CHAPTER 16

Advanced Clientless SSL VPN Configuration

June 25, 2014

Microsoft Kerberos Constrained Delegation Solution

Many organizations want to authenticate their Clientless VPN users and extend their authentication credentials seamlessly to web-based resources using authentication methods beyond what the ASA SSO feature can offer today. With the growing demand to authenticate remote access users with smart cards and One-time Passwords (OTPs), the SSO feature falls short in meeting that demand, because it forwards only conventional user credentials, such as static username and password, to clientless web-based resources when authentication is required.

For example, neither certificate- nor OTP-based authentication methods encompass a conventional username and password necessary for the ASA to seamlessly perform SSO access to web-based resources. When authenticating with a certificate, a username and password are not required for the ASA to extend to web-based resources, making it an unsupported authentication method for SSO. On the other hand, OTP does include a static username; however, the password is dynamic and will subsequently change throughout the VPN session. In general, Web-based resources are configured to accept static usernames and passwords, thus also making OTP an unsupported authentication method for SSO.

Microsoft's Kerberos Constrained Delegation (KCD), a new feature introduced in software release 8.4 of the ASA, provides access to Kerberos-protected Web applications in the private network. With this benefit, you can seamlessly extend certificate- and OTP-based authentication methods to Web applications. Thus, with SSO and KCD working together although independently, many organizations can now authenticate their clientless VPN users and extend their authentication credentials seamlessly to Web applications using all authentication methods supported by the ASA.

Requirements

In order for the **kcd-server** command 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, using its unique format, 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.

Understanding 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 Microsoft Web services protected by Kerberos. Supported Web services or applications include Outlook Web Access (OWA), Sharepoint, and Internet Information Server (IIS).

Note Web services from providers other than Microsoft are not currently supported.

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.

Protocol 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.

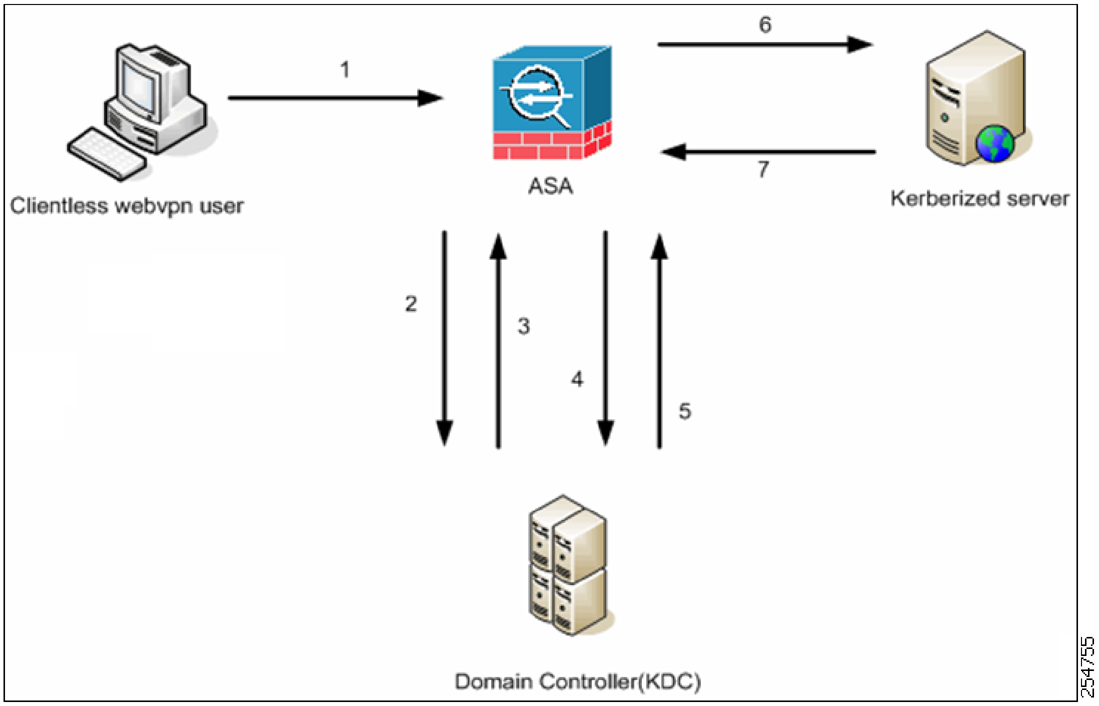
For more information on constrained delegation, see RFC 1510 via the IETF website (http://www.ietf.org).

Authentication Flow with KCD

Figure 16-1 depicts the packet and process flow a user will experience 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 16-1 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, 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.ASA requests a service ticket from the KDC 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 ASA using a non-Kerberos authentication protocol is transparently authenticated to the key distribution center using Kerberos.

