Advanced Clientless SSL VPN Configuration

Microsoft Kerberos Constrained Delegation Solution

Microsoft's Kerberos Constrained Delegation (KCD) provides access to Kerberos-protected Web applications in the private network.

In order for Kerberos Constrained Delegation to function, the ASA must establish a trust relationship between the source domain (the domain where the ASA resides) and the target or resource domain (the domain where the Web services reside). The ASA crosses the certification path from the source to the destination domain and acquires the necessary tickets on behalf of the remote access user to access the services.

This crossing of the certificate path is called cross-realm authentication. During each phase of cross-realm authentication, the ASA relies on the credentials at a particular domain and the trust relationship with the subsequent domain.

How KCD Works

Kerberos relies on a trusted third party to validate the digital identity of entities in a network. These entities (such as users, host machines, and services running on hosts) are called principals and must be present in the same domain. Instead of secret keys, Kerberos uses tickets to authenticate a client to a server. The ticket is derived from the secret key and consists of the client's identity, an encrypted session key, and flags. Each ticket is issued by the key distribution center and has a set lifetime.

The Kerberos security system is a network authentication protocol used to authenticate entities (users, computers, or applications) and protect network transmissions by scrambling the data so that only the device that the information was intended for can decrypt it. You can configure KCD to provide Clientless SSL VPN users with SSO access to any Web services protected by Kerberos. Examples of such Web services or applications include Outlook Web Access (OWA), Sharepoint, and Internet Information Server (IIS).

Two extensions to the Kerberos protocol were implemented: *protocol transition* and *constrained delegation*. These extensions allow the Clientless SSL VPN remote access users to access Kerberos-authenticated applications in the private network.

P*rotocol transition* provides you with increased flexibility and security by supporting different authentication mechanisms at the user authentication level and by switching to the Kerberos protocol for security features (such as mutual authentication and constrained delegation) in subsequent application layers. *Constrained delegation* provides a way for domain administrators to specify and enforce application trust boundaries by limiting where application services can act on a user's behalf. This flexibility improves application security designs by reducing the chance of compromise by an untrusted service.

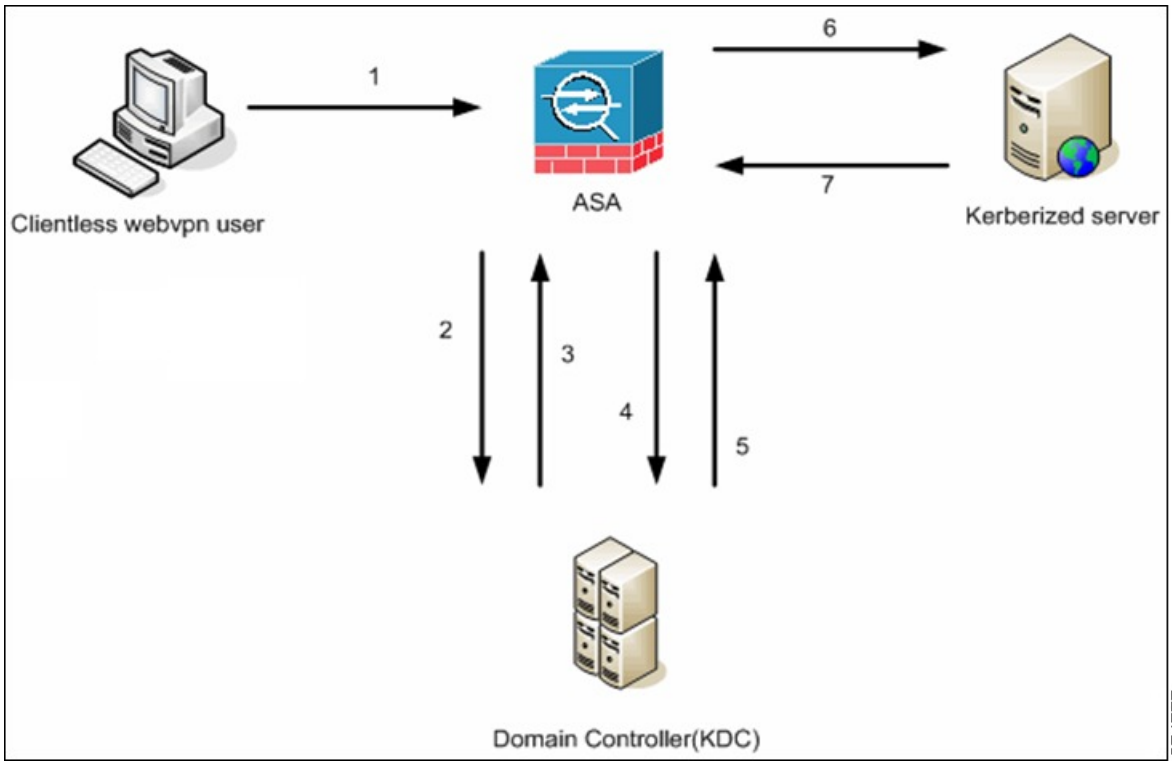
For more information on constrained delegation, see RFC 1510 via the IETF website (http://www.ietf.org).

