Advanced Clientless SSL VPN Configuration

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.
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.

*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 12-1* 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
Chapter 12      Advanced Clientless SSL VPN Configuration

How KCD Works

Figure 12-1   KCD Process

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.

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 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).

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.
How KCD Works

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

Configure the ASA for Cross-Realm Authentication

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

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Join the Active Directory domain. A 10.1.1.10 domain controller (which is reachable inside the interface).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ntp hostname</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>hostname(config)# configure terminal</td>
</tr>
<tr>
<td></td>
<td>#Create an alias for the Domain Controller</td>
</tr>
<tr>
<td></td>
<td>hostname(config)# name 10.1.1.10 DC</td>
</tr>
<tr>
<td></td>
<td>#Configure the Name server</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Perform a lookup.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This example shows a domain name of private.net and a service account on the domain controller using dcuser as the username and dcuser123! as the password.</td>
</tr>
<tr>
<td></td>
<td>dns domain-lookup</td>
</tr>
<tr>
<td></td>
<td>dns server-group</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>hostname(config)# ntp server DC</td>
</tr>
<tr>
<td></td>
<td>#Enable a DNS lookup by configuring the DNS server and Domain name</td>
</tr>
<tr>
<td></td>
<td>hostname(config)# dns domain-lookup inside</td>
</tr>
<tr>
<td></td>
<td>hostname(config)# dns server-group DefaultDNS</td>
</tr>
<tr>
<td></td>
<td>hostname(config-dns-server-group)# name-server DC</td>
</tr>
<tr>
<td></td>
<td>hostname(config-dns-server-group)# domain-name private.net</td>
</tr>
<tr>
<td></td>
<td>#Configure the AAA server group with Server and Realm</td>
</tr>
<tr>
<td></td>
<td>hostname(config)# aaa-server KerberosGroup protocol Kerberos</td>
</tr>
<tr>
<td></td>
<td>hostname(config-asa-server-group)# aaa-server KerberosGroup (inside) host DC</td>
</tr>
<tr>
<td></td>
<td>hostname(config-asa-server-group)# Kerberos-realm PRIVATE.NET</td>
</tr>
<tr>
<td></td>
<td>#Configure the Domain Join</td>
</tr>
</tbody>
</table>
Configure KCD

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

**Step 1**
Switch to Clientless SSL VPN configuration mode.
```
webvpn
```

**Step 2**
Configure the KCD.
```
kcd-server
```

**Step 3**
Specify the domain controller name and realm. The AAA server group must be a Kerberos type.
```
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)# kerberos-realm test.edu
ASA(webvpn-config)# kcd-server KG username user1 password abc123
ASA(webvpn-config)# no kcd-server
```

**Step 4**
(Optional) Remove the specified behavior for the ASA.
```
no kcd-server
```

**Step 5**
(Optional) Reset to the internal state.
```
kcd-server reset
```

**Step 6**
Check for the presence of a KCD server and start 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.
```
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

**Step 7**
Verify whether the KCD server command has a valid domain join status and then initiate a domain leave.
```
kcd domain-leave
```
How KCD Works

Show KCD Status Information

To display the domain controller information and the domain join status, enter the following command:

```
show webvpn kcd
```

**Example:**

ASA# show webvpn kcd
KCD-Server Name: DC
User : user1
Password : ****
KCD State : Joined

Show Cached Kerberos Tickets

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

```
show aaa kerberos [username user | host ip | hostname]
```

**Example:**

ASA# show aaa kerberos

<table>
<thead>
<tr>
<th>Default Principal</th>
<th>Valid Starting</th>
<th>Expires</th>
<th>Service Principal</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:asa@example.COM">asa@example.COM</a></td>
<td>06/29/10 18:33:00</td>
<td>06/30/10 18:33:00</td>
<td>krbtgt/example.COM@example.COM</td>
</tr>
<tr>
<td><a href="mailto:kcduser@example.COM">kcduser@example.COM</a></td>
<td>06/29/10 17:33:00</td>
<td>06/30/10 17:33:00</td>
<td>asa$/example.COM@example.COM</td>
</tr>
<tr>
<td><a href="mailto:kcduser@example.COM">kcduser@example.COM</a></td>
<td>06/29/10 17:33:00</td>
<td>06/30/10 17:33:00</td>
<td>http/owa.example.com@example.COM</td>
</tr>
<tr>
<td>http/owa.example.com</td>
<td>06/29/10 17:33:00</td>
<td>06/30/10 17:33:00</td>
<td></td>
</tr>
</tbody>
</table>

ASA# show aaa kerberos username kcduser

<table>
<thead>
<tr>
<th>Default Principal</th>
<th>Valid Starting</th>
<th>Expires</th>
<th>Service Principal</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:kcduser@example.COM">kcduser@example.COM</a></td>
<td>06/29/10 17:33:00</td>
<td>06/30/10 17:33:00</td>
<td>asa$/example.COM@example.COM</td>
</tr>
<tr>
<td><a href="mailto:kcduser@example.COM">kcduser@example.COM</a></td>
<td>06/29/10 17:33:00</td>
<td>06/30/10 17:33:00</td>
<td>http/owa.example.com@example.COM</td>
</tr>
<tr>
<td>http/owa.example.com</td>
<td>06/29/10 17:33:00</td>
<td>06/30/10 17:33:00</td>
<td></td>
</tr>
</tbody>
</table>

ASA# show aaa kerberos host owa.example.com

<table>
<thead>
<tr>
<th>Default Principal</th>
<th>Valid Starting</th>
<th>Expires</th>
<th>Service Principal</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:kcduser@example.COM">kcduser@example.COM</a></td>
<td>06/30/10 17:33:00</td>
<td>06/30/10 17:33:00</td>
<td></td>
</tr>
</tbody>
</table>

Clear Cached Kerberos Tickets

To clear all Kerberos ticket information on the ASA, enter the following command:

```
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

Step 8

What To Do Next
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.

