP
Router(config-webvpn-port-fwd)# exit
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# port-forward EMAIL
Router(config-webvpn-group)# end
```

Example: Configuring VRF Virtualization

The following example, starting in global configuration mode, associates the VRF under the SSL VPN context configuration:

```
Router(config)# ip vrf vrf1
Router(config-vrf)# rd 10.100.100.1:1
Router(config-vrf)# exit
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group group1
Router(config-webvpn-group)# exit
Router(config-webvpn-context)# default-group-policy policy1
Router(config-webvpn-context)# vrf-name vrf2
Router(config-webvpn-context)# end
```

When you configure the VRF Virtualization feature in Cisco IOS Release 12.4(24)T1 and later releases, the following message is displayed:

```
% IP VRF vrf1 configuration applied.
% But please use Virtual-Template to configure VRF.
```

See the SSLVPN DVTI Support, page 27 for an example on how to use a virtual template to configure a VRF.

Example: RADIUS Accounting for SSL VPN Sessions

The following example shows that RADIUS accounting has been configured for SSL VPN user sessions:

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname host1
!
aaa new-model
!
!
aaa accounting network SSL VPNaaa start-stop group radius
aaa accounting update periodic 1
aaa session-id common
ip subnet-zero
ip cef
!
!
no ip domain lookup
ip domain name cisco.com
ip name-server 172.16.2.133
ip name-server 172.16.11.48
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
!
!
webvpn gateway GW1
 ip address 172.19.216.141 port 443
 inservice
 !
webvpn gateway SSL VPN
 no inservice
 !
webvpn install svc flash:/webvpn/svc.pkg
webvpn aaa accounting-list SSL VPNaaa
 !
```

```
webvpn context Default_context
 ssl encryption
 ssl authenticate verify all
 !
 no inservice
!
!
```

Example: URL Obfuscation (Masking)

The following output example shows that URL obfuscation (masking) has been configured for policy group “gp_urlobf.”

```
!
!
policy group gp_urlobf
   mask-urls
 default-group-policy gp_urlobf
 gateway gw domain dom
 inservice
!
!
```

Example: Adding a CIFS Server URL List and Attaching It to a Policy List

The following example shows that the CIFS server URLs "SSLVPN-SERVER2" and "SSL-SERVER2" have been added as portal page URLs to which a user has access. The example also shows that the two servers are attached to a policy group.

```
webvpn context context_1
 ssl authenticate verify all
 !
 acl "acl1"
   error-msg "warning!!!..."
   permit url "http://www.example1.com"
   deny url "http://www.example2.com"
   permit http any any
 !
 nbns-list l1
   nbns-server 10.1.1.20
 !
 cifs-url-list "c1"
   heading "cifs-url"
   url-text "SSLVPN-SERVER2" url-value "\\SSLVPN-SERVER2"
   url-text "SSL-SERVER2" url-value "\\SSL-SERVER2"
 !
 policy group default
   acl "acl1"
   cifs-url-list "c1"
   nbns-list "l1"
   functions file-access
   functions file-browse
   functions file-entry
 default-group-policy default
 gateway public
 inservice
!
```

Example: Typical SSL VPN Configuration

The following is an example of an SSL VPN configuration that includes most of the features that are available using SSL VPN:

```
hostname sslvpn
```

```
!
!
aaa new-model
!
!
aaa authentication login default local group radius
!
!
crypto pki trustpoint Gateway
 enrollment selfsigned
 ip-address 192.168.22.13
 revocation-check crl
 rsakeypair keys 1024 1024
!
!
crypto pki certificate chain Gateway
 certificate self-signed 02
!
!
interface Loopback0
 ip address 10.10.10.1 255.255.255.0
!
!
interface GigabitEthernet0/1
 ip address 192.168.22.14 255.255.255.0 secondary
 ip address 192.168.22.13 255.255.255.0
 duplex auto
 speed auto
 media-type rj45
!
!
ip local pool svc-pool 10.10.10.100 10.10.10.110
!
!
ip radius source-interface FastEthernet1/1
!
!
webvpn gateway ssl-vpn
 ip address 192.168.22.13 port 443
 http-redirect port 80
 ssl trustpoint Gateway
 inservice
 !
! The following line is required for SSLVPN Client.
webvpn install svc flash:/webvpn/svc.pkg
!
! The following line is required for Cisco Secure Desktop.
webvpn install csd flash:/webvpn/sdesktop.pkg
!
webvpn context ssl-vpn
 ssl authenticate verify all
!
 url-list "sslvpn-dt"
   url-text "sslvpn-dt" url-value "http://10.1.1.40"
   url-text "Exchange Server" url-value "http://10.1.1.40/exchange"
!
 sso-server "netegrity"
   web-agent-url "http://10.1.1.37/vpnauth/"
   secret-key "sslvpn1"
   retries 3
   timeout 15
!
 nbns-list cifs
   nbns-server 10.1.1.40
!
 port-forward "mail_test"
   local-port 30016 remote-server "example1.com" remote-port 143 description "IMAP-test"
   local-port 30017 remote-server "example2.com" remote-port 110 description "POP3-test"
   local-port 30018 remote-server "example3.com" remote-port 25 description "SMTP-test"
 !
 policy group default
! The following line applies the URL list.
   url-list "sslvpn-dt"
```

```
! The following line applies TCP port forwarding.
   port-forward "mail_test"
! The following line applies CIFS.
   nbns-list "cifs"
! The following line enables CIFS functionality.
   functions file-access
! The following line enables CIFS functionality.
   functions file-browse
! The following line enables CIFS functionality.
   functions file-entry
! The following line enables SSLVPN Client.
   functions svc-enabled
! The following line enables clientless Citrix.
   citrix enabled
 default-group-policy default
! The following line maps this context to the virtual gateway and defines the domain to
use.
 gateway ssl-vpn domain sslvpn
! The following line enables Cisco Secure Desktop.
 csd enable
 inservice
!
!
end
```