Authentication Flow with KCD

The following figure depicts the packet and process flow a user experiences directly and indirectly when accessing resources trusted for delegation via the clientless portal. This process assumes that the following tasks have been completed:

- Configured KCD on ASA.
- Joined the Windows Active Directory and ensured services are trusted for delegation.
- Delegated ASA as a member of the Windows Active Directory domain.

Figure 1: KCD Process



Note A clientless user session is authenticated by the ASA using the authentication mechanism configured for the user. (In the case of smartcard credentials, ASA performs LDAP authorization with the userPrincipalName from the digital certificate against the Windows Active Directory).

1. After successful authentication, the user logs in to the ASA clientless portal page. The user accesses a Web service by entering a URL in the portal page or by clicking on the bookmark. If the Web service requires authentication, the server challenges ASA for credentials and sends a list of authentication methods supported by the server.

Note KCD for Clientless SSL VPN is supported for all authentication methods (RADIUS, RSA/SDI, LDAP, digital certificates, and so on). Refer to the AAA Support table at http://www.cisco.com/en/US/docs/security/asa/asa84/configuration/guide/access_aaa.html#wp1069492.

2. Based on the HTTP headers in the challenge, the ASA determines whether the server requires Kerberos authentication. (This is part of the SPNEGO mechanism.) If connecting to a backend server requires Kerberos authentication, the ASA requests a service ticket for itself on behalf of the user from the key distribution center.
3. The key distribution center returns the requested tickets to the ASA. Even though these tickets are passed to the ASA, they contain the user's authorization data. The ASA requests a service ticket from the KCD for the specific service that the user wants to access.

Note Steps 1 to 3 comprise protocol transition. After these steps, any user who authenticates to the ASA using a non-Kerberos authentication protocol is transparently authenticated to the key distribution center using Kerberos.

4. The ASA requests a service ticket from the key distribution center for the specific service that the user wants to access.
5. The key distribution center returns a service ticket for the specific service to the ASA.
6. The ASA uses the service ticket to request access to the Web service.
7. The Web server authenticates the Kerberos service ticket and grants access to the service. The appropriate error message is displayed and requires acknowledgment if there is an authentication failure. If the Kerberos authentication fails, the expected behavior is to fall back to basic authentication.

Create a Kerberos Server Group for Constrained Delegation

To use Kerberos Constrained Delegation, you must first configure a Kerberos AAA server group. The server group must contain the Active Directory (AD) domain controller.

Procedure

Step 1 Create the Kerberos AAA server group and enter aaa-server-group configuration mode.

aaa-server *server_group_name* **protocol kerberos**

Example:

```
ciscoasa(config)# aaa-server MSKCD protocol kerberos
```

Step 2 (Optional.) Specify the maximum number of failed AAA transactions with a AAA server in the group before trying the next server.

max-failed-attempts *number*

Example:

```
ciscoasa(config-aaa-server-group)# max-failed-attempts 2
```

The *number* argument can range from 1 and 5. The default is 3.

Step 3 (Optional.) Specify the method (reactivation policy) by which failed servers in a group are reactivated.

reactivation-mode {**depletion** [**deadtime** *minutes*] | **timed**}

Example:

```
ciscoasa(config-aaa-server-group)# reactivation-mode depletion deadtime 20
```

The **depletion** keyword reactivates failed servers only after all of the servers in the group are inactive. This is the default mode.

The **deadtime** *minutes* keyword pair specifies the amount of time in minutes, between 0 and 1440, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes.

The **timed** keyword reactivates failed servers after 30 seconds of down time.

Step 4 Add the Kerberos server to the Kerberos server group.

aaa-server *server_group* [**(***interface_name***)**] **host** *server_ip*

Example:

```
ciscoasa(config-aaa-server-group)# aaa-server MSKCD (inside) host 10.1.1.10
```

If you do not specify an interface, then the ASA uses the **inside** interface by default.

