AnyConnect VPN Client Connections

This section describes how to configure AnyConnect VPN Client Connections.

About the Secure Client VPN Client

The Secure Client provides secure SSL and IPsec/IKEv2 connections to the ASA for remote users. Without a previously-installed client, remote users enter the IP address in their browser of an interface configured to accept SSL or IPsec/IKEv2 VPN connections. Unless the ASA is configured to redirect http:// requests to https://, users must enter the URL in the form https://<*address*>.

After entering the URL, the browser connects to that interface and displays the login screen. If the user satisfies the login and authentication, and the ASA identifies the user as requiring the client, it downloads the client that matches the operating system of the remote computer. After downloading, the client installs and configures itself, establishes a secure SSL or IPsec/IKEv2 connection and either remains or uninstalls itself (depending on the configuration) when the connection terminates.

In the case of a previously installed client, when the user authenticates, the ASA examines the revision of the client, and upgrades the client as necessary.

When the client negotiates an SSL VPN connection with the ASA, it connects using Transport Layer Security (TLS), and optionally, Datagram Transport Layer Security (DTLS). DTLS avoids latency and bandwidth problems associated with some SSL connections and improves the performance of real-time applications that are sensitive to packet delays.

The Secure Client can be downloaded from the ASA, or it can be installed manually on the remote PC by the system administrator. For more information about installing the client manually, see the appropriate release of the Cisco AnyConnect Secure Mobility Configuration Guide .

The ASA downloads the client based on the group policy or username attributes of the user establishing the connection. You can configure the ASA to automatically download the client, or you can configure it to

prompt the remote user about whether to download the client. In the latter case, if the user does not respond, you can configure the ASA to either download the client after a timeout period or present the login page.

Requirements for Secure Client

For the requirements of endpoint computers running the Secure Client, see the appropriate release of the Cisco AnyConnect Secure Mobility Release Notes.

Guidelines and Limitations for Secure Client

- The ASA does not verify remote HTTPS certificates.
- Supported in single or multiple context mode. AnyConnect Apex license is required for remote-access VPN in multi-context mode. Although ASA does not specifically recognize an AnyConnect Apex license, it enforces licenses characteristics of an Apex license such as AnyConnect Premium licensed to the platform limit, Secure Client for mobile, Secure Client for Cisco VPN phone, and advanced endpoint assessment. Shared licensing, AnyConnect Essentials, failover license aggregation, and flex/time-based licenses are not supported.
- Issuing commands such as **curl** against the RA VPN headend is not directly supported, and might not have desirable results. For example, the headend does not respond to HTTP HEAD requests.
- When hardware VPN phones such as the Cisco 88xx series use Secure Client, they can experience a reconnection despite having DTLS up and Dead Peer Detection (DPD) configured.
- When a client connects to Secure Client, the IP address of the client before and after the connection changes. ASA supports this behavior.

Licensing Requirements for Secure Client

Note This feature is not available on No Payload Encryption models.

VPN Licenses require an AnyConnect Plus or Apex license, available separately. See Cisco ASA Series Feature Licenses for maximum values per model.

If you start a clientless SSL VPN session and then start the Secure Client session from the portal, 1 session is used in total. However, if you start the Secure Client first (from a standalone client, for example) and then log into the clientless SSL VPN portal, then 2 sessions are used.

Configure Secure Client Connections

This section describes prerequisites, restrictions, and detailed tasks to configure the ASA to accept AnyConnect VPN client connections.

Configure the ASA to Web-Deploy the Client

The section describes the steps to configure the ASA to web-deploy the Secure Client.

Before you begin

Copy the client image package to the ASA using TFTP or another method.

Note Even though the clientless VPN feature is disabled on ASA, when you use a web browser to access AnyConnect webdeploy (https://x.x.x.x<ASA IP address>, the VPN session on the ASA is counted as clientless.

Procedure

Step 1 Identify a file on flash as the Secure Client package file.

The ASA expands the file in cache memory for downloading to remote PCs. If you have multiple clients, assign an order to the client images with the order argument.

The ASA downloads portions of each client in the order you specify until it matches the operating system of the remote PC. Therefore, assign the lowest number to the image used by the most commonly-encountered operating system.

anyconnect image *filename order*

Example:

```
hostname(config-webvpn)# anyconnect image
anyconnect-win-2.3.0254-k9.pkg 1
hostname(config-webvpn)# anyconnect image
anyconnect-macosx-i386-2.3.0254-k9.pkg 2
hostname(config-webvpn)# anyconnect image
anyconnect-linux-2.3.0254-k9.pkg 3
```

Note You must issue the **anyconnect enable** command after configuring the Secure Client images with the **anyconnect image** command. If you do not enable Secure Client, it will not operate as expected, and **show webvpn anyconnect** considers the SSL VPN client as not enabled rather than listing the installed Secure Client packages.

Step 2 Enable SSL on an interface for clientless or Secure Client SSL connections.

enable *interface*

Example:

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

Step 3 Without issuing this command, Secure Client does not function as expected, and a **show webvpn anyconnect** command returns that the "SSL VPN is not enabled," instead of listing the installed Secure Client packages.

anyconnect enable

Step 4 (Optional) Create an address pool. You can use another method of address assignment, such as DHCP and/or user-assigned addressing.

ip local pool *poolname startaddr-endaddr* **mask** *mask*

Example:

```
hostname(config)# ip local pool vpn_users 209.165.200.225-209.165.200.254
mask 255.255.255.224
```

Step 5 Assign an address pool to a tunnel group.

address-pool *poolname*

Example:

```
hostname(config)# tunnel-group telecommuters general-attributes
hostname(config-tunnel-general)# address-pool vpn_users
```

Step 6 Assign a default group policy to the tunnel group.

default-group-policy *name*

```
hostname(config-tunnel-general)# default-group-policy sales
```

Step 7 Enable the display of the tunnel-group list on the clientless portal and Secure Client GUI login page. The list of aliases is defined by the *group-alias name enable* command.

group-alias *name* **enable**

Example:

```
hostname(config)# tunnel-group telecommuters webvpn-attributes
hostname(config-tunnel-webvpn)# group-alias sales_department enable
```

Step 8 Specify the Secure Clients as a permitted VPN tunneling protocol for the group or user.

tunnel-group-list enable

Example:

```
hostname(config)# webvpn
hostname(config-webvpn)# tunnel-group-list enable
```

Step 9 Specify SSL as a permitted VPN tunneling protocol for the group or user. You can also specify additional protocols. For more information, see the vpn-tunnel-protocol command in the command reference.

vpn-tunnel-protocol

Example:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# vpn-tunnel-protocol
```

What to do next

For more information about assigning users to group policies, see Chapter 6, Configuring Connection Profiles, Group Policies, and Users.

Enable Permanent Client Installation

Enabling permanent client installation disables the automatic uninstalling feature of the client. The client remains installed on the remote computer for subsequent connections, reducing the connection time for the remote user.

To enable permanent client installation for a specific group or user, use the anyconnect keep-installer command from group-policy or username webvpn modes.

The default is that permanent installation of the client is enabled. The client remains on the remote computer at the end of the session. The following example configures the existing group-policy *sales* to remove the client on the remote computer at the end of the session:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-policy)# anyconnect keep-installer installed none
```

Configure DTLS

Datagram Transport Layer Security (DTLS) allows the Secure Client establishing an SSL VPN connection to use two simultaneous tunnels—an SSL tunnel and a DTLS tunnel. Using DTLS avoids latency and bandwidth problems associated with SSL connections and improves the performance of real-time applications that are sensitive to packet delays.

Before you begin

See, Configure Advanced SSL Settings to configure DTLS on this headend, and which version of DTLS is used.