Example: Cisco Express Forwarding-Processed Packets

The following output example from the **show webvpn stats** command shows information about Cisco Express Forwarding-processed packets:

```
Router# show webvpn stats
User session statistics:
    Active user sessions       : 56         AAA pending reqs           : 0
    Peak user sessions         : 117        Peak time                  : 00:13:19
    Active user TCP conns      : 0         Terminated user sessions    : 144
    Session alloc failures     : 0          Authentication failures    : 0
    VPN session timeout        : 0          VPN idle timeout           : 0
    User cleared VPN sessions  : 0      Exceeded ctx user limit        : 0
    Exceeded total user limit  : 0
    Client process rcvd pkts   : 1971       Server process rcvd pkts   : 441004
    Client process sent pkts   : 921291     Server process sent pkts   : 2013
    Client CEF received pkts   : 1334       Server CEF received pkts   : 951610
    Client CEF rcv punt pkts   : 0          Server CEF rcv punt pkts   : 779
    Client CEF sent pkts       : 1944439    Server CEF sent pkts       : 0
    Client CEF sent punt pkts  : 21070      Server CEF sent punt pkts  : 0
```

Example: Multiple AnyConnect VPN Client Package Files

The following example shows that three AnyConnect VPN Client packages have been installed to a gateway and shows the resulting **show webvpn install** command output:

```
Router(config)# webvpn install svc vpn1_i386-Release-2.0.0077-k9.pkg sequence 6
Router(config)# webvpn install svc vpn2_powerpc-Release-2.0.0077-k9.pkg sequence 8
Router(config)# webvpn install svc svc_1.pkg sequence 4
Router# show webvpn install status svc

SSLVPN Package SSL-VPN-Client version installed:
CISCO STC win2k+
2,0,0148
Fri 12/29/2006 19:13:56.37
SSLVPN Package SSL-VPN-Client version installed:
CISCO STC Darwin_i386
2,0,0
Wed Nov  8 04:01:57 MST 2006
SSLVPN Package SSL-VPN-Client version installed:
CISCO STC Darwin_powerpc
```

```
2,0,0
Wed Nov  8 03:54:50 MST 2006
```

The following example shows that three AnyConnect VPN client packages have been configured and typical output from the **show running-config** command:

```
Router# show running-config | begin webvpn
webvpn install svc flash:/webvpn/svc_4.pkg sequence 4
!
webvpn install svc flash:/webvpn/svc_6.pkg sequence 6
!
webvpn install svc flash:/webvpn/svc_9.pkg sequence 9
```

Example: Local Authorization

The following example shows that local authorization has been configured:

```
aaa new-model
!
aaa authentication login default local
aaa authorization network default local
!
aaa attribute list l2
  attribute type banner "user2"
!
aaa attribute list l1
  attribute type banner "user1"
  attribute type urllist-name "my-url-list"
!
username user1 password 0 passwd1
username user1 aaa attribute list l1
username user2 password 0 passwd2
username user2 aaa attribute list l2
!
webvpn context best
  ssl authenticate verify all
  !
  url-list "my-url-list"
    heading "external url"
    url-text "example" url-value "http://www.example.com"
  !
  policy group default
  default-group-policy default
  aaa authorization list default
  gateway public domain d1
  inservice
```

Example: URL Rewrite Splitter

The following example shows that URL mangling has been configured for a specific host and IP address. The unmatched action has been defined as direct access.

```
webvpn context e1
  !
  url rewrite
    host "www.example.com"
    ip 10.1.0.0 255.255.0.0
    unmatched-action direct-access
  !
```

Example: Backend HTTP Proxy

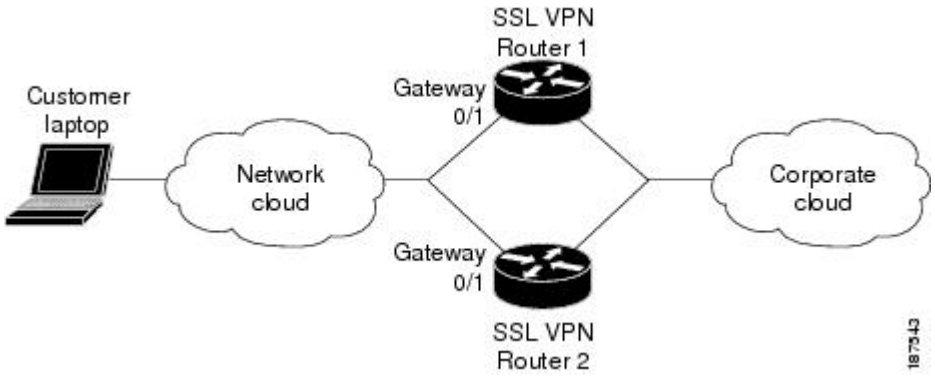
The following example shows that a backend HTTP proxy has been configured:

```
webvpn context e1
!
  policy group g1
    http proxy-server "192.0.2.0" port 2034
  default-group-policy g1
```

Example: Stateless High Availability with HSRP

The figure below shows the topology of a typical stateless high availability with HSRP setup. Router 1 and Router 2 are configured for HSRP on gateway Webvpn. The example following the figure below shows the actual configuration.

Figure 15 Stateless High Availability with HSRP Setup



Router 1 Configuration

```
Router(config)# interface gateway 0/1
Router(config-if)# standby 0 ip 10.1.1.1
Router(config-if)# standby 0 name SSLVPN
Route(config-if)# exit
Router(config)# webvpn gateway Webvpn
Router(config-webvpn-gateway)# ip address 10.1.1.1 port 443 standby SSLVPN
```

Router 2 Configuration

```
Router(config)# interface gateway 0/0
Router(config-if)# standby 0 ip 10.1.1.1
Router(config-if)# standby 0 name SSLVPN2
Router(config-if)# exit
Router(config)# webvpn gateway Webvpn
Router(config-webvpn-gateway)# ip address 10.1.1.1 port 443 standby SSLVPNigh2
```

Example: Internationalization

- Example: Generated Browser Attribute Template, page 128
- Example: Copying the Browser Attribute File to Another PC for Editing, page 128
- Example: Copying the Edited File to flash, page 128

Example: Generated Browser Attribute Template

The following is an example of a generated browser attribute template:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  - Template file for browser attributes import
    <color> - primary color
    <scolor> - secondary color
    <tcolor> - text color
    <stcolor> - secondary text color
    <lmsg> - login message
    <title> - browser title
    <ticolor> - title color
    Default value will be used if the field is not defined
  Copyright (c) 2007-2008 by Cisco Systems, Inc. All rights reserved.
-->
<settings>
  <color>#003333</color>
  <scolor>#336666</scolor>
  <tcolor>white</tcolor>
  <stcolor>black</stcolor>
  <lmsg>Welcome to<p>Cisco Systems WebVPN Service</lmsg>
  <title>WebVPN Service</title>
  <ticolor>#003333</ticolor>
</settings>
```

Example: Copying the Browser Attribute File to Another PC for Editing

The following example shows how to copy a browser attribute file to another PC for editing:

```
Router# copy flash: tftp:
Source filename [battr_tpl.xml
]?
Address or name of remote host []? 10.1.1.30
Destination filename [battr_tpl.xml
]?
!!
677 bytes copied in 0.004 secs (169250 bytes/sec)
```

Example: Copying the Edited File to flash

The following example shows how to copy an edited attribute file to flash:

```
Router# copy tftp://directory/edited_battr_tpl.xml
 flash:
```

Example: Output Showing That the Edited File Was Imported

The following **show running-config** output shows that the browser attribute file was correctly copied to flash:

```
Router# show running-config
```

```
webvpn context g
 browser-attribute import flash:battr_tpl.xml
 ssl authenticate verify all
```

Example: Copying the Language File to Another PC for Editing

The following example shows how to copy a language file to another PC for editing:

```
Router# copy flash: tftp:
Source filename [lang.js
]?
Address or name of remote host []? 10.1.1.30
Destination filename [lang.js
]?
!!
10649 bytes copied in 0.028 secs (380321 bytes/sec)
```

Example: Copying the Edited Language File to the Storage Device

The following example shows how to copy the edited language file to flash:

```
Router# copy tftp://directory/edited_lang.js flash:
```

Example: Language Template Created

The following **show running-config** command output shows that the language file "lang.js" has been imported correctly:

```
Router# show running-config
policy group default
   functions file-access
   functions file-browse
   functions file-entry
   functions svc-enabled
   mask-urls
   svc address-pool "mypool"
   svc keep-client-installed
   svc split include 10.1.1.0 255.255.255.0
  default-group-policy default
  gateway g
  language customize mylang flash:lang.js
inservice
```

Example: URL List

The following example shows that the URL list template file has been copied to another PC for editing:

```
Router# copy flash: tftp:
  Source filename [url_list_tpl.xml
]?
  Address or name of remote host []? 10.1.1.30
Destination filename [url_list_tpl.xml
```

The following example shows that the URL template file has been copied to flash:

```
Router# copy tftp://directory/edited_url_list_tpl.xml

flash:
```

The following **show running-config** command output shows that URL list file has been imported into the url-list and that it has been bound to the policy group:

```
Router# show running-config
```

```
policy group default
   url-list "test"
   functions file-access
   functions file-browse
   functions file-entry
   functions svc-enabled
   mask-urls
   svc address-pool "mypool"
   svc keep-client-installed
   svc split include 10.1.1.0 255.255.255.0
  default-group-policy default
  gateway g
  language customize mylang flash:lang.js
  inservice
```

Example: Virtual Template

The following configuration and output examples display various aspects of the virtual template feature. The following example, starting in global configuration mode, creates a virtual template and associates it with an SSL VPN context configuration. It also configures the virtual template for VRF and NAT:

```
Router(config)# interface virtual-template 100
Router(config-if)# ip unnumbered GigabitEthernet 0/0
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# webvpn context context1
Router(config-webvpn-context)# virtual-template 100
Router(config-webvpn-context)# exit
```

The following example creates a virtual template and associates it with a security zone:

```
Router(config)# interface virtual-template 200
Router(config-if)# ip unnumbered GigabitEthernet 0/0
Router(config-if)# zone-member security vpn
Router(config-if)# exit
Router(config)# webvpn context context2
Router(config-webvpn-context)# virtual-template 200
Router(config-webvpn-context)# exit
```

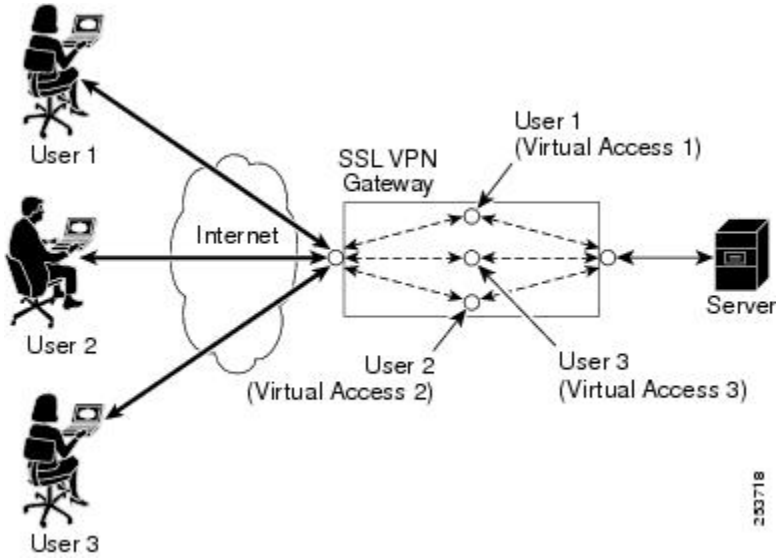
Example: SSL VPN DVTI Support

- Example: Configuring per-Tunnel Virtual Templates, page 130
- Example: Configuring per-Context Virtual Templates, page 133

Example: Configuring per-Tunnel Virtual Templates

The figure below shows an example network where remote users User1 and User2 belong to a context called Context1, User3 belongs to a context called Context2, and they connect to the SSL VPN gateway and access the backend server in the corporate network.

Figure 16 Topology Showing a per-Tunnel Virtual Template



This section contains the following examples:

- Example: Configuring in the per-Tunnel Context Using Virtual Templates, page 131
- Example: Configuring in the per-Tunnel Context Using Virtual Templates and a AAA Server, page 132

Example: Configuring in the per-Tunnel Context Using Virtual Templates

The following example shows how to apply VRF, a firewall policy, and ACLs to each user based on the virtual template configuration.

If the VRF, firewall policy, and ACL features are configured in the virtual template and user policies are not configured on the AAA server, then only the IP features configured in the virtual template are applied to the users. In this example, User1 and User2 belonging to Context1 have zone1, vrf1, and ACL 1 configured whereas User3 belonging to Context2 has zone3, vrf3, and ACL 3 configured. Hence, different users have different IP features configured.

Virtual Template for User1 and User2

```
configure terminal
 interface virtual-template 1
 zone-member security zone1
 ip vrf forwarding vrf1
 ip access-group 1 in
 ip unnumbered GigabitEthernet 0/1
```

Virtual Template for User3