Restrictions

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

Manage APCF Packets

Step 1  Switch to Clientless SSL VPN configuration mode.

   webvpn

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

   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.

   apcf

   Example:
   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 Table 12-1.

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 12-1   APCF XML Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;APCF&gt;…&lt;/APCF&gt;</td>
<td>The mandatory root element that opens any APCF XML file.</td>
</tr>
<tr>
<td>&lt;version&gt;1.0&lt;/version&gt;</td>
<td>The mandatory tag that specifies the APCF implementation version. Currently the only version is 1.0.</td>
</tr>
<tr>
<td>&lt;application&gt;…&lt;/application&gt;</td>
<td>The mandatory tag that wraps the body of the XML description.</td>
</tr>
<tr>
<td>&lt;id&gt; text &lt;/id&gt;</td>
<td>The mandatory tag that describes this particular APCF functionality.</td>
</tr>
<tr>
<td>&lt;apcf-entities&gt;…&lt;/apcf-entities&gt;</td>
<td>The mandatory tag that wraps a single or multiple APCF entities.</td>
</tr>
<tr>
<td>&lt;js-object&gt;…&lt;/js-object&gt;</td>
<td>One of these tags specifies type of content or the stage at which the APCF processing should take place.</td>
</tr>
<tr>
<td>&lt;html-object&gt;…&lt;/html-object&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;process-request-header&gt;…&lt;/process-request-header&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;process-response-header&gt;…&lt;/process-response-header&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;preprocess-response-body&gt;…&lt;/preprocess-response-body&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;postprocess-response-body&gt;…&lt;/postprocess-response-body&gt;</td>
<td></td>
</tr>
</tbody>
</table>
## Table 12-1   APCF XML Tags (continued)

<table>
<thead>
<tr>
<th>Tag</th>
<th>Use</th>
</tr>
</thead>
</table>
| <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> |
| <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. |
**Table 12-1  APCF XML Tags (continued)**

<table>
<thead>
<tr>
<th>Tag</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;rewrite-header&gt;</code></td>
<td>Child element of the action tag. Changes the value of the HTTP header specified in the child element <code>&lt;header&gt;</code> tag shown below.</td>
</tr>
<tr>
<td><code>&lt;add-header&gt;</code></td>
<td>Child element of the action tag used to add a new HTTP header specified in the child element <code>&lt;header&gt;</code> tag shown below.</td>
</tr>
<tr>
<td><code>&lt;delete-header&gt;</code></td>
<td>Child element of the action tag used to delete the specified HTTP header specified by the child element <code>&lt;header&gt;</code> tag shown below.</td>
</tr>
<tr>
<td><code>&lt;header&gt;</code></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;rewrite-header&gt;</code>&lt;header&gt;Connection&lt;/header&gt;&lt;value&gt;close&lt;/value&gt; &lt;/rewrite-header&gt;</td>
</tr>
</tbody>
</table>

**Configuration Examples for APCF**

**Example:**

```xml
<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:**

```xml
<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.

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
- unicode

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.
Encoding

- 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](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](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

Configure Email Proxies

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

Restrictions

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

Step 1 Specify the previously configured accounting servers to use with email proxy:
accounting-server-group

Step 2 Specify the authentication method(s) for email proxy users. The default values are as follows:
  • IMAP: Mailhost (required)
  • POP3 Mailhost (required)
  • SMTP: AAA
authentication

Step 3 Specify the previously configured authentication servers to use with email proxy. The default is LOCAL.
authentication-server-group

Step 4 Specify the previously configured authorization servers to use with Clientless SSL VPN:
authorization-server-group

Step 5 Determine if you want to require users to have successful authorization to connect. The default is switched off.
authorization-required

Step 6 Identify the DN of the peer certificate to use as a username for authorization. The defaults are as follows:
  • Primary attribute: CN
  • Secondary attribute: OU
authorization-dn-attributes

Step 7 Specify the name of the group policy to use. The default is DfltGrpPolicy.
default-group-policy

Step 8 Enables email proxy on the specified interface. The default is switched off.
enable

Step 9 Defines the separator between the email and VPN usernames and passwords. The default is colon (:).
name-separator

Step 10 Configure the maximum number of outstanding non-authenticated sessions. The default is 20.
outstanding
Step 11  Set the port the email proxy listens to. The default is as follows:

- IMAP: 143
- POP3: 110
- SMTP: 25

Step 12  Specify the default email server.

Step 13  Define the separator between the email and server names. The default is @.

---

**Configure Web email: MS Outlook Web App**


---

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.

Step 4

Step 5

**What To Do Next**