You can use an IPv4 or IPv6 address.

Step 5 Specify the timeout value for connection attempts to the server.

timeout *seconds*

Specify the timeout interval (1-300 seconds) for the server; the default is 10 seconds. For each AAA transaction the ASA retries connection attempts (based on the interval defined on the **retry-interval** command) until the timeout is reached. If the number of consecutive failed transactions reaches the limit specified on the max-failed-attempts command in the AAA server group, the AAA server is deactivated and the ASA starts sending requests to another AAA server if it is configured.

Example:

```
ciscoasa(config-aaa-server-host)# timeout 15
```

Step 6 Specify the retry interval, which is the time the system waits before retrying a connection request.

retry-interval *seconds*

You can specify 1-10 seconds. The default is 10.

Example:

```
ciscoasa(config-aaa-server-host)# retry-interval 6
```

Step 7 Specify the server port if it is different from the default Kerberos port, which is TCP/88. The ASA contacts the Kerberos server on this port.

server-port *port_number*

Example:

```
ciscoasa(config-aaa-server-host)# server-port 8888
```

Step 8 Configure the Kerberos realm.

kerberos-realm *name*

Kerberos realm names use numbers and upper case letters only, and can be up to 64 characters. The name should match the output of the Microsoft Windows **set USERDNSDOMAIN** command when it is run on the Active Directory server for the Kerberos realm. In the following example, EXAMPLE.COM is the Kerberos realm name:

```
C:\>set USERDNSDOMAIN
USERDNSDOMAIN=EXAMPLE.COM
```

Although the ASA accepts lower case letters in the name, it does not translate lower case letters to upper case letters. Be sure to use upper case letters only.

Example:

```
ciscoasa(config-asa-server-group)# kerberos-realm EXAMPLE.COM
```

Example

The following example creates a Kerberos server group named MSKCD, adds a server, and sets the realm to EXAMPLE.COM.

```
hostname(config)# aaa-server MSKCD protocol kerberos
hostname(config-aaa-server-group)# aaa-server MSKCD (inside) host 10.1.1.10
hostname(config-aaa-server-host)# kerberos-realm EXAMPLE.COM
```

Configure Kerberos Constrained Delegation (KCD)

The following procedure explains how to implement Kerberos Constrained Delegation (KCD).

Before you begin

- Enable DNS lookup on the interface through which the domain controller is reached. When using KCD as the authentication delegation method, DNS is required to enable hostname resolution and communication

between the ASA, Domain Controller (DC), and the services trusted for delegation. Clientless VPN deployments require DNS Lookups through the internal corporate network, typically the inside interface.

For example, to enable DNS lookup on the inside interface:

```
hostname(config)# dns domain-lookup inside
```

- Configure DNS to use the Active Directory (AD) domain controller as the DNS server, with the domain realm as the DNS domain.

  For example, to configure the DefaultDNS group to use domain controller at 10.1.1.10 off the inside interface with the realm EXAMPLE.COM:

```
hostname(config)# dns server-group DefaultDNS
hostname(config-dns-server-group)# name-server 10.1.1.10 inside
hostname(config-dns-server-group)# domain-name EXAMPLE.COM
```

Procedure

Step 1 Switch to Clientless SSL VPN configuration mode.

webvpn

Step 2 Enable KCD.

kcd-server *kerberos_server_group* **username** *user_id* **password** *password* [**validate-server-certificate**]

Where:

- *kerberos_server_group* is the name of the Kerberos AAA server group you created for KCD. See Create a Kerberos Server Group for Constrained Delegation, on page 3.
- **username** *user_id* specifies a username defined on the domain controller that the system can use to join the domain. The user account must have administrative privileges or service level privileges for adding devices to the domain.
- **password** *password* specifies the password for the user account.
- **validate-server-certificate** instructs the ASA to validate the server certificate and thus the identity of the server when joining the domain. This parameter is optional.

Example:

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)# kcd-server MSKCD username administrator
password !ou8one2 validate-server-certificate
```

Monitoring Kerberos Constrained Delegation

You can use the following commands to monitor KCD.

- **show webvpn kcd**

Shows the KCD configuration and join status.