```
configure terminal
 interface virtual-template 3
```

```
 zone-member security zone3
 ip vrf forwarding vrf3
 ip access-group 3 in
 ip unnumbered GigabitEthernet 0/1
```

WebVPN Context for User1 and User2

```
configure terminal
 webvpn context context1
  virtual-template 1 tunnel
  inservice
```

WebVPN Context for User3

```
configure terminal
 webvpn context context2
  virtual-template 3 tunnel
  inservice
```

Example: Configuring in the per-Tunnel Context Using Virtual Templates and a AAA Server

The following example shows how to apply the IP feature configuration to the users based on the user-specific configuration available on the AAA server. The user-specific attributes configured on the AAA server are applied to the users when an SSL VPN session establishes a virtual tunnel. The configuration applied to the users will be a combination of the configurations in the virtual template and the AAA server, where AAA attributes have a higher priority when there is a configuration conflict.

In this example, ACL 1 is configured for User1, ACL 2 is configured for User2, and ACL 3 is configured for User3 on the AAA server using the**inacl** attribute. Even though ACL 4 is applied to all the users in the virtual template, User1 has ACL 1, User2 has ACL 2, and User3 has ACL 3 configured along with zone and VRF configurations available in the virtual template.

Virtual Template for User1 and User2

```
configure terminal
 interface virtual-template 1
 zone-member security zone1
 ip vrf forwarding vrf1
 ip access-group 4 in
 ip unnumbered GigabitEthernet 0/1
```

Virtual Template for User3

```
configure terminal
 interface virtual-template 3
 zone-member security zone3
 ip vrf forwarding vrf3
 ip access-group 4 in
 ip unnumbered GigabitEthernet 0/1
```

WebVPN Context for User1 and User2

```
configure terminal
 webvpn context context1
  virtual-template 1 tunnel
  inservice
```

WebVPN Context for User3

```
configure terminal
```

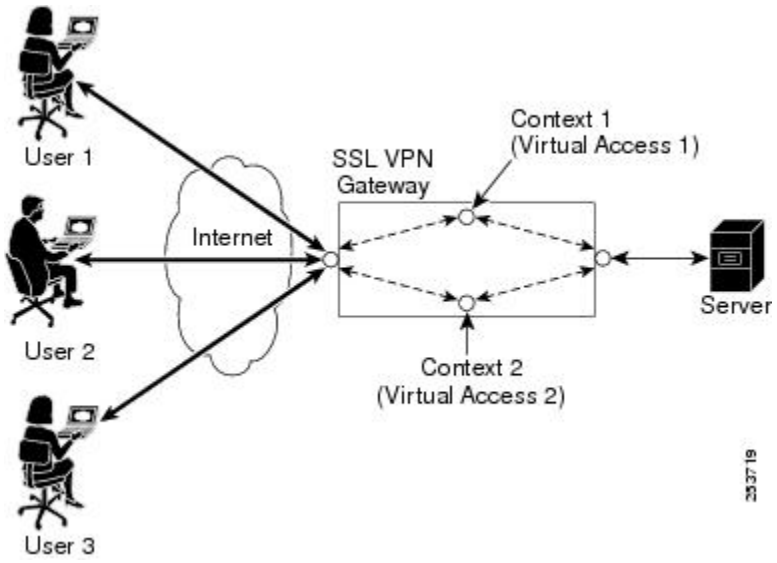
```
webvpn context context2
 virtual-template 3 tunnel
 inservice
```

Note You can configure different IP feature commands in the virtual template to configure SSL VPN interoperability with different IP features.

Example: Configuring per-Context Virtual Templates

The following figure shows remote users User1 and User2 belonging to context1 and User3 belonging to context2, connecting to the SSL VPN gateway and accessing the backend server in the corporate network. Here, the IP feature configuration is applied to each user based on the configuration applied to the WebVPN context of the user.

Figure 17 Topology Showing a per-Context Virtual Template



The following example shows how to apply VRF and a firewall policy to each user based on the WebVPN context of the user. In this example, User1 and User 2 connected to Context1 have zone1 and vrf1 configured on the virtual template 1, and User3 connected to Context2 has zone2 and vrf2 configured on virtual template 2.

Virtual Template for User1

```
configure terminal
 interface virtual-template 1
 zone-member security zone1
 ip vrf forwarding vrf1
 ip unnumbered GigabitEthernet 0/1
```

Virtual Template for User2

```
configure terminal
 interface virtual-template 2
```

```
 zone-member security zone2
 ip vrf forwarding vrf2
 ip unnumbered GigabitEthernet 0/1
```

WebVPN Context for User1

```
configure terminal
 webvpn context context1
  virtual-template 1
  inservice
```

WebVPN Context for User2

```
configure terminal
 webvpn context context2
  virtual-template 2
  inservice
```

Note You can configure different IP features in the virtual template to configure SSL VPN interoperability with different IP features.

Example: SSL VPN Phase-4 Features

Example: Configuring the Start Before Logon Functionality

The following example shows how to configure SBL functionality:

```
enable
 configure terminal
 webvpn import svc profile profile1 flash:newName
  policy group group1
   svc profile profile1
   end
```

Example: Configuring Split ACL Support

The following example shows how to configure split ACL support:

```
enable
 configure terminal
 ip access-list standard 1
  permit 10.0.0.1
  deny 10.0.0.2
  exit
 webvpn context context1
  policy group policy1
   svc split include acl 1
   end
```

Example: Configuring IP NetMask Functionality

The following example shows how to configure IP netmask functionality:

```
enable
 configure terminal
 webvpn context context1
  policy group policy1
   svc address-pool pool1 netmask 255.255.0.0
   end
```

Example: Debug Command Output

- Example: Configuring SSO, page 135

Example: Configuring SSO

The following output example displays ticket creation, session setup, and response handling information for an SSO configuration:

```
Router# debug webvpn sso
*Jun 12 20:37:01.052: WV-SSO: Redirect to SSO web agent URL - http://
example.examplecompany.com/vpnauth/
*Jun 12 20:37:01.052: WV_SSO: Set session cookie with SSO redirect
*Jun 12 20:37:01.056: WV-SSO: Set SSO auth flag
*Jun 12 20:37:01.056: WV-SSO: Attach credentials - building auth ticket
*Jun 12 20:37:01.060: WV-SSO: user: [user11], secret: [secret123], version: [1.0], login
time: [BCEFC86D], session key: [C077F97A], SHA1 hash :
[B07D0A924DB33988D423AE9F937C1C5A66404819]
*Jun 12 20:37:01.060: WV-SSO: auth_ticket :
user11:1.0@C077F97A@BCEFC86D@B07D0A924DB33988D423AE9F937C1C5A66404819
*Jun 12 20:37:01.060: WV-SSO: Base64 credentials for the auth_ticket:
dXNlcjExOjEuMEBDMDc3Rjk3QUBCQ0VGQzg2REBCMDdEMEE5MjREQjMzOTg4RDQyM0FFOUY5MzdDMUM1QTY2NDA0OD
E5
*Jun 12 20:37:01.060: WV-SSO: Decoded credentials =
user11:1.0@C077F97A@BCEFC86D@B07D0A924DB33988D423AE9F937C1C5A66404819
*Jun 12 20:37:01.060: WV-SSO: Starting SSO request timer for 15-second
*Jun 12 20:37:01.572: WV-SSO: SSO auth response rcvd - status[200]
*Jun 12 20:37:01.572: WV-SSO: Parsed non-SM cookie: SMCHALLENGE
*Jun 12 20:37:01.576: WV-SSO: Parsed SMSESSION cookie
*Jun 12 20:37:01.576: WV-SSO: Sending logon page after SSO auth success
```

Example: Show Command Output

- Example: show webvpn context, page 136
- Example: show webvpn context name, page 136
- Example: show webvpn gateway, page 136
- Example: show webvpn gateway name, page 136
- Example: show webvpn install file, page 137
- Example: show webvpn install package svc, page 137
- Example: show webvpn install status svc, page 137
- Example: show webvpn nbns context all, page 137
- Example: show webvpn policy, page 138
- Example: show webvpn policy (with NTLM Disabled), page 138
- Example: show webvpn session, page 138

Example: show webvpn context

The following is sample output from the **show webvpn context** command:

```
Router# show webvpn context
Codes: AS - Admin Status, OS - Operation Status
       VHost - Virtual Host
Context Name        Gateway  Domain/VHost      VRF      AS    OS
------------        -------  ------------      -------  ----  --------
Default_context     n/a      n/a               n/a      down  down
con-1               gw-1     one               -        up    up
con-2               -        -                 -        down  down
```

Example: show webvpn context name

The following is sample output from the **show webvpn context** command, entered with the name of a specific SSL VPN context:

```
Router# show webvpn context context1
Admin Status: up
Operation Status: up
CSD Status: Disabled
Certificate authentication type: All attributes (like CRL) are verified
AAA Authentication List not configured
AAA Authentication Domain not configured
Default Group Policy: PG_1
Associated WebVPN Gateway: GW_ONE
Domain Name: DOMAIN_ONE
Maximum Users Allowed: 10000 (default)
NAT Address not configured
VRF Name not configured
```

Example: show webvpn gateway

The following is sample output from the **show webvpn gateway** command:

```
Router# show webvpn gateway

Gateway Name                      Admin  Operation
------------                      -----  ---------
GW_1                              up     up
GW_2                              down   down
```

Example: show webvpn gateway name

The following is sample output from the **show webvpn gateway** command, entered with a specific SSL VPN gateway name:

```
Router# show webvpn gateway GW_1
Admin Status: up
Operation Status: up
IP: 10.1.1.1, port: 443
SSL Trustpoint: TP-self-signed-26793562
```

Example: show webvpn install file

The following is sample output from the **show webvpn install** command, entered with the **file** keyword:

```
Router# show webvpn install file \webvpn\stc\version.txt

SSL VPN File \webvpn\stc\version.txt installed:
CISCO STC win2k+ 1.0.0
1,1,0,116
Fri 06/03/2005 03:02:46.43
```

Example: show webvpn install package svc

The following is sample output from the **show webvpn install** command, entered with the **package svc**keywords:

```
Router# show webvpn install package svc
SSL VPN Package SSL-VPN-Client installed:
File: \webvpn\stc\1\binaries\detectvm.class, size: 555
File: \webvpn\stc\1\binaries\java.htm, size: 309
File: \webvpn\stc\1\binaries\main.js, size: 8049
File: \webvpn\stc\1\binaries\ocx.htm, size: 244
File: \webvpn\stc\1\binaries\setup.cab, size: 176132
File: \webvpn\stc\1\binaries\stc.exe, size: 94696
File: \webvpn\stc\1\binaries\stcjava.cab, size: 7166
File: \webvpn\stc\1\binaries\stcjava.jar, size: 4846
File: \webvpn\stc\1\binaries\stcweb.cab, size: 13678
File: \webvpn\stc\1\binaries\update.txt, size: 11
File: \webvpn\stc\1\empty.html, size: 153
File: \webvpn\stc\1\images\alert.gif, size: 2042
File: \webvpn\stc\1\images\buttons.gif, size: 1842
File: \webvpn\stc\1\images\loading.gif, size: 313
File: \webvpn\stc\1\images\title.gif, size: 2739
File: \webvpn\stc\1\index.html, size: 4725
File: \webvpn\stc\2\index.html, size: 325
File: \webvpn\stc\version.txt, size: 63
Total files: 18
```

Example: show webvpn install status svc

The following is sample output from the **show webvpn install** command, entered with the **status svc** keywords:

```
Router# show webvpn install status svc

SSL VPN Package SSL-VPN-Client version installed:
CISCO STC win2k+ 1.0.0
1,0,2,127
Fri 07/22/2005 12:14:45.43
```

Example: show webvpn nbns context all

The following sample output from the **show webvpn nbns** command, entered with the **context all** keywords:

```
Router# show webvpn nbns context all

NetBIOS name        IP Address       Timestamp
0 total entries
NetBIOS name        IP Address       Timestamp
0 total entries
NetBIOS name        IP Address       Timestamp
```

0 total entries

Example: show webvpn policy

The following is sample output from the **show webvpn policy** command:

```
Router# show webvpn policy group ONE context all
WEBVPN: group policy = ONE ; context = SSL VPN
      idle timeout = 2100 sec
      session timeout = 43200 sec
      citrix disabled
      dpd client timeout = 300 sec
      dpd gateway timeout = 300 sec
      keep SSL VPN client installed = disabled
      rekey interval = 3600 sec
      rekey method =
      lease duration = 43200 sec
WEBVPN: group policy = ONE ; context = SSL VPN_TWO
      idle timeout = 2100 sec
      session timeout = 43200 sec
      citrix disabled
      dpd client timeout = 300 sec
      dpd gateway timeout = 300 sec
      keep SSL VPN client installed = disabled
      rekey interval = 3600 sec
      rekey method =
      lease duration = 43200 sec
```

Example: show webvpn policy (with NTLM Disabled)

The following is sample output from the **show webvpn policy** command. NTLM authentication has been disabled.