4. 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. 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 acknowledgement if there is an authentication failure. If the Kerberos authentication fails, the expected behavior is to fall back to basic authentication.

Before Configuring KCD

To configure the ASA for cross-realm authentication, you must use the following commands.

	Command	Purpose
Step 1	**ntp** hostname **Example:** ciscoasa(config)# **configure terminal** #Create an alias for the Domain Controller ciscoasa(config)# **name 10.1.1.10 DC** #Configure the Name server	Joins the Active Directory domain. A 10.1.1.10 domain controller (which is reachable inside the interface).
Step 2	**dns domain-lookup** **dns server-group** **Example:** ciscoasa(config)# **ntp server DC** #Enable a DNS lookup by configuring the DNS server and Domain name ciscoasa(config)# **dns domain-lookup inside** ciscoasa(config)# **dns server-group DefaultDNS** ciscoasa(config-dns-server-group)# **name-server DC** ciscoasa(config-dns-server-group)# **domain-name private.net** #Configure the AAA server group with Server and Realm ciscoasa(config)# **aaa-server KerberosGroup protocol Kerberos** ciscoasa(config-asa-server-group)# **aaa-server KerberosGroup (inside) host DC** ciscoasa(config-asa-server-group)# **Kerberos-realm PRIVATE.NET** #Configure the Domain Join ciscoasa(config)# **webvpn** ciscoasa(config-webvpn)# **kcd-server KerberosGroup username dcuser password dcuser123**! ciscoasa(config)#	Performs a lookup. A domain name of private.net and a service account on the domain controller using dcuser as the username and dcuser123! as the password.

Configuring KCD

To have the ASA join a Windows Active Directory domain and return a success or failure status, perform these steps.

DETAILED STEPS

	Command	Purpose
Step 1	**webvpn**	Switches to Clientless SSL VPN configuration mode.
Step 2	**kcd-server**	Configure the KCD.
Step 3	**kcd-server** ***aaa-server-group*** **Example:** ASA(config)# **aaa-server KG protocol kerberos** ASA(config)# **aaa-server KG (inside) host DC** ASA(config-aaa-server-host)# k**erberos-realm test.edu** ASA(webvpn-config)# k**cd-server KG username user1 password abc123** ASA(webvpn-config)# **no kcd-server**	Specifies the domain controller name and realm. The AAA server group must be a Kerberos type.
Step 4	(Optional) **no kcd-server**	Removes the specified behavior for the ASA.
Step 5	(Optional) **kcd-server reset**	Resets to the internal state.
Step 6	**kcd domain-join username** *<user>* **password** *<pass>* user—Does not correspond to a specific administrative user but simply a user with service-level privileges to add a device on the Windows domain controller. pass—The password does not correspond to a specific password but simply a user with service-level password privileges to add a device on the Windows domain controller.	Checks for the presence of a KCD server and starts the domain join process. The Active Directory username and password are used only in EXEC mode and are not saved in the configuration. **Note** Administrative privileges are required for initial join. A user with service-level privileges on the domain controller will not get access.
Step 7	kcd domain-leave	Verifies whether the KCD server command has a valid domain join status and then initiates a domain leave.

Showing KCD Status Information

To display the domain controller information and the domain join status, perform this step.

	Command	Purpose
Step 8	**show webvpn kcd** **Example:** ASA# show webvpn kcd KCD-Server Name: DC User : user1 Password : **** KCD State : Joined	Displays the domain controller information and the domain join status.

Showing Cached Kerberos Tickets

To display all Kerberos tickets cached on the ASA, enter the following commands:

<table>
<tr><th></th><th>Command</th><th>Purpose</th></tr>
<tr><td><b>Step 9</b></td><td><b>show aaa kerberos</b></td><td>Displays all Kerberos tickets cached on the ASA.</td></tr>
<tr><td><b>Step 10</b></td><td><b>show aaa kerberos [username <i>user</i> | host <i>ip</i> | <i>hostname</i>]</b>

<b>Example:</b>
ASA# <b>show aaa kerberos</b>

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.COM06/29/10 17:33:00   06/30/10
17:33:00   asa$/example.COM@example.COM
kcduser@example.COM06/29/10 17:33:00   06/30/10
17:33:00   http/owa.example.com@example.COM

ASA# show aaa kerberos username kcduser

Default Principal Valid Starting   Expires
Service Principal
kcduser@example.COM06/29/10 17:33:00   06/30/10
17:33:00   asa$/example.COM@example.COM
kcduser@example.COM06/29/10 17:33:00   06/30/10
17:33:00   http/owa.example.com@example.COM