In order for DTLS to fall back to a TLS connection, Dead Peer Detection (DPD) must be enabled. If you do not enable DPD, and the DTLS connection experiences a problem, the connection terminates instead of falling back to TLS. For more information on DPD, see Configure Dead Peer Detection, on page 16.

Procedure

Step 1 Specify DTLS options for Secure Client VPN connections:

a) Enable SSL and DTLS on the interface in webvpn mode.

By default, DTLS is enabled when SSL VPN access is enabled on an interface.

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

Disable DTLS for all Secure Client users with the **enable** *interface* **tls-only** command in webvpn configuration mode.

If you disable DTLS, SSL VPN connections connect with an SSL VPN tunnel only.

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

b) Configure the ports for SSL and DTLS using the **port** and **dtls port** commands.

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

```
hostname(config-webvpn)# port 555
hostname(config-webvpn)# dtls port 556
```

Step 2 Specify DTLS options for specific group policies.

a) Enable DTLS for specific groups or users with the **anyconnect ssl dtls** command in group policy webvpn or username webvpn configuration mode.

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect ssl dtls enable
```

b) If desired, enable DTLS compression using the anyconnect dtls compression command.

```
hostname(config-group-webvpn)# anyconnect dtls compression lzs
```

Prompt Remote Users

Procedure

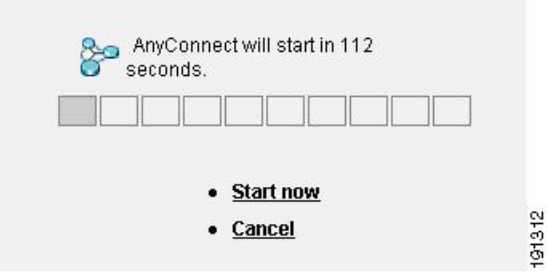
You can enable the ASA to prompt remote SSL VPN client users to download the client with the **anyconnect ask** command from group policy webvpn or username webvpn configuration modes:

[**no**] **anyconnect ask** {**none** | **enable** [**default** {**webvpn** | } **timeout** *value*]}

- **anyconnect enable** prompts the remote user to download the client or go to the clientless portal page and waits indefinitely for user response.
- **anyconnect ask enable default** immediately downloads the client.
- **anyconnect ask enable default webvpn** immediately goes to the portal page.
- **anyconnect ask enable default timeout** value prompts the remote user to download the client or go to the clientless portal page and waits the duration of *value* before taking the default action—downloading the client.
- **anyconnect ask enable default clientless timeout** value prompts the remote user to download the client or go to the clientless portal page, and waits the duration of *value* before taking the default action—displaying the clientless portal page.

The figure below shows the prompt displayed to remote users when either **default anyconnect timeout** *value or* **default webvpn timeout** *value* is configured:

Figure 1: Prompt Displayed to Remote Users for SSL VPN Client Download



Example

The following example configures the ASA to prompt the user to download the client or go to the clientless portal page and wait *10 seconds for a response* before downloading the client:

```
hostname(config-group-webvpn)# anyconnect ask enable default anyconnect timeout
 10
```

Enable Secure Client Profile Downloads

You enable Secure Client features in the Secure Client profiles—XML files that contain configuration settings for the core client with its VPN functionality and for the optional client modules. The ASA deploys the profiles during Secure Client installation and updates. Users cannot manage or modify profiles.

The file downloaded to the client is of the format: <*profile_name*>.xml.

You can configure a profile using the Secure Client profile editor, a convenient GUI-based configuration tool launched from ASDM or ISE. The Secure Client software package for Windows includes the editor, which activates when you load the client package on the chosen headend device and specify it as an Secure Client image.

We also provide a standalone version of the profile editor for Windows that you can use as an alternative to the profile editor integrated with ASDM or ISE. If you are predeploying the client, you can use the standalone profile editor to create profiles for the VPN service and other modules that you deploy to computers using your software management system.

For more information on the Secure Client and its Profile Editor, see the appropriate release of the Cisco AnyConnect Secure Mobility Configuration Guide .

Note The Secure Client protocol defaults to SSL. To enable IPsec IKEv2, you must configure the IKEv2 settings on the ASA and also configure IKEv2 as the primary protocol in the client profile. The IKEv2enabled profile must be deployed to the endpoint computer; otherwise the client attempts to connect using SSL.

Procedure

Step 1 Use the profile editor from ASDM/ISE or the standalone profile editor to create a profile.

Step 2 Load the profile file into flash memory on the ASA using tftp or another method.

Step 3 Use the **anyconnect profiles** command from webvpn configuration mode to identify the file as a client profile to load into cache memory.

Example:

The following example specifies the files sales_hosts.xml and engineering_hosts.xml as profiles:

```
asa1(config-webvpn)# anyconnect profiles sales
disk0:/sales_hosts.xml
asa1(config-webvpn)# anyconnect profiles engineering
disk0:/engineering_hosts.xml
```

The profiles are now available to group policies.

View the profiles loaded in cache memory using the **dir cache:stc/profiles** command:

```
hostname(config-webvpn)# dir cache:/stc/profiles

Directory of cache:stc/profiles/

0      ----  774         11:54:41 Nov 22 2006  engineering.xml
0      ----  774         11:54:29 Nov 22 2006  sales.xml

2428928 bytes total (18219008 bytes free)
hostname(config-webvpn)#
```

Step 4 Enter group policy webvpn configuration mode and specify a client profile for a group policy with the **anyconnect profiles** command:

Example:

You can enter the client profiles value command followed by a question mark (**?**) to view the available profiles. For example:

```
asa1(config-group-webvpn)# anyconnect profiles value ?

config-group-webvpn mode commands/options:
Available configured profile packages: engineering sales
```

The next example configures the group policy to use the profile *sales* with the client profile type *vpn*:

```
asa1(config-group-webvpn)# anyconnect profiles value sales type vpn
asa1(config-group-webvpn)#
```

Enable Secure Client Deferred Upgrade

Deferred Upgrade allows the Secure Client user to delay download of a client upgrade. When a client update is available, Secure Client opens a dialog asking the user if they would like to update, or to defer the upgrade. This upgrade dialog will not appear unless you have AutoUpdate set to *Enabled* in the Secure Client profile setting.

Deferred Upgrade is enabled by adding custom attribute types and named values to the ASA; then referencing and configuring those attributes in a group policy.

The following custom attributes support Deferred Upgrade:

Table 1: Custom Attributes for Deferred Upgrade

Custom Attribute Type	Valid Values	Default Value	Notes
DeferredUpdateAllowed	true false	false	True enables deferred update. If deferred update is disabled (false), the settings below are ignored.
DeferredUpdateMinimumVersion	x.y.z	0.0.0	Minimum version of Secure Client that must be installed for updates to be deferrable. The minimum version check applies to all modules enabled on the headend. If any enabled module (including VPN) is not installed or does not meet the minimum version, then the connection is not eligible for deferred update. If this attribute is not specified, then a deferral prompt is displayed (or auto-dismissed) regardless of the version installed on the endpoint.
DeferredUpdateDismissTimeout	0-300 (seconds)	none (disabled)	Number of seconds that the deferred upgrade prompt is displayed before being dismissed automatically. This attribute only applies when a deferred update prompt is to be displayed (the minimum version attribute is evaluated first). If this attribute is missing, then the auto-dismiss feature is disabled, and a dialog is displayed (if required) until the user responds. Setting this attribute to zero allows automatic deferral or upgrade to be forced based on: • The installed version and the value of DeferredUpdateMinimumVersion. • The value of DeferredUpdateDismissResponse.
DeferredUpdateDismissResponse	defer update	update	Action to take when DeferredUpdateDismissTimeout occurs.

Procedure

Step 1 Create the custom attribute types with the **anyconnnect-custom-attr** command in webvpn configuration mode:

[**no**] **anyconnect-custom-attr** *attr-type* [**description** *description*]

Example:

The following example shows how to add the custom attribute types DeferredUpdateAllowed and DeferredUpdateDismissTimeout:

```
hostame(config-webvpn)# anyconnect-custom-attr DeferredUpdateAllowed
description Indicates if the deferred update feature is enabled or not
hostame(config-webvpn)# anyconnect-custom-attr DeferredUpdateDismissTimeout
```

Step 2 Add named values for custom attributes with the **anyconnect-custom-data** command in global configuration mode. For attributes with long values, you can provide a duplicate entry, and it allows concatenation. However, with a duplicate configuration entry, the Defer Update dialog will not appear, and a user cannot defer the upgrade; instead, the upgrade happens automatically.

[**no**] **anyconnect-custom-data** *attr-type attr-name attr-value*

Example:

The following example shows how to add a named value for the custom attribute type DeferredUpdateDismissTimeout and for enabling DeferredUpdateAllowed:

```
hostname(config)# anyconnect-custom-data DeferredUpdateDismissTimeout
def-timeout 150
hostname(config)# anyconnect-custom-data DeferredUpdateAllowed
def-allowed true
```

Step 3 Add or remove the custom attribute named values to a group policy using the **anyconnect-custom** command:

- **anyconnect-custom** *attr-type* **value** *attr-name*
- **anyconnect-custom** *attr-type* **none**
- **no anyconnect-custom** *attr-type*

Example:

The following example shows how to enable Deferred Update for the group policy named sales and set the timeout to 150 seconds:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# anyconnect-custom DeferredUpdateAllowed
value def-allowed
hostname(config-group-policy)# anyconnect-custom DeferredUpdateDismissTimeout
value def-timeout
```

Enable DSCP Preservation

By setting another custom attribute, you can control Differentiated Services Code Point (DSCP) on Windows or OS X platforms for DTLS connections only. Enabling DSCP preservation allows devices to prioritize latency sensitive traffic; the router takes into account whether this is set and marks prioritized traffic to improve outbound connection quality.

Procedure

Step 1 Create the custom attribute types with the **anyconnect-custom-attr**command in webvpn configuration mode:

[no] anyconnect-custom-attr DSCPPreservationAllowed description Set to control Differentiated Services Code Point (DSCP) on Windows or OS X platforms for DTLS connections only.

Step 2 Add named values for custom attributes with the **anyconnect-custom-data** command in global configuration mode:

[no] anyconnect-custom-data DSCPPreservationAllowed true

Note By default, Secure Client performs DSCP preservation (true). To disable it, set the custom attributes to false on the headend and reinitiate the connection.

Enable Additional Secure Client Features

To minimize download time, the client only requests downloads (from the ASA or ISE) of the core modules that it needs. As additional features become available for the Secure Client, you need to update the remote clients in order for them to use the features.

To enable new features, you must specify the new module names using the **anyconnect modules** command from group policy webvpn or username webvpn configuration mode:

[**no**]**anyconnect modules** {**none** | **value** *string*}

Separate multiple strings with commas.

Enable Start Before Logon

Start Before Logon (SBL) allows login scripts, password caching, drive mapping, and more, for the Secure Client installed on a Windows PC. For SBL, you must enable the ASA to download the module which enables graphical identification and authentication (GINA) for the Secure Client. The following procedure shows how to enable SBL:

Procedure

Step 1 Enable the ASA to download the GINA module for VPN connection to specific groups or users using the **anyconnect modules** *vpngina* command from group policy webvpn or username webvpn configuration modes.

Example:

In the following example, the user enters group-policy attributes mode for the group policy *telecommuters*, enters webvpn configuration mode for the group policy, and specifies the string *vpngina*:

```
hostname(config)# group-policy telecommuters attributes
hostname(config-group-policy)# webvpn
hostame(config-group-webvpn)#anyconnect modules value vpngina
```

Step 2 Retrieve a copy of the client profiles file (AnyConnectProfile.tmpl).

Step 3 Edit the profiles file to specify that SBL is enabled. The example below shows the relevant portion of the profiles file (AnyConnectProfile.tmpl) for Windows:

```
<Configuration>
    <ClientInitialization>
        <UseStartBeforeLogon>false</UseStartBeforeLogon>
```

```
    </ClientInitialization>
```

The <UseStartBeforeLogon> tag determines whether the client uses SBL. To turn SBL on, replace *false* with *true*. The example below shows the tag with SBL turned on:

```
<ClientInitialization>
        <UseStartBeforeLogon>true</UseStartBeforeLogon>
    </ClientInitialization>
```

Step 4 Save the changes to AnyConnectProfile.tmpl and update the profile file for the group or user on the ASA using the **profile** command from webvpn configuration mode. For example:

```
asa1(config-webvpn)#anyconnect profiles sales disk0:/sales_hosts.xml
```

Translating Languages for Secure Client User Messages

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

This section describes how to configure the ASA to translate these user messages.

Understand Language Translation

Functional areas and their messages that are visible to remote users are organized into translation domains. All messages displayed on the user interface of the Cisco AnyConnect VPN Client are located in the Secure Client domain.

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

You can also export an existing translation table. The XML file created displays the messages you edited previously. Reimporting this XML file with the same language name creates an new version of the translation table object, overwriting previous messages. Changes to the translation table for the Secure Client domain are immediately visible to Secure Client users.

Create Translation Tables

The following procedure describes how to create translation tables for the Secure Client domain:

Procedure

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

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

```
hostname# show import webvpn translation-table
Translation Tables' Templates:
```

```
customization
AnyConnect

PortForwarder
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
```

Then the user exports the translation table for the Secure Client translation domain. The filename of the XML file created is named *client* and contains empty message fields:

```
hostname# export webvpn translation-table AnyConnect
template tftp://209.165.200.225/client
```

In the next example, the user exports a translation table named *zh*, which was previously imported from a template. zh is the abbreviation by Microsoft Internet Explorer for the Chinese language.

```
hostname# export webvpn translation-table customization
language zh tftp://209.165.200.225/chinese_client
```

Step 2 Edit the Translation Table XML file. The following example shows a portion of the Secure Client template. The end of this output includes a message ID field (msgid) and a message string field (msgstr) for the message *Connected*, which is displayed on the Secure Client GUI when the client establishes a VPN connection. The complete template contains many pairs of message fields:

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2006-11-01 16:39-0700\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"

#: C:\cygwin\home\<user>\cvc\main\Api\AgentIfc.cpp:23
#: C:\cygwin\home\<user>\cvc\main\Api\check\AgentIfc.cpp:22
#: C:\cygwin\home\<user>\cvc\main\Api\save\AgentIfc.cpp:23
#: C:\cygwin\home\<user>\cvc\main\Api\save\AgentIfc.cpp~:20
#: C:\cygwin\home\<user>\cvc\main\Api\save\older\AgentIfc.cpp:22
msgid "Connected"
msgstr ""
```

The msgid contains the default translation. The msgstr that follows msgid provides the translation. To create a translation, enter the translated text between the quotes of the msgstr string. For example, to translate the message “Connected” with a Spanish translation, insert the Spanish text between the quotes:

```
msgid "Connected"
msgstr "Conectado"
```

Be sure to save the file.

Step 3 Import the translation table using the **import webvpn translation-table** command from privileged EXEC mode. Be sure to specify the name of the new translation table with the abbreviation for the language that is compatible with the browser.

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

```
hostname# import webvpn translation-table AnyConnect
language es-us tftp://209.165.200.225/client
hostname# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
hostname# show import webvpn translation-table
Translation Tables' Templates:
AnyConnect
PortForwarder

customization
keepout
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
es-us AnyConnect
```

Remove Translation Tables

If you no longer need a translation table, you can remove it.

Procedure

Step 1 List the existing translation tables.

In the following example, the **show import webvpn translation-table** command shows available translation table templates and tables. Various tables are available for French (fr), Japanese (ja), and Russian (ru).