```
Router# show webvpn policy group ntlm context ntlm
WEBVPN: group policy = ntlm; context = ntlm
      url list name = "ntlm-server"
      idle timeout = 2100 sec
      session timeout = 43200 sec
      functions =
                httpauth-disabled
                file-access
                svc-enabled

      citrix disabled
      dpd client timeout = 300 sec
      dpd gateway timeout = 300 sec
      keep SSL VPN client installed = disabled
      rekey interval = 3600 sec
      rekey method =
      lease duration = 43200 sec
```

Example: show webvpn session

The following is sample output from the **show webvpn session** command. The output is filtered to display user session information for only the specified context.

```
Router# show webvpn session context SSL VPN

WebVPN context name: SSL VPN
Client_Login_Name  Client_IP_Address  No_of_Connections  Created  Last_Used
user1              10.2.1.220                 2         04:47:16  00:01:26
user2              10.2.1.221                 2         04:48:36  00:01:56
```

Example: show webvpn session user

The following is sample output from the **show webvpn session** command. The output is filtered to display session information for a specific user.

```
Router# show webvpn session user user1 context all

WebVPN user name = user1 ; IP address = 10.2.1.220; context = SSL VPN
    No of connections: 0
    Created 00:00:19, Last-used 00:00:18
    CSD enabled
    CSD Session Policy
       CSD Web Browsing Allowed
       CSD Port Forwarding Allowed
       CSD Full Tunneling Disabled
       CSD FILE Access Allowed
    User Policy Parameters
      Group name = ONE
    Group Policy Parameters
      url list name = "Cisco"
      idle timeout = 2100 sec
      session timeout = 43200 sec
      port forward name = "EMAIL"
      tunnel mode = disabled
      citrix disabled
      dpd client timeout = 300 sec
      dpd gateway timeout = 300 sec
      keep stc installed = disabled
      rekey interval = 3600 sec
      rekey method = ssl
      lease duration = 3600 sec
```

Example: show webvpn stats

The following is sample output from the **show webvpn stats** command entered with the **detail** and **context** keywords:

```
Router# show webvpn stats detail context SSL VPN
WebVPN context name : SSL VPN
User session statistics:
    Active user sessions     : 0          AAA pending reqs         : 0
    Peak user sessions       : 0          Peak time                : never
    Active user TCP conns    : 0          Terminated user sessions : 0
    Session alloc failures   : 0          Authentication failures  : 0
    VPN session timeout      : 0          VPN idle timeout         : 0
    User cleared VPN sessions: 0          Exceeded ctx user limit  : 0
    CEF switched packets - client: 0         , server: 0
    CEF punted packets - client: 0         , server: 0
Mangling statistics:
    Relative urls            : 0          Absolute urls            : 0
    Non-http(s) absolute urls: 0          Non-standard path urls   : 0
    Interesting tags         : 0          Uninteresting tags       : 0
    Interesting attributes   : 0          Uninteresting attributes : 0
    Embedded script statement: 0          Embedded style statement : 0
    Inline scripts           : 0          Inline styles            : 0
    HTML comments            : 0          HTTP/1.0 requests        : 0
    HTTP/1.1 requests        : 0          Unknown HTTP version     : 0
    GET requests             : 0          POST requests            : 0
    CONNECT requests         : 0          Other request methods    : 0
    Through requests         : 0          Gateway requests         : 0
    Pipelined requests       : 0          Req with header size >1K : 0
    Processed req hdr bytes  : 0          Processed req body bytes : 0
    HTTP/1.0 responses       : 0          HTTP/1.1 responses       : 0
    HTML responses           : 0          CSS responses            : 0
    XML responses            : 0          JS responses             : 0
    Other content type resp  : 0          Chunked encoding resp    : 0
    Resp with encoded content: 0          Resp with content length : 0
    Close after response     : 0          Resp with header size >1K: 0
    Processed resp hdr size  : 0          Processed resp body bytes: 0
    Backend https response   : 0          Chunked encoding requests: 0
CIFS statistics:
  SMB related Per Context:
    TCP VC's                 : 0          UDP VC's                 : 0
```

```
    Active VC's             : 0          Active Contexts          : 0
    Aborted Conns           : 0
  NetBIOS related Per Context:
    Name Queries            : 0          Name Replies             : 0
    NB DGM Requests         : 0          NB DGM Replies           : 0
    NB TCP Connect Fails    : 0          NB Name Resolution Fails : 0
  HTTP related Per Context:
    Requests                : 0          Request Bytes RX         : 0
    Request Packets RX      : 0          Response Bytes TX        : 0
    Response Packets TX     : 0          Active Connections       : 0
    Active CIFS context     : 0          Requests Dropped         : 0