```
ciscoasa# show webvpn kcd

KCD state:      Domain Join Complete
Kerberos Realm: EXAMPLE.COM
ADI version:    6.8.0_1252
Machine name:   ciscoasa
ADI instance:   root      1181  1178  0 15:35 ?        00:00:01 /asa/bin/start-adi
Keytab file:    -rw------- 1 root root 79 Jun 16 16:06 /etc/krb5.keytab
```

- **show aaa kerberos** [**username** *user_id*]

  Shows the Kerberos tickets cached on the system. You can view all tickets, or just those tickets for a given user.

```
ASA# show aaa kerberos

Default Principal      Valid Starting        Expires             Service Principal
asa@example.COM        06/29/10 18:33:00     06/30/10 18:33:00
krbtgt/example.COM@example.COM
kcduser@example.COM    06/29/10 17:33:00     06/30/10 17:33:00
asa$/example.COM@example.COM
kcduser@example.COM    06/29/10 17:33:00     06/30/10 17:33:00
http/owa.example.com@example.COM
```

- **clear aaa kerberos tickets** [**username** *user_id*]

  Clears the Kerberos tickets cached on the system. You can clear all tickets, or just those tickets for a given user.

Configure Application Profile Customization Framework

Clientless SSL VPN includes an Application Profile Customization Framework (APCF) option that lets the ASA handle non-standard applications and Web resources so they display correctly over a Clientless SSL VPN connection. An APCF profile contains a script that specifies when (pre, post), where (header, body, request, response), and what (data) to transform for a particular application. The script is in XML and uses sed (stream editor) syntax to transform strings/text.

You can configure and run multiple APCF profiles in parallel on an ASA. Within an APCF profile script, multiple APCF rules can apply. The ASA processes the oldest rule first, based on configuration history, the next oldest rule next.

You can store APCF profiles on the ASA flash memory, or on an HTTP, HTTPS, or TFTP server.

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

Manage APCF Packets

Procedure

Step 1 Switch to Clientless SSL VPN configuration mode.

webvpn

Step 2 Identify and locate an APCF profile to load on the ASA.

apcf

Example:

This example shows how to enable an APCF profile named apcf1.xml, located in flash memory and how to enable an APCF profile named apcf2.xml, located on an HTTPS server called myserver, port 1440, with the path being /apcf.

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

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

APCF Syntax

APCF profiles use XML format, and sed script syntax, with the XML tags in the following table.

Guidelines for APCF

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

Table 1: APCF XML Tags

Tag	Use
<APCF>...</APCF>	The mandatory root element that opens any APCF XML file.
<version>1.0</version>	The mandatory tag that specifies the APCF implementation version. Currently the only version is 1.0.
<application>...</application>	The mandatory tag that wraps the body of the XML description.
<id> text </id>	The mandatory tag that describes this particular APCF functionality.
<apcf-entities>...</apcf-entities>	The mandatory tag that wraps a single or multiple APCF entities.

Tag	Use
<js-object>…</js-object> <html-object>…</html-object> <process-request-header>...</process-request-header> <process-response-header>...</process-response-header> <preprocess-response-body>...</preprocess-response-body> <postprocess-response-body>...</postprocess-response-body>	One of these tags specifies type of content or the stage at which the APCF processing should take place.
<conditions>… </conditions>	A child element of the pre/post-process tags that specifies criteria for processing such as: • http-version (such as 1.1, 1.0, 0.9) • http-method (get, put, post, webdav) • http-scheme (“http/”, “https/”, other) • server-regexp regular expression containing ("a".."z" | "A".."Z" | "0".."9" | ".-_*[]?") • server-fnmatch (regular expression containing ("a".."z" | "A".."Z" | "0".."9" | ".-_*[]?+()\{},"), • user-agent-regexp • user-agent-fnmatch • request-uri-regexp • request-uri-fnmatch • If more than one of condition tags is present, the ASA performs a logical AND for all tags.
<action> … </action>	Wraps one or more actions to perform on the content under specified conditions; you can use the following tags to define these actions (shown below): • <do> • <sed-script> • <rewrite-header> • <add-header> • <delete-header>

Tag	Use
<do>…</do>	Child element of the action tag used to define one of the following actions: • <no-rewrite/>—Do not mangle the content received from the remote server. • <no-toolbar/>—Do not insert the toolbar. • <no-gzip/>—Do not compress the content. • <force-cache/>—Preserve the original caching instructions. • <force-no-cache/>—Make object non-cacheable. • < downgrade-http-version-on-backend>—Use HTTP/1.0 when sending the request to remote server.
<sed-script> TEXT </sed-script>	Child element of the action tag used to change the content of text-based objects. The Text must be a valid Sed script. The <sed-script> applies to the <conditions> tag defined before it.
<rewrite-header></rewrite-header>	Child element of the action tag. Changes the value of the HTTP header specified in the child element <header> tag shown below.
<add-header></add-header>	Child element of the action tag used to add a new HTTP header specified in the child element <header> tag shown below.
<delete-header></delete-header>	Child element of the action tag used to delete the specified HTTP header specified by the child element <header> tag shown below.

<header></header>	Specifies the name HTTP header to be rewritten, added, or deleted. For example, the following tag changes the value of the HTTP header named Connection:
	
	`<rewrite-header>`
	`<header>Connection</header>`
	`<value>close</value>`
	`</rewrite-header>`

Configuration Examples for APCF