```
hostname# show import webvpn translation-table
Translation Tables' Templates:
  AnyConnect
  PortForwarder
  banners
  csd
  customization
```

```
  url-list
  webvpn
Translation Tables:
  fr                  PortForwarder
  fr                  AnyConnect
  fr                  customization
  fr                  webvpn
  ja                  PortForwarder
  ja                  AnyConnect
  ja                  customization
  ja                  webvpn
  ru                  PortForwarder
  ru                  customization
  ru                  webvpn
```

Step 2 Remove the unwanted translation table.

revert webvpn translation-table *translationdomain* **language** *language*

Where *translationdomain* is the domain listed on the right in the Translation Tables listing shown above, and *language* is the 2-character language name.

You must remove each table individually. You cannot remove all of the tables for a given language with one command.

For example, to remove the French translation table for Secure Client:

```
ciscoasa# revert webvpn translation-table anyconnect language fr
ciscoasa#
```

Configuring Advanced Secure Client SSL Features

The following section describes advanced features that fine-tune Secure Client SSL VPN connections.

Enable Rekey

When the ASA and the Secure Client perform a rekey on an SSL VPN connection, they renegotiate the crypto keys and initialization vectors, increasing the security of the connection.

To enable the client to perform a rekey on an SSL VPN connection for a specific group or user, use the **anyconnect ssl** rekey command from group-policy or username webvpn modes.

[no]anyconnect ssl rekey {**method** {**new-tunnel** | **none** | **ssl**} | **time** *minutes*}

- **method new-tunnel** specifies that the client establishes a new tunnel during rekey.
- **method ssl** specifies that the client establishes a new tunnel during rekey.
- **method none** disables rekey.
- **time** *minutes* specifies the number of minutes from the start of the session, or from the last rekey, until the rekey takes place, from 1 to 10080 (1 week).

Note Configuring the rekey method as **ssl** or **new-tunnel** specifies that the client establishes a new tunnel during rekey instead of the SSL renegotiation taking place during the rekey. See the command reference for a history of the **anyconnect ssl rekey** command.

In the following example, the client is configured to renegotiate with SSL during rekey, which takes place 30 minutes after the session begins, for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect ssl rekey method ssl
hostname(config-group-webvpn)# anyconnect ssl rekey time 30
```

Configure Dead Peer Detection

Dead Peer Detection (DPD) ensures that the ASA (gateway) or the client can quickly detect a condition where the peer is not responding, and the connection has failed. To enable dead peer detection (DPD) and set the frequency with which either the Secure Client or the ASA gateway performs DPD, do the following:

Before you begin

- This feature applies to connectivity between the ASA gateway and the Secure Client SSL VPN Client only. It does not work with IPsec since DPD is based on the standards implementation that does not allow padding.
- If you enable DTLS, enable Dead Peer Detection (DPD) also. DPD enables a failed DTLS connection to fallback to TLS. Otherwise, the connection terminates.
- When DPD is enabled on the ASA, you can use the Optimal MTU (OMTU) function to find the largest endpoint MTU at which the client can successfully pass DTLS packets. Implement OMTU by sending a padded DPD packet to the maximum MTU. If a correct echo of the payload is received from the head end, the MTU size is accepted. Otherwise, the MTU is reduced, and the probe is sent again until the minimum MTU allowed for the protocol is reached.

Procedure

Step 1 Go to the desired group policy.

Enter group policy or username webvpn mode:

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

Or,

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

Step 2 Set Gateway Side Detection.

Use the [**no**] **anyconnect dpd-interval** {[**gateway** {*seconds* | **none**}] command.

The gateway refers to the ASA. You enable DPD and specify the interval with which the ASA waits for any packet from the client as a range of from 30 (default) to 3600 seconds (1 hour). A value of 300 is recommended. If no packets are received within that interval, the ASA performs the DPD test with three attempts at the same interval. If the ASA does not receive a response from the client, it tears down the TLS/DTLS tunnel.

Note Specifying **none** disables the DPD testing that the ASA performs. Use **no anyconnect dpd-interval** to remove this command from the configuration.

Specifying **none** disables the DPD testing that the ASA performs. Use **no anyconnect dpd-interval** to remove this command from the configuration.

Step 3 Set Client Side Detection.

Use the [**no**] **anyconnect dpd-interval** {[**client** {*seconds* | **none**}]} command.

The client refers to the Secure Client. You enable DPD and specify the frequency with which the client performs the DPD test as a range of from 30 (default) to 3600 seconds (1 hour). A value of 30 seconds is recommended.

Specifying **client none** disables DPD performed by the client. Use **no anyconnect dpd-interval** to remove this command from the configuration.

Example

The following example sets the frequency of DPD performed by the ASA to 30 seconds, and the frequency of DPD performed by the client set to 10 seconds for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect dpd-interval gateway 30
hostname(config-group-webvpn)# anyconnect dpd-interval client 10
```

Enable Keepalive

You can adjust the frequency of keepalive messages to ensure that an SSL VPN connection through a proxy, firewall, or NAT device remains open, even if the device limits the time that the connection can be idle. Adjusting the frequency also ensures that the client 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.

Keepalives are enabled by default. If you disable keepalives, in the event of a failover, SSL VPN client sessions are not carried over to the standby device.

To set the frequency of keepalive messages, use the **keepalive** command from group-policy webvpn or username webvpn configuration mode: Use the **no** form of the command to remove the command from the configuration and cause the value to be inherited:

[**no**] **anyconnect ssl keepalive** {**none** | *seconds*}

- **none** disables client keepalive messages.
- *seconds* enables the client to send keepalive messages, and specifies the frequency of the messages in the range of 15 to 600 seconds.

In the following example, the ASA is configured to enable the client to send keepalive messages with a frequency of 300 seconds (5 minutes), for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect ssl keepalive 300
```

Use Compression

Compression increases the communications performance between the ASA and the client by reducing the size of the packets being transferred for low-bandwidth connections. By default, compression for all SSL VPN connections is enabled on the ASA, both at the global level and for specific groups or users.

Note When implementing compression on broadband connections, you must carefully consider the fact that compression relies on loss-less connectivity. This is the main reason that it is not enabled by default on broadband connections.

Compression must be turned-on globally using the **compression** command from global configuration mode, and then it can be set for specific groups or users with the **anyconnect ssl compression** command in group-policy and username webvpn modes.

Changing Compression Globally

To change the global compression settings, use the anyconnect ssl **compression** command from global configuration mode. To remove the command from the configuration, use the **no** form of the command.

In the following example, compression is disabled for all SSL VPN connections globally:

```
hostname(config)# no compression
```

Changing Compression for Groups and Users

To change compression for a specific group or user, use the anyconnect ssl compression command in the group-policy and username webvpn modes:

[**no**] **anyconnect ssl compression** {**deflate** | **none**}

By default, for groups and users, SSL compression is set to *deflate* (enabled).