Socket statistics:
    Sockets in use          : 0          Sock Usr Blocks in use   : 0
    Sock Data Buffers in use : 0         Sock Buf desc in use     : 0
    Select timers in use    : 0          Sock Select Timeouts     : 0
    Sock Tx Blocked         : 0          Sock Tx Unblocked        : 0
    Sock Rx Blocked         : 0          Sock Rx Unblocked        : 0
    Sock UDP Connects       : 0          Sock UDP Disconnects     : 0
    Sock Premature Close    : 0          Sock Pipe Errors         : 0
    Sock Select Timeout Errs : 0
Port Forward statistics:
    Connections serviced    : 0          Server Aborts (idle)     : 0
  Client                                Server
    in pkts                 : 0          out pkts                 : 0
    in bytes                : 0          out bytes                : 0
    out pkts                : 0          in pkts                  : 0
    out bytes               : 0          in bytes                 : 0
WEBVPN Citrix statistics:
Connections serviced : 0
              Server                    Client
  Packets in  : 0                        0
  Packets out : 0                        0
  Bytes in    : 0                        0
  Bytes out   : 0                        0
Tunnel Statistics:
    Active connections      : 0
    Peak connections        : 0          Peak time                : never
    Connect succeed         : 0          Connect failed           : 0
    Reconnect succeed       : 0          Reconnect failed         : 0
    SVCIP install IOS succeed: 0         SVCIP install IOS failed : 0
    SVCIP clear IOS succeed  : 0         SVCIP clear IOS failed   : 0
    SVCIP install TCP succeed: 0         SVCIP install TCP failed : 0
    DPD timeout             : 0
  Client                                Server
    in  CSTP frames         : 0          out IP pkts              : 0
    in  CSTP data           : 0          out stitched pkts        : 0
    in  CSTP control        : 0          out copied pkts          : 0
    in  CSTP Addr Reqs      : 0          out bad pkts             : 0
    in  CSTP DPD Reqs       : 0          out filtered pkts        : 0
    in  CSTP DPD Resps      : 0          out non fwded pkts       : 0
    in  CSTP Msg Reqs       : 0          out forwarded pkts       : 0
    in  CSTP bytes          : 0          out IP bytes             : 0
    out CSTP frames         : 0          in  IP pkts              : 0
    out CSTP data           : 0          in  invalid pkts         : 0
    out CSTP control        : 0          in  congested pkts       : 0
    out CSTP Addr Resps     : 0          in  bad pkts             : 0
    out CSTP DPD Reqs       : 0          in  nonfwded pkts        : 0
    out CSTP DPD Resps      : 0          in  forwarded pkts       : 0
    out CSTP Msg Reqs       : 0
    out CSTP bytes          : 0          in  IP bytes             : 0
```

Example: show webvpn stats sso

The following output example displays statistics for an SSO server:

```
Router# show webvpn stats sso
Single Sign On statistics:
   Auth Requests          : 4         Pending Auth Requests    :0
   Successful Requests    : 1         Failed Requests          :3
   Retranmissions         : 0         DNS Errors               :0
```

```
   Connection Errors       : 0         Request Timeouts         :0
   Unknown Responses       :
```

The following output example displays extra information about SSO servers that are configured for the SSL VPN context:

```
Router# show webvpn context test_sso
  Context SSO server: sso-server
      Web agent URL :  "http://example1.examplecompany.com/vpnauth/"
      Policy Server Secret : "Secret123"
      Request Re-tries : 5, Request timeout: 15-second
```

The following output example displays extra information about an SSO server that is configured for the policy group of the SSL VPN context:

```
Router# show webvpn policy group sso context test_sso

WV: group policy = sso ; context = test_sso
      idle timeout = 2100 sec
      session timeout = 43200 sec
      sso server name = "server1"
      citrix disabled
      dpd client timeout = 300 sec
      dpd gateway timeout = 300 sec
      keep SSL VPN client installed = disabled
      rekey interval = 3600 sec
      rekey method =
      lease duration = 43200 sec
```

Example: FVRF show Command Output

The following output example shows that FVRF has been configured:

```
Router# show webvpn gateway mygateway
Admin Status: down
Operation Status: down
Error and Event Logging: Disabled
GW IP address not configured
SSL Trustpoint: TP-self-signed-788737041
FVRF Name: vrf_1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco AnyConnect VPN Client	• http://www.cisco.com/en/US/partner/products/ps6496/tsd_products_support_series_home.html Cisco SSL VPN Client Home Page • Cisco AnyConnect VPN Client Administrator Guide, Release 2.4 • Release Notes for Cisco AnyConnect VPN Client, Release 2.4
Cisco Secure Desktop	Cisco Secure Desktop Home Page

Related Topic	Document Title
Configuring IP VRF (**ip vrf** command)	*Cisco IOS IP Application Services Command Reference*
IANA Application Port Numbers	Port Numbers
RADIUS accounting	Configuring RADIUS module in the *RADIUS Configuration Guide*
Security commands	Cisco IOS Security Command Reference
SSL VPN platforms	Cisco IOS SSL VPN (“Feature Availability” section)
SSL VPN remote users guide	SSL VPN Remote User Guide

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SSL VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6 Feature Information for SSL VPN

Feature Name	Release	Feature Information
Access Control Enhancements	12.4(20)T	This feature allows administrators to configure automatic authentication and authorization for users. Users provide their usernames and passwords via the gateway page URL and do not have to reenter their usernames and passwords from the login page. Authorization is enhanced to support more generic authorization, including local authorization. The following commands were introduced by this feature: **aaa authentication auto, aaa authorization list.**

Feature Name	Release	Feature Information
AnyConnect Client Support	12.4(20)T	Effective with this release, AnyConnect Client adds support for several client-side platforms, such as Microsoft Windows, Apple-Mac, and Linux. The ability to install AnyConnect in a standalone mode is also added. In addition, this feature allows multiple SSL VPN client package files to be configured on a gateway. The following command was modified by this feature: **webvpn install.**
Application ACL Support	12.4(11)T	This feature provides administrators with the flexibility to fine-tune access control at the application layer level. The following commands were introduced by this feature: **acl add error-msg, error-url, list.**
Auto Applet Download	12.4(9)T	This feature provides administrators with the option of automatically downloading the port-forwarding applet under the policy group. The following command was modified by this feature: **port-forward** (policy group).
Backend HTTP Proxy	12.4(20)T	This feature allows administrators to route user requests through a backend HTTP proxy, providing more flexibility and control than routing through internal web servers. The following command was added by this feature: **http proxy-server.**

Feature Name	Release	Feature Information
Cisco AnyConnect VPN Client	12.4(15)T	This feature is the next-generation SSL VPN Client. The feature provides remote users with secure VPN connections to the router platforms supported by SSL VPN and to the Cisco 5500 Series Adaptive Security Appliances. Users having Cisco IOS releases before Release 12.4(15)T see *SSL VPN Client GUI*. Users having Release 12.4(15)T and later releases see *Cisco AnyConnect VPN Client GUI*. The task configurations in this document for tunnel mode apply to SVC and AnyConnect VPN Client. For more information about the Cisco AnyConnect VPN Client feature, see the Cisco AnyConnect VPN Client Administrator Guide, Release 2.4 and the Release Notes for Cisco AnyConnect VPN Client, Release 2.4. **Note** Many of the features listed in the documents *Cisco AnyConnect VPN Client Administrator Guide* and *Release Notes for Cisco AnyConnect VPN Client, Version 2.0* apply only to the Cisco ASA 5500 Series Adaptive Security Appliances. For a list of features that do not currently apply to other Cisco platforms, see the restriction in the Cisco AnyConnect VPN Client, page 3 of this document.