ASA# show aaa kerberos host owa.example.com

Default Principal Valid Starting   Expires
Service Principal
kcduser@example.COM06/29/1006/30/10 17:33:00
http/owa.example.com@example.COM
ASA# show aaa kerberos username kcduser

Default Principal Valid Starting   Expires
Service Principal
kcduser@example.COM06/29/10 17:33:00   06/30/10
17:33:00   asa$/example.COM@example.COM
kcduser@example.COM06/29/10 17:33:00   06/30/10
17:33:00   http/owa.example.com@example.COM

ASA# show aaa kerberos host owa.example.com

Default Principal Valid Starting   Expires
Service Principal
kcduser@example.COM06/29/10   06/30/10
17:33:00   http/owa.example.com@example.COM</td><td>• user—Used to view the Kerberos tickets of a specific user
• hostname—Used to view the Kerberos tickets issued for a specific host</td></tr>
</table>

Clearing Cached Kerberos Tickets

To clear all Kerberos ticket information on the ASA, perform these steps.

	Command	Purpose
Step 11	`clear aaa kerberos`	Clears all Kerberos ticket information on the ASA.
Step 12	`clear aaa kerberos [username` *`user`* `| host` *`ip`* `|` *`hostname`*`]`	• *user*—Used to clear the Kerberos tickets of a specific user • *hostname*—Used to clear the Kerberos tickets of a specific host

Note

Restrictions

When creating a bookmark to an application that uses Kerberos constrained delegation (KCD), do not check Enable Smart Tunnel.

DETAILED STEPS

Configuring 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.

Restrictions

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

Managing APCF Packets

DETAILED STEPS

	Command	Purpose
Step 1	**webvpn**	Switches to Clientless SSL VPN configuration mode.

Step 2	**apcf**
	Example:
	`ciscoasa(config)# webvpn`
	`ciscoasa(config-webvpn)# apcf flash:/apcf/apcf1.xml`
	`ciscoasa(config)# webvpn`
	`ciscoasa(config-webvpn)# apcf https://myserver:1440/apcf/apcf2.xml`
		Identifies and locates an APCF profile to load on the ASA.
		Shows how to enable an APCF profile named apcf1.xml, located in flash memory.
		Shows how to enable an APCF profile named apcf2.xml, located on an HTTPS server called myserver, port 1440, with the path being /apcf.

APCF Syntax

APCF profiles use XML format, and sed script syntax, with the XML tags in Table 16-1.

Guidelines

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 16-1 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.

Table 16-1 APCF XML Tags (continued)

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>

Table 16-1 APCF XML Tags (continued)

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

Example:

```
<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>
```

Example:

```
<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

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

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.

DETAILED STEPS

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.

Using Email over Clientless SSL VPN

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

- Configuring Email Proxies
- Configuring Web email: MS Outlook Web App

Configuring Email Proxies

Clientless SSL VPN supports IMAP, POP3, and SMTP email proxies. The following attributes apply globally to email proxy users.

Restrictions

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

DETAILED STEPS

	Command	Purpose
Step 1	**accounting-server-group**	Specifies the previously configured accounting servers to use with email proxy.
Step 2	**authentication**	Specifies the authentication method(s) for email proxy users. The default values are as follows: • IMAP: Mailhost (required) • POP3 Mailhost (required) • SMTP: AAA
Step 3	**authentication-server-group**	Specifies the previously configured authentication servers to use with email proxy. The default is LOCAL.
Step 4	**authorization-server-group**	Specifies the previously configured authorization servers to use with Clientless SSL VPN.
Step 5	**authorization-required**	Requires users to authorize successfully to connect. The default is switched off.
Step 6	**authorization-dn-attributes**	Identifies the DN of the peer certificate to use as a username for authorization. The defaults are as follows: • Primary attribute: CN • Secondary attribute: OU
Step 7	**default-group-policy**	Specifies the name of the group policy to use. The default is DfltGrpPolicy.
Step 8	**enable**	Enables email proxy on the specified interface. The default is switched off.

	Command	Purpose
Step 9	**name-separator**	Defines the separator between the email and VPN usernames and passwords. The default is colon (:).
Step 10	**outstanding**	Configures the maximum number of outstanding non-authenticated sessions. The default is 20.
Step 11	**port**	Sets the port the email proxy listens to. The default is as follows: • IMAP:143 • POP3: 110 • SMTP: 25
Step 12	**server**	Specifies the default email server.
Step 13	**server-separator**	Defines the separator between the email and server names. The default is @.

Configuring 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 2007, 2003, and 2000.

DETAILED STEPS

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.