To remove the **anyconnect ssl compression** command from the configuration and cause the value to be inherited from the global setting, use the **no** form of the command:

In the following example, compression is disabled for the group-policy sales:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# no anyconnect ssl compression none
```

Adjust MTU Size

You can adjust the MTU size (from 576 to 1406 bytes) for SSL VPN connections established by the client with the **anyconnect mtu** command from group policy webvpn or username webvpn configuration mode:

[**no**] **anyconnect mtu** *size*

This command affects only the Secure Client. The legacy Cisco SSL VPN Client () is not capable of adjusting to different MTU sizes. Also, client connections established in SSL and those established in SSL with DTLS are impacted by this command.

The default for this command in the default group policy is **no anyconnect mtu**. The MTU size is adjusted automatically based on the MTU of the interface that the connection uses, minus the IP/UDP/DTLS overhead.

You may receive an "MTU configuration sent from the secure gateway is too small" message, for example, when running the ISE Posture AnyConnect module. If you enter **anyconnect mtu 1200** along with **anyconnect ssl df-bit-ignore disable**, you can avoid these system scan errors.

Example

The following example configures the MTU size to 1200 bytes for the group policy telecommuters:

```
hostname(config)# group-policy telecommuters attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect mtu 1200
```

Update Secure Client Images

You can update the client images on the ASA at any time using the following procedure:

Procedure

Step 1 Copy the new client images to the ASA using the **copy** command from privileged EXEC mode, or using another method.

Step 2 If the new client image files have the same filenames as the files already loaded, reenter the **anyconnect image** command that is in the configuration. If the new filenames are different, uninstall the old files using the [**no**]**anyconnect image**image command. Then use the **anyconnect image** command to assign an order to the images and cause the ASA to load the new images.

Enable IPv6 VPN Access

If you want to configure IPv6 access, you must use the command-line interface. Release 9.0(x) of the ASA adds support for IPv6 VPN connections to its outside interface using SSL and IKEv2/IPsec protocols.

You enable IPv6 access using the **ipv6 enable** command as part of enabling SSL VPN connections. The following is an example for an IPv6 connection that enables IPv6 on the outside interface:

```
hostname(config)# interface GigabitEthernet0/0
hostname(config-if)# ipv6 enable
```

To enable IPV6 SSL VPN, do the following general actions:

1. Enable IPv6 on the outside interface.
2. Enable IPv6 and an IPv6 address on the inside interface.
3. Configure an IPv6 address local pool for client assigned IP Addresses.

4. Configure an IPv6 tunnel default gateway.

Procedure

Step 1 Configure Interfaces:

```
interface GigabitEthernet0/0
 nameif outside
 security-level 0
 ip address 192.168.0.1 255.255.255.0
 ipv6 enable      ; Needed for IPv6.
!
interface GigabitEthernet0/1
 nameif inside
 security-level 100
 ip address 10.10.0.1 255.255.0.0
 ipv6 address 2001:DB8::1/32        ; Needed for IPv6.
 ipv6 enable      ; Needed for IPv6.
```

Step 2 Configure an 'ipv6 local pool' (used for IPv6 address assignment):

```
ipv6 local pool ipv6pool 2001:DB8:1:1::5/32 100     ; Use your IPv6 prefix here
```

Note You can configure the ASA to assign an IPv4 address, an IPv6 address, or both an IPv4 and an IPv6 address to the Secure Client by creating internal pools of addresses on the ASA or by assigning a dedicated address to a local user on the ASA.

Step 3 Add the ipv6 address pool to your tunnel group policy (or group-policy):

```
tunnel-group YourTunGrp1 general-attributes  ipv6-address-pool ipv6pool
```

Note You must also configure an IPv4 address pool here as well (using the 'address-pool' command).

Step 4 Configure an IPv6 tunnel default gateway:

```
ipv6 route inside ::/0 X:X:X:X::X tunneled
```

SAML 2.0

The ASA supports SAML 2.0 so that the VPN end users will be able to input their credentials only one time when they switch between SAAS applications outside of the private network.

For instance, an enterprise customer has enabled PingIdentity as their SAML Identity Provider (IdP) and has accounts on Rally, Salesforce, Oracle OEM, Microsoft ADFS, onelogin, or Dropbox which have been SAML 2.0 SSO enabled. When you configure the ASA to support SAML 2.0 SSO as a Service Provider (SP), end users are able to sign in once and have access to all these services.

AnyConnect SAML support was added to allow an AnyConnect 4.4 client to access SAAS-based applications using SAML 2.0. AnyConnect 4.6 introduced an enhanced version of SAML integration with an embedded browser which replaced the native (external) browser integration from previous releases. The new enhanced

version with embedded browser required you to upgrade to AnyConnect 4.6 (or later) and ASA 9.7.1.24 (or later), 9.8.2.28 (or later), or 9.9.2.1 (or later).

ASA release 9.17.1/ASDM release 7.17.1 introduced support for AnyConnect VPN SAML external browser with AnyConnect 4.10.04065 (or later). When you use SAML as the primary authentication method for the AnyConnect VPN connection profile, you can choose for the Secure Client to use a local browser, instead of the Secure Client embedded browser, when performing web authentication. With this feature, Secure Client supports WebAuthN and any other SAML-based web authentication options, such as Single Sign On, biometric authentication, or other enhanced methods that are unavailable with the embedded browser. For SAML external browser use, you must perform the configuration described here: Configure Default OS Browser for SAML Authentication, on page 27.

The ASA is SP enabled when SAML is configured as the authentication method for a tunnel group, the default tunnel group or any other. The VPN user initiates Single sign-on by accessing an enabled ASA or the SAML IdP. Each of these scenarios is described below.

SAML SP-initiated SSO

When the end user initiates login by accessing the ASA, sign-on behavior proceeds as follows:

1. When the VPN user accesses or chooses a SAML enabled tunnel group, the end user will be redirected to the SAML IdP for authentication. The user will be prompted unless the user access the group-url directly, in which case the redirect is silent.

   The ASA generates a SAML Authentication Request, which the browser redirects to the SAML IdP.

2. The IdP challenges the end user for credential and the end user logs in. The entered credentials must satisfy the IdP authentication configuration.

3. The IdP Response is sent back to the browser and posted to the ASA's sign-in URL. The ASA verifies the response to complete the login.

SAML IdP-initiated SSL

When the user initiates login by accessing the IdP, sign-on behavior proceeds as follows:

1. An end user accesses the IdP. The IdP challenges the end user for credentials according to the IdP's authentication configuration. The end user submits credentials and logs in to the IdP.

2. In general, the end user gets a list of SAML enabled services that have been configured with the IdP. The end user chooses the ASA.

3. A SAML response is sent back to the browser, and posted to the ASA sign-in URL. The ASA verifies the response to complete the login.

Circle of Trust

The trust relationship between the ASA and the SAML Identity Provider is established through configured certificates (ASA trustpoints).

The trust relationship between the end user and SAML Identity Provider is established through the authentication configured on IdP.

SAML Timeouts

In SAML assertion, there are NotBefore and NotOnOrAfter as follows: <saml:Conditions NotBefore="2015-03-10T19:47:41Z" NotOnOrAfter="2015-03-10T20:47:41Z">

A SAML timeout configured on the ASA will override NotOnOrAfter if the sum of NotBefore and timeout is earlier than NotOnOrAfter. If NotBefore + timeout is later than NotOnOrAfter, then NotOnOrAfter will take effect.

The timeout should be very short to prevent the assertion from being re-used after the timeout. You must synchronize your ASA's Network Time Protocol (NTP) server with the IdP NTP server in order to use the SAML feature.

Support in Private Network

SAML 2.0-based service provider IdP is supported in a private network. When the SAML IdP is deployed in the private cloud, ASA and other SAML-enabled services are in peer positions, and all in the private network. With the ASA as a gateway between the user and services, authentication on IdP is handled with a restricted anonymous webvpn session, and all traffic between IdP and the user is translated. When the user logs in, the ASA modifies the session with the corresponding attributes and stores the IdP sessions. Then you can use service provider on the private network without entering credentials again.

The SAML IdP *NameID* attribute determines the user's username and is used for authorization, accounting, and VPN session database.

Note You cannot exchange authentication information between private and public networks. If you use the same IdP for both internal and external service providers, you must authenticate separately. Internal-only IdP cannot be used with external services: external-only IdP cannot be used with service providers in the private network.

Guidelines and Limitations for SAML 2.0

- ASA supports the following signatures for SAML authentication:
  - SHA1 with RSA and HMAC
  - SHA2 with RSA and HMAC
- ASA supports SAML 2.0 Redirect-POST binding , which is supported by all SAML IdPs.
- The ASA functions as a SAML SP only. It cannot act as an Identity Provider in gateway mode or peer mode.
- This SAML SSO SP feature is a mutual exclusion authentication method. It cannot be used with AAA and certificate together.
- Features that are based on username/password authentication, certificate authentication, and KCD are not supported. For instance, username/password pre-filling feature, form-based Auto sign-on, Macro Substitution based Auto sign-on, KCD SSO, and so on.
- ASA supports VPN load balancing with AnyConnect SAML authentication.
- While using Safari for SAML authentication, ensure that you have Safari update 14.1.2 or higher.

- ASA administrators need to ensure clock synchronization between the ASA and the SAML IdP for proper handling of authentication assertions and proper timeout behavior.
- ASA administrators have the responsibility to maintain a valid signing certificate on both ASA and IdP considering the following:
  - The IdP signing certificate is mandatory when configuring an IdP on the ASA.
  - The ASA does not do a revocation check on the signing certificate received from the IdP.
- In SAML assertions, there are NotBefore and NotOnOrAfter conditions. The ASA SAML configured **timeout** interacts with these conditions as follows:
  - Timeout overrides NotOnOrAfter if the sum of NotBefore and timeout is earlier than NotOnOrAfter.
  - If NotBefore + timeout is later than NotOnOrAfter, then NotOnOrAfter takes effect.
  - If the NotBefore attribute is absent, the ASA denies the login request. If the NotOnOrAfter attribute is absent and SAML timeout is not set, ASA denies the login request.
- ASA does not work with Duo in a deployment using an internal SAML, which forces the ASA to proxy for the client to authenticate, due to the FQDN change that occurs during challenge/response for Two-factor authentication (push, code, password).
- Untrusted server certificates are not allowed in the embedded browser.
- The embedded browser SAML integration is not supported in CLI or SBL modes.
- SAML authentication established in a web browser is not shared with AnyConnect and vice versa.
- Depending on the configuration, various methods are used when connecting to the headend with the embedded browser. For example, while AnyConnect might prefer an IPv4 connection over an IPv6 connection, the embedded browser might prefer IPv6, or vice versa. Similarly, AnyConnect may fall back to no proxy after trying proxy and getting a failure, while the embedded browser may stop navigation after trying proxy and getting a failure.
- You must synchronize your ASA's Network Time Protocol (NTP) server with the IdP NTP server in order to use the SAML feature.
- The VPN Wizard on ASDM does not currently support SAML configurations.
- You cannot access internal servers with SSO after logging in using an internal IdP.
- The SAML IdP NameID attribute determines the user's username and is used for authorization, accounting, and VPN session database.
- SAML is not supported in the Multicontext mode.
- Multiple attributes received with a SAML assertion are not supported.

Configure a SAML 2.0 Identity Provider (IdP)

Before you begin

Get the Sign-in and Sign-out URLs for your SAML (IdP) provider. You can get the URLs from the provider's website, or they may provide that information in a metadata file.

Procedure

Step 1 Create a SAML identity provider in webvpn config mode and enter saml-idp sub-mode under webvpn.

[**no**] **saml idp** *idp-entityID*

idp-entityID— The SAML IdP entityID must contain 4 to 256 characters.

To remove a SAML IdP, use the **no** form of this command.

Step 2 Configure the IdP URLs.

url [**sign-in** | **sign-out**] *value*

value —This is the URL for signing into the IdP or the URL for redirecting to when signing out of the IdP. The **sign-in** URL is required, the **sign-out** URL is optional The url value must contain 4 to 500 characters.

Step 3 (Optional) Configure the Clientless VPN base URL.

base-url *URL*

This URL is provided to third-party IdPs to redirect end users back to the ASA.

When base-url is configured, we use it as the base URL of the AssertionConsumerService and SingleLogoutService attribute in **show saml metadata**.

When base-url is not configured, the URL is determined by the ASA's hostname and domain-name. For example, we use `https://ssl-vpn.cisco.com` when hostname is `ssl-vpn` and domain-name is `cisco.com`.

An error occurs if neither base-url nor the hostname/domain-name are configured when entering **show saml metadata**.

Step 4 Configure trustpoints between the IdP and SP (ASA).

trustpoint idp *trustpoint-name1* [*trustpoint-name2*]

trustpoint sp *trustpoint-name1*

idp—Specifies the trustpoint that contains the IdP certificate for the ASA to verify SAML assertions.

You can now configure two trustpoints for a SAML IdP on a device. This feature allows you to gracefully transition to a new Identity Provider (IdP) certificate without any loss of service. You do not need to open a maintenance window to simultaneously update all ASAs and the IdP with the same certificate. When the new IdP certificate is enabled on the IdP, the device automatically detects the new certificate. You can safely delete the original trustpoint after the transition.

sp —Specifies the trustpoint that contains the ASA (SP)'s certificate for the IdP to verify ASA's signature or encrypted SAML assertion.

trustpoint-name—Must be a previously configured trustpoint.

Step 5 (Optional) Configure the local base URL.

local base-url *URL*

In a DNS load balancing cluster, when SAML authentication is configured on ASAs, you can specify a base URL that uniquely resolves to the device on which the configuration is applied.

Step 6 (Optional) Configure SAML timeout.

timeout assertion *timeout-in-seconds*

If specified, this configuration overrides NotOnOrAfter if the sum of NotBefore and timeout-in-seconds is earlier than NotOnOrAfter.

If not specified, NotBefore and NotOnOrAfter in the assertion is used to determine the validity.

Note For a tunnel group with existing SAML IdP configured, any changes to the saml idp CLI under webvpn are only applied to the tunnel group when SAML is re-enabled for that particular tunnel group. After you configure the timeout, the updated timeout takes effect only after re-issuing the saml identity-provider CLI in the tunnel group webvpn-attributes.

Step 7 (Optional) Enable or disable (default setting) the signature in SAML request.

signature <value>

Note With the upgrade to SSO 2.5.1, the default signing method changes from SHA1 to SHA256, and you can configure which signing method option you prefer by entering the *value* rsa-sha1, rsa-sha256, rsa-sha384, or rsa-sha512.

Step 8 (Optional) To set the flag determining that the IdP is an internal network, use the **internal** command. The ASA will then work in a gateway mode.

Step 9 Use **show webvpn saml idp** to view the configuration.

Step 10 Use **force re-authentication** to cause the identity provider to authenticate directly rather than rely on a previous security context when a SAML authentication request occurs. This setting is the default; therefore, to disable, use **no force re-authentication**.

Example

The following example configures an IdP named salesforce_idp and uses preconfigured trustpoints:

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)#saml idp salesforce_idp

ciscoasa(config-webvpn-saml-idp)#url sign-in
https://asa-dev-ed.my.salesforce.com/idp/endpoint/HttpRedirect
ciscoasa(config-webvpn-saml-idp)#url sign-out
https://asa-dev-ed.my.salesforce.com/idp/endpoint/HttpRedirect

ciscoasa(config-webvpn-saml-idp)#trustpoint idp salesforce_trustpoint salesforce_trustpoint2
ciscoasa(config-webvpn-saml-idp)#trustpoint sp asa_trustpoint

ciscoasa(config)#show webvpn saml idp
saml idp salesforce_idp
url sign-in https://asa-dev-ed.my.salesforce.com/idp/endpoint/HttpRedirect
url sign-out https://asa-dev-ed.my.salesforce.com/idp/endpoint/HttpRedirect
trustpoint idp salesforce_trustpoint salesforce_trustpoint2
trustpoint  sp asa_trustpoint
```

The following web page shows an example of how to get URLs for Onelogin,

https://onelogin.zendesk.com/hc/en-us/articles/202767260-Configuring-SAML-for-Clarizen

The following web page is an example of how to use metadata to find the URLs from OneLogin.

http://onlinehelp.tableau.com/current/online/en-us/saml_config_onelogin.htm

What to do next

Apply SAML authentication to connection profiles, as described in Configure ASA as a SAML 2.0 Service Provider (SP), on page 26.