Feature Name	Release	Feature Information
Debug Infrastructure	12.4(11)T	Updates to the **webvpn debug** command provide administrators with the ability to turn debugging on for any one user or group. The following keywords were introduced by this feature: **acl, entry sso, verbose.** The following keyword options were added for the **http** keyword: **authentication**, **trace**, and **verbose**. The **verbose** keyword option was added for the **citrix**, **cookie**, **tunnel**, and **webservice** keywords. The **port-forward** keyword was deleted and the **detail** keyword option for the **tunnel** keyword was deleted.
Front-Door VRF Support	12.4(15)T	Coupled with the already supported internal VRF, this feature allows the SSL VPN gateway to be fully integrated into an MPLS network.
Full-Tunnel CEF Support	12.4(20)T	This feature provides better performance for full-tunnel packets.
GUI Enhancements	12.4(15)T	These enhancements provide updated examples and explanation of the Web VPN GUIs.
Internationalization	12.4(22)T	The Internationalization feature provides multilanguage support for SSL VPN clients, such as Cisco Secure Desktop (CSD) and SSL VPN Client (SVC). The following commands were introduced: **browser-attribute import, import language, webvpn create template.**

Feature Name	Release	Feature Information
Licensing support for Cisco IOS SSL VPNs	15.0(1)M	A license count is associated with each counted license and the count indicates the instances of the feature available for use in the system. In Cisco IOS Release 15.0(1)M, support was added for Cisco 880, Cisco 890, Cisco 1900, Cisco 2900, and Cisco3900 series routers. The following commands were introduced or modified: **debug webvpn license, show webvpn license.**
Max-user limit message	12.4(22)T	This error message is received when a user tries to log in to a Web VPN context and his or her maximum user limit has been reached.
Netegrity Cookie-Based Single SignOn (SSO) Support	12.4(11)T	This feature allows administrators to configure an SSO server that sets a SiteMinder cookie in the browser of a user when the user initially logs in. The benefit of this feature is that users are prompted to log in only a single time. The following commands were modified for this feature: **clear webvpn stats**, **debug webvpn**, **show webvpn context**, **show webvpn policy**, and **show webvpn stats**. The following commands were added for this feature: **max-retry-attempts**, **request-timeout**, **secret-key**, **sso-server**, and **web-agent-url**.
NTLM Authentication	12.4(9)T	This feature provides NT LAN Manager (NTLM) authentication support. The following command was modified by this feature: **functions**.

Feature Name	Release	Feature Information
Port-Forward Enhancements	12.4(11)T	This feature provides administrators with more options for configuring HTTP proxy and portal pages. The following commands were added for this feature: **acl**, **add**, **deny**, **error-msg**, **error-url**, **list**, and **permit**.
RADIUS Accounting	12.4(9)T	This feature provides for RADIUS accounting for SSL VPN sessions. The following command was added by this feature: **webvpn aaa accounting-list**.
SSL VPN	12.4(6)T	This feature enhances SSL VPN support in Cisco IOS software. This feature provides a comprehensive solution that allows easy access to a broad range of web resources and web-enabled applications using native HTTP over SSL (HTTPS) browser support. SSL VPN introduced three modes of SSL VPN access: clientless, thin-client, and full-tunnel client support. The following command was introduced in Cisco IOS Release 12.4(15)T: **cifs-url-list**.
SSL VPN Client-Side Certificate-Based Authentication	15.0(1)M	This feature enables SSL VPN to authenticate clients based on the client's AAA username and password and also supports webvpn gateway authentication of clients using AAA certificates. The following command was modified by this feature: **authentication certificate**, **ca trustpoint**, **match-certificate**, **svc profile**, **username-prefill**, **webvpn import svc profile**.

Feature Name	Release	Feature Information
SSLVPN DVTI Support	15.1(1)T	The SSLVPN DVTI Support feature adds DVTI support to the SSLVPN and hence enables seamless interoperability with IP features such as firewalls, NAT, ACL, and VRF. This feature also provides DVTI support, which allows the configuration of IP features on a per-tunnel basis. The following command was introduced or modified: **virtual-template**.
SSL VPN Phase-4 Features	15.1(1)T	The SSL VPN Phase-4 Features feature provides the following enhancements to the Cisco IOS SSL VPN: • ACL support for split tunneling • IP mask for IP pool address assignment • Undoing the renaming of AnyConnect or SVC Full Tunnel Cisco package during installation on a Cisco IOS router • Adding per-user SSL VPN session statistics • Start Before Logon option for the Cisco IOS SSL VPN headend The following commands were introduced or modified: **show webvpn session**, **svc address-pool**, **svc module**, **svc split**.
DTLS Support for IOS SSL VPN	15.1(2)T	The DTLS Support for IOS SSL VPN feature enables DTLS as a transport protocol for the traffic tunneled through SSL VPN. The following commands were introduced or modified: **debug webvpn dtls**, **dtls port**, **svc dtls**.

Feature Name	Release	Feature Information
Stateless High Availability with Hot Standby Router Protocol (HSRP)	12.4(20)T	This feature allows stateless failover to be applied to VPN routers by using HSRP. The following command was modified by this feature: **ip address**.
URL Obfuscation	12.4(11)T	This feature provides administrators with the ability to obfuscate, or mask, sensitive portions of an enterprise URL, such as IP addresses, hostnames, or port numbers. The following command was added by this feature: **mask-urls**.
URL Rewrite Splitter	12.4(20)T	This feature allows administrators to selectively mangle requests to the gateway. The following commands were added by this feature: **host**, **ip**, **unmatched-action**, and **url rewrite**.
User-Level Bookmarking	12.4(15)T	This feature allows a user to bookmark URLs while connected through an SSL VPN tunnel. The following command was added by this feature: **user-profile location**.
Virtual Templates	12.4(24)T1	A virtual template enables SSL VPN to interoperate with IP features such as NAT, firewall, and policy-based routing. The following command was introduced: **virtual-template**.

Notices

The following notices pertain to this software license.

OpenSSL Project

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

- License Issues, page 151

License Issues

The OpenSSL toolkit stays under a dual license; that is, both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.
4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

“This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS”' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

"This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)".

The word 'cryptographic' can be left out if the routines from the library being used are not cryptography-related.

1. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)".

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed; that is, this code cannot simply be copied and put under another distribution license [including the GNU Public License].