Linux Meets Windows CA

Bridges

Microsoft's Certificate Enrollment Web Service offers an easy way to obtain X.509 certificates from Active Directory Certificate Services. We introduce the protocols and investigate how to use the certmonger tool to issue certificates for Linux systems. By Thorsten Scherf

The Certificate Enrollment Web Service was introduced in Windows Server 2008 R2 to modernize certificate requests and make them more flexible. Unlike traditional requests by Remote Procedure Call (RPC) and Distributed Component Object Model (DCOM) protocols, which require a direct connection to internal network ports and domain membership, both Certificate Enrollment Policy (CEP) web service and Certificate Enrollment Web Service (CES) are implemented on the Simple Object Access Protocol (SOAP) standard, which allows certificate requests to be made over an HTTPS interface, facilitating the integration of systems that are not part of the Active Directory (AD) domain or even reside on remote networks.

Two Central Services

The CEP web service is based on X.509 CEP (MS-XCEP) [1] and is used to provide clients with information about available certificate templates and certification authorities. The service provides this information over an HTTPS interface. Authentication is handled either by Kerberos with a username/password combination, or it relies on a client certificate.

In contrast, the CES web service is based on the WS-Trust X.509v3 Token Enrollment Protocol (MS-WSTEP) [2] – a Microsoft-specific implementation of the OASIS WS-TRUST [3] standard. It is responsible for requesting the certificate, which it does by forwarding certificate signing requests (CSRs) to the certification authority (CA). As with CEP, communication takes place over HTTPS, and authentication is identical to the CEP protocol.

Managing Certificates with certmonger

The `certmonger` tool [4] helps with all the tasks related to managing X.509 certificates on Linux systems, which means everything from generation of private keys, through certificate requests (CSRs), to automatic renewal of certificates before they expire.

The `cepces` [5] plugin lets you use CEP/CES to procure a certificate from AD Certificate Services (CS) and place it under the control of `certmonger`. This function is used by Samba to provision certificates automatically for clients (Certificate Auto Enrollment) with a Group Policy Object (GPO) [6].

To ensure that communication with AD CS over CEP and CES protocols works, make sure the Certificate Enrollment Web Service and Certificate Enrollment Policy Web Service roles are installed on an AD system, in addition to the Certificate Services. If these roles are not available, you can discover online how to add the roles to your existing AD CA [7][8].

Requesting a Certificate

The following example is based on a current Fedora system, but it also works on all other Linux systems on which the `certmonger` tool and the `cepces` plugin are available. As usual, the two packages are installed on Fedora

Photo by Sandip Roy on Unsplash

by the `dnf` package manager from the distribution's standard repository:

```
dnf install certmonger cepces-certmonger
```

The package manager automatically adds the Cepces CA plug-in to the `certmonger` configuration. To verify that this install worked, use:

```
getcert list-cas

[...]
CA 'cepces':
  is-default: no
  ca-type: EXTERNAL
  helper-location: ➥
    /usr/libexec/certmonger/ cepces-submit
```

In addition to several other plugins, you should now see a CA named *cepces* in the command output. If this entry does not appear, simply add the plugin manually by typing the command:

```
getcert add-ca ➥
  -c cepces ➥
  -e '/usr/libexec/certmonger/➥
      cepces-submit'
```

In the `/etc/cepces/cepces.conf` configuration file, the next step is to enter

```
grep '^server' /etc/cepces/cepces.conf
server=ad1-1g7p.win2022-1g7p.test
```

to find the name of the AD system on which you previously installed the CEP and CES roles.

Into the Domain with realmd

Before you can request a certificate from AD CS, you first need to add the client system to the domain, which might seem a little surprising because CEP and CES support different authentication methods. Unfortunately, the `certmonger` plugin currently only uses Kerberos to log in to an AD system. The easiest way to add the client to the AD domain is to use the `realmd` tool [9]. The package is available for most Linux distributions. Once the package is installed on the system, the first step is to perform a domain discovery:

```
realm discover win2022-1g7p.test
```

Make sure you use the server that has the AD DNS entries as the DNS resolver [10]. After making sure this worked, add the system to the domain:

```
realm join win2022-1g7p.test
```

A simple `id` command lets you verify that you can query users from the domain; then finally, test the authentication:

```
id Administrator@win2022-1g7p.test
kinit Administrator@win2022-1g7p.test
```

If everything worked, you can now manually request the certificate for your system:

```
getcert request ➥
  -c cepces ➥
  -k /etc/pki/tls/private/machine.key ➥
  -f /etc/pki/tls/certs/machine.crt
```

Type the `-c` option here to use the previously installed `cepces` plugin. If everything worked, you will see from the output of `getcert list` that a certificate was issued, and the system journal will also display information about a successful certificate issuance. You can also use `openssl` to query the certificate's details (Listing 1).

Conclusion

The `certmonger` tool and `cepces` plugin make it very easy to obtain certificates from an AD CS if the CEP and CES CA features are available. Currently, the client must be a domain member, because `certmonger` only supports Kerberos for authentication. However, this situation could change in future versions of the tool. Alternatively, you could check with `curl` and `openssl` whether additional wrappers or manual requests let you log in with a certificate or password. At the end of the day, though, the combination of CEP, CES, and `certmonger` offers a very useful approach for automated certificate requests in heterogeneous environments. ■

Info

[1] MS-XCEP: [https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-xcep/08ec4475-32c2-457d-8c27-5a176660a210]

[2] MS-WSTEP: [https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-wstep/4766a85d-0d18-4fa1-a51f-e5cb98b752ea]

[3] WS-TRUST: [https://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html]

[4] certmonger: [https://pagure.io/certmonger]

[5] cepces on GitHub: [https://github.com/openSUSE/cepces]

[6] Certificate Auto Enrollment: [https://wiki.samba.org/index.php/Certificate_Auto_Enrollment]

[7] Configuring the Certificate Enrollment Web Service: [https://learn.microsoft.com/en-us/windows-server/identity/ad-cs/configure-certificate-enrollment-web-service]

[8] Configuring the Certificate Enrollment Policy Web Service: [https://learn.microsoft.com/en-us/windows-server/identity/ad-cs/configure-certificate-enrollment-policy-web-service]

[9] realmd: [https://www.freedesktop.org/software/realmd/]

[10] realmd and AD: [https://www.freedesktop.org/software/realmd/docs/guide-active-directory.html]

The Author

Thorsten Scherf is the global Product Lead for Identity Management and Platform Security in Red Hat's Product Operations group. He is a regular speaker at various international conferences and writes a lot about open source software.

Listing 1: Certificate Details

```
# openssl x509 -in /etc/pki/tls/certs/machine.crt -noout -issuer -subject -dates
issuer=DC=test, DC=win2022-1g7p, CN=win2022-1g7p-AD1-1G7P-CA
subject=CN=client.win2022-yn6a.test
notBefore=May 30 10:18:31 2025 GMT
notAfter=May 30 10:18:31 2026 GMT
```