Configure ASA as a SAML 2.0 Service Provider (SP)

Before you begin

The IdP must have been previously configured. See Configure a SAML 2.0 Identity Provider (IdP), on page 23.

Procedure

Step 1 In tunnel-group webvpn sub-mode, use the saml identity-provider command to assign an IdP.

saml identity-provider *idp-entityID*

idp-entityID—Must be one of the existing IdPs previously configured.

To disable SAML SP, use the **no** form of this command.

Step 2 Enable the SAML authentication method.

authentication saml

Step 3 (Optional) Configure the SAML IdP trustpoint for the tunnel group.

saml idp-trustpoint *trustpoint-name* [*trustpoint-name2*]

If your IdP, for example Azure IdP, has multiple applications but shares the same SAML entity ID, and each application has its own certificate. You can use the above command associated with a given application to override the main webvpn saml configuration.

Example

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)# tunnel-group-list enable
ciscoasa(config)# tunnel-group cloud_idp_onelogin type remote-access
ciscoasa(config)# tunnel-group cloud_idp_onelogin webvpn-attributes
ciscoasa(config-tunnel-webvpn)# authentication saml
ciscoasa(config-tunnel-webvpn)# group-alias cloud_idp enable
ciscoasa(config-tunnel-webvpn)# saml identity-provider
https://app.onelogin.com/saml/metadata/462950
```

Configure an IdP trustpoint for a tunnel group:

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)# tunnel-group-list enable
ciscoasa(config)# tunnel-group partner-saml type remote-access
ciscoasa(config)# tunnel-group partner-saml webvpn-attributes
ciscoasa(config-tunnel-webvpn)# authentication saml
ciscoasa(config-tunnel-webvpn)# saml identity-provider https://sts.windows.net/123-3456-7890/
ciscoasa(config-tunnel-webvpn)# saml idp-trustpoint azure-partner-idp-tp
```

Configure Default OS Browser for SAML Authentication

Specify whether or not AnyConnect should handle the SSO authentication process using the platform's native browser (the operating system's default browser) or using the browser that is embedded in AnyConnect.

You must download the AnyConnect external browser package (Example, *external-sso-4.10.04065-webdeploy-k9.pkg*) and upload it to ASA. You can then choose the SAML login method (AnyConnect's embedded browser or the operating system's default browser) for SAML authentication. This bundle is a script that allows the VPN client to launch the default OS web browser for authentication purposes, and is independent of the operating system, browser, and VPN Client version. As long as the feature is enabled, the VPN client version and the external browser package version file do not need to match.

Choosing the default operating system browser enables single sign-on (SSO) between your VPN authentication and other corporate logins. Choose this option if you want to support web authentication methods, such as biometric authentication, that cannot be performed in the VPN client's embedded browser. Before selecting the operating system's browser, you must upload a package that can be run in the browser to enable web authentication.

Procedure

Step 1 In webvpn sub-mode, use the anyconnect external-browser-pkg command to enable AnyConnect SAML authentication through the operating system's default browser.

anyconnect external-browser-pkg *path*

To disable the operating system's default browser for SAML authentication, use the **no** form of this command.

Step 2 In tunnel-group webvpn sub-mode, use the external-browser command to enable AnyConnect SAML authentication through the operating system's default browser.

external-browser enable *idp-entityID*

To disable the operating system's default browser for SAML authentication, use the **no** form of this command.

Example

This example selects the path for the AnyConnect external browser package and enables an external browser (the operating system's default browser) for SAML authentication.

```
asa(config-webvpn)# anyconnect external-browser-pkg flashshow :
asa(config)# tunnel-group SAML webvpn-attributes
asa(config-tunnel-webvpn)# external-browser enable
asa(config-tunnel-webvpn)#
```

Configure Certificate and SAML Authentication

You can configure certificate and SAML authentication for SAML-based connection profiles to validate customer owned assets without profiling for a particular file/registry key. SAML based authentications can be tied to sanctioned assets and/or users. You can use a single certificate or multiple certificates with SAML for authentication.

When the Secure Client initiates a connection, ASA or FTD will request and authenticate one or more certificates from the endpoint before SAML authentication is performed.

Once SAML authentication is complete, the SAML and certificate username can be

Once SAML authentication is complete, the SAML and certificate username can be compared before proceeding to the authorization phase.

Before you begin

Ensure that you configure required SAML settings before configuration Certificate and SAML authentication:

- Get the Sign-in and Sign-out URLs for your SAML (IdP) provider. You can get the URLs from the provider's website, or they may provide that information in a metadata file.
- Configure SAML identity provider and trustpoint settings. See Configure Certificate and SAML Authentication, on page 27

Procedure

Step 1 To configure certificate and SAML authentication, enter tunnel-group webvpn-attributes mode by entering the following command. The prompt changes to indicate the mode change:

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

Step 2 To specify the authentication method to use, enter the following command:

```
hostname(config-tunnel-webvpn)#authentication authentication_method
```

For example, The following command allows both SAML and certificate authentication:

```
hostname(config-tunnel-webvpn)#authentication saml certificate
```

The following command allows certificate and SAML authentication:

```
hostname(config-tunnel-webvpn)#authentication certificate saml
```

The following command allows both multiple certificate and SAML authentication:

```
hostname(config-tunnel-webvpn)#authentication multiple-certificate saml
```

Step 3 Add or edit a connection profile and then select **Basic** connection profile attribute settings.

Step 4 To specify the authentication method for certificate and SAML authentication, select SAML and certificate or Multiple certificates and SAML from the drop-down.

Example

The following example configures multiple certificates and SAML authentication for the sales_group connection profile:

```
ciscoasa(config)# tunnel-group sales_group webvpn
ciscoasa(config-tunnel-webvpn)#authentication multiple-certificate saml
```

Example SAML 2.0 and Onelogin

Follow this example using your third party SAML 2.0 IdP in place of the Onelogin information and naming.