```
<APCF>
<version>1.0</version>
<application>
  <id>Do not compress content from example.com</id>
  <apcf-entities>
```

```
      <process-request-header>
         <conditions>
           <server-fnmatch>*.example.com</server-fnmatch>
         </conditions>
           <action>
             <do><no-gzip/></do>
           </action>
      </process-request-header>
  </apcf-entities>
</application>
</APCF>

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

Encoding

Character encoding, also called “character coding” and “a character set,” is the pairing of raw data (such as 0s and 1s) with characters to represent the data. The language determines the character encoding method to use. Some languages use a single method, while others do not. Usually, the geographic region determines the default encoding method used by the browser, but the remote user can change it. The browser can also detect the encoding specified on the page, and render the document accordingly.

The encoding attribute lets you specify the value of the character-encoding method used on the portal page to ensure that the browser renders it properly, regardless of the region in which the user is using the browser, and regardless of any changes made to the browser.

By default, the ASA applies the “Global Encoding Type” to pages from Common Internet File System servers. The mapping of CIFS servers to their appropriate character encoding, globally with the “Global Encoding Type” attribute, and individually with the file-encoding exceptions displayed in the table, provides for the accurate handling and display of CIFS pages when the proper rendering of filenames or directory paths, as well as pages, is an issue.

View or Specify Character Encoding

With encoding, you can view or specify the character encoding for Clientless SSL VPN portal pages.

Procedure

Step 1 Global Encoding Type determines the character encoding that all Clientless SSL VPN portal pages inherit except for those from the CIFS servers listed in the table. You can type the string or choose one of the options from the drop-down list, which contains the most common values, as follows:

- big5
- gb2312
- ibm-850
- iso-8859-1
- shift_jis

  Note If you are using Japanese Shift_jis Character encoding, click **Do Not Specify** in the Font Family area of the associated Select Page Font pane to remove the font family.

- unicode
- windows-1252
- none

  Note If you click **none** or specify a value that the browser on the Clientless SSL VPN session does not support, it uses its own default encoding.

You can type a string consisting of up to 40 characters, and equal to one of the valid character sets identified in http://www.iana.org/assignments/character-sets. You can use either the name or the alias of a character set listed on that page. The string is case-insensitive. The command interpreter converts upper-case to lower-case when you save the ASA configuration.

Step 2 Enter the name or IP address of a CIFS server for which the encoding requirement differs from the "Global Encoding Type" attribute setting. The ASA retains the case you specify, although it ignores the case when matching the name to a server.

Step 3 Choose the character encoding that the CIFS server should provide for Clientless SSL VPN portal pages. You can type the string, or choose one from the drop-down list, which contains only the most common values, as follows:

- big5
- gb2312
- ibm-850
- iso-8859-1
- shift_jis

  Note If you are using Japanese Shift_jis Character encoding, click **Do Not Specify** in the Font Family area of the associated Select Page Font pane to remove the font family.

- unicode
- windows-1252

- none

If you click **none** or specify a value that the browser on the Clientless SSL VPN session does not support, it uses its own default encoding.

You can type a string consisting of up to 40 characters, and equal to one of the valid character sets identified in http://www.iana.org/assignments/character-sets. You can use either the name or the alias of a character set listed on that page. The string is case-insensitive. The command interpreter converts upper-case to lower-case when you save the ASA configuration.

Use Email over Clientless SSL VPN

Configure Web email: MS Outlook Web App

The ASA supports Microsoft Outlook Web App to Exchange Server 2010 and Microsoft Outlook Web Access to Exchange Server 2013.

Procedure

Step 1 Enter the URL of the email service into the address field or click an associated bookmark in the Clientless SSL VPN session.

Step 2 When prompted, enter the email server username in the format domain\username.

Step 3 Enter the email password.