1. Set time synchronization between the IdP and the ASA(SP).

   ```
   ciscoasa(config)# ntp server 209.244.0.4
   ```

2. Obtain the IdP's SAML metadata from the IdP following procedures provided by your third party IdP.

3. Import the IdP's signing certificate into a trustpoint.

   ```
   ciscoasa(config)# crypto ca trustpoint onelogin
   ciscoasa(config-ca-trustpoint)# enrollment terminal
   ciscoasa(config-ca-trustpoint)# no ca-check
   ciscoasa(config-ca-trustpoint)# crypto ca authenticate onelogin
   Enter the base 64 encoded CA certificate.
   End with the word "quit" on a line by itself
   quit
   INFO: Certificate has the following attributes:
   Fingerprint:     85de3781 07388f5b d92d9d14 1e22a549
   Do you accept this certificate? [yes/no]: yes
   Trustpoint CA certificate accepted.
   % Certificate successfully imported
   ```

4. Import the SP (ASA) signing PKCS12 into a trustpoint

   ```
   ciscoasa(config)# crypto ca import asa_saml_sp pkcs12 password
   Enter the base 64 encoded pkcs12.
   End with the word "quit" on a line by itself:
   quit
   INFO: Import PKCS12 operation completed successfully
   ```

5. Add a SAML IdP:

   ```
   ciscoasa(config-webvpn)# saml idp https://app.onelogin.com/saml/metadata/462950
   ```

6. Configure attributes under saml-idp sub-mode:

   Configure the IdP sign-in URL and sign-ou URL:

   ```
   ciscoasa(config-webvpn-saml-idp)# url sign-in
   https://ross.onelogin.com/trust/saml2/http-post/sso/462950
   ciscoasa(config-webvpn-saml-idp)# url sign-out
   https://ross.onelogin.com/trust/saml2/http-redirect/slo/462950
   ```

   Configure the IdP trustpoint and the SP trustpoint

   ```
   ciscoasa(config-webvpn-saml-idp)# trustpoint idp onelogin
   ciscoasa(config-webvpn-saml-idp)# trustpoint sp asa_saml_sp
   ```

   Configure the Clientless VPN base URL, SAML request signature and SAML assertion timout:

   ```
   ciscoasa(config-webvpn-saml-idp)# base-url https://172.23.34.222
   ciscoasa(config-webvpn-saml-idp)# signature
   ciscoasa(config-webvpn-saml-idp)# timeout assertion 7200
   ```

7. Configure an IdP for a tunnel group and enable SAML authentication.

   ```
   ciscoasa(config)# webvpn
   ciscoasa(config-webvpn)# tunnel-group-list enable
   ```

```
ciscoasa(config)# tunnel-group cloud_idp_onelogin type remote-access
ciscoasa(config)# tunnel-group cloud_idp_onelogin webvpn-attributes
ciscoasa(config-tunnel-webvpn)# authentication saml
ciscoasa(config-tunnel-webvpn)# group-alias cloud_idp enable
ciscoasa(config-tunnel-webvpn)# saml identity-provider
https://app.onelogin.com/saml/metadata/462950
```

8. Show the ASA's SAML SP metadata:

   You can get the ASA's SAML SP metadata from `https://172.23.34.222/saml/sp/metadata/cloud_idp_onelogin`. In the URL, `cloud_idp_onelogin` is the tunnel group name.

9. Configure a SAML SP on your third party IdP following procedures provided by your third party IdP.

Troubleshooting SAML 2.0

Use **debug webvpn saml***value* to debug SAML 2.0 behavior. The following SAML messages will be displayed depending on the *value* :

- 8—errors
- 16—warnings and errors
- 128 or 255—debug, warnings, and errors

Monitor Secure Client Connections

To view information about active sessions, use the **show vpn-sessiondb** command:

Command	Purpose
show vpn-sessiondb	Displays information about active sessions.
vpn-sessiondb logoff	Logs off VPN sessions.
show vpn-sessiondb anyconnect	Enhances the VPN session summary to show OSPFv3 session info
show vpn-sessiondb ratio encryption	Shows the number of tunnels and percentages for the Suite B algorit as AES-GCM-128, AES-GCM-192, AES-GCM-256, AES-GMAC so on).

Note **AnyConnect Parent Tunnel**

AnyConnect parent tunnels do not have assigned IP addresses.

This is the main session that is created during the negotiation in order to set up the session token that is necessary in case a reconnect is needed due to network connectivity issues or hibernation. Based on the connection mechanism, the Cisco Adaptive Security Appliance (ASA) lists the session as Clientless (Weblaunch via the Portal) or Parent (Standalone AnyConnect).

AnyConnect parent represents the session when the client is not actively connected. Effectively, it works similar to a cookie, in that it is a database entry on the ASA that maps to the connection from a particular client. If the client sleeps/hibernates, the tunnels (IPsec/Internet Key Exchange (IKE)/ Transport Layer Security (TLS)/Datagram Transport Layer Security (DTLS) protocols) are torn down, but the Parent remains until the idle timer or maximum connect time takes effect. This allows the user to reconnect without reauthenticating.

Example

The Inactivity field shows the elapsed time since an Secure Client session lost connectivity. If the session is active, 00:00m:00s appears in this field.

```
hostname# show vpn-sessiondb

Session Type: SSL VPN Client

Username     : lee
Index        : 1                        IP Addr      : 209.165.200.232
Protocol     : SSL VPN Client           Encryption   : 3DES
Hashing      : SHA1                     Auth Mode    : userPassword
TCP Dst Port : 443                      TCP Src Port : 54230
Bytes Tx     : 20178                    Bytes Rx     : 8662
Pkts Tx      : 27                       Pkts Rx      : 19
Client Ver   : Cisco STC 1.1.0.117
Client Type  : Internet Explorer
Group        : DfltGrpPolicy
Login Time   : 14:32:03 UTC Wed Mar 20 2007
Duration     : 0h:00m:04s
Inactivity    : 0h:00m:04s
Filter Name  :

hostname# vpn-sessiondb logoff
INFO: Number of sessions of type "" logged off : 1

hostname# vpn-sessiondb logoff name tester
Do you want to logoff the VPN session(s)? [confirm]
INFO: Number of sessions with name "tester" logged off : 1
```

Log Off AnyConnect VPN Sessions

To log off all VPN sessions, use the **vpn-sessiondb logoff** command in global configuration mode:

The following example logs off all VPN sessions:

```
hostname# vpn-sessiondb logoff
INFO: Number of sessions of type "" logged off : 1
```

You can log off individual sessions using either the name argument or the index argument:

vpn-sessiondb logoff name *name*
vpn-sessiondb logoff index *index*

The sessions that have been inactive the longest time are marked as idle (and are automatically logged off) so that license capacity is not reached and new users can log in. If the session resumes at a later time, it is removed from the inactive list.

You can find both the username and the index number (established by the order of the client images) in the output of the **show vpn-sessiondb anyconnect** command. The following examples shows the username *lee* and index number *1*.

```
hostname# show vpn-sessiondb anyconnect

Session Type: AnyConnect

Username     : lee                    Index        : 1
Assigned IP  : 192.168.246.1           Public IP    : 10.139.1.2
Protocol     : AnyConnect-Parent SSL-Tunnel DTLS-Tunnel
License      : AnyConnect Premium
Encryption   : RC4 AES128             Hashing      : SHA1
Bytes Tx     : 11079                  Bytes Rx     : 4942
Group Policy : EngPolicy            Tunnel Group : EngGroup
Login Time   : 15:25:13 EST Fri Jan 28 2011
Duration     : 0h:00m:15s
Inactivity   : 0h:00m:00s
NAC Result   : Unknown
VLAN Mapping : N/A                    VLAN         : none
```

The following example terminates the session using the **name** option of the **vpn-session-db logoff** command:

```
hostname# vpn-sessiondb logoff name lee
Do you want to logoff the VPN session(s)? [confirm]
INFO: Number of sessions with name "lee" logged off : 1

hostname#
```

Feature History for Secure Client Connections

The following table lists the release history for this feature.

Table 2: Feature History for Secure Client Connections

Feature Name	Releases	Feature Information
Secure Client Connections	7.2(1)	The following commands were introduced or modified: authentication e authentication ms-chap-v1, authentication ms-chap-v2, authentication p tunnel hello, vpn-tunnel-protocol l2tp-ipsec.
IPsec IKEv2	8.4(1)	IKEv2 was added to support IPsec IKEv2 connections for Secure Clien LAN-to-LAN.