Certificate handling with SCEP

Short Guide

Last edited: 2023-07-21

1. Certificate handling with SCEP

Certificate handling with SCEP

SCEP is a protocol designed to simplify secure and scalable issuing of certificates to network devices in large-scale environments.

With SCEP, the devices take care of their certificates themselves. To do so, they require a one-time password created by an authorized person. With such a time-limited password (Windows: 60 minutes), a device can request a certificate from the SCEP service.

To prevent the certification authority (CA) from issuing arbitrary certificates with the one-time password, create a template in which you restrict the certificate classes.

By using SCEP you can simplify certificate issuing and management for uses such as

- 802.1X authentication for LAN and WLAN
- VPN connection via Cisco AnyConnect
- Connection via Citrix NetScaler Gateway

The eLux implementation is based on the **OpenSCEP** project. The following description requires the use of Windows NDES.

2. Windows as SCEP server

Note
For Windows Server 2012 and later versions, NDES (Network Device Enrollment Service) is integrated in the Certification Authority (CA).
The described procedure is to be seen as an example and can vary depending on the version and environment.

Installing NDES

1. Install the server role **AD CS** with the **Network Device Enrolment Service** feature .
2. Set up a service account for NDES which must be authorized on the CA later on.
3. Enter the information for the certificate from the Registration Authority (RA) to issue a signing certificate for this enrollment process.
4. Define the key length.

The new virtual directories are displayed in the IIS console below the default website.

Configuring templates

The SCEP server uses templates for submission to the CA.

1. Set the LDAP name of the template in the registry:

   [HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\MSCEP

Default:	`IPSecIntermediateOffline`
EncryptionTemplate	<*templatename*>
GeneralPurposeTemplate	<*templatename*>
SignatureTemplate	<*templatename*>

2. Add the new template to the CA.
3. Configure the use of a one-time password in the registry (validity period, maximum number).

The one-time password allows a client that supports SCEP/NDES to contact the SCEP server directly and request a certificate. The client connects to the Windows SCEP server with the URL http://<CA_FQDN>/certsrv/mscep/mscep.dll

Configuring one-time password

Either

- disable passwords (option A):
  [`HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\MSCEP\EnforcePassword`]
  set to `0`

or

- define fixed password (option B):
  [`HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\MSCEP\UseSinglePassword`]

Permissions for NDES service account

- Set the permission to `full control`.

Modify SCEP application pool defaults

- Set **Load User Profile** option from `false` to `true`.

Requesting one-time passwords

1. Sign in to the SCEP Administrator website using your NDES service account:
   `http://<CA_FQDN>/certsrv/mscep_admin/`
2. Copy the challenge password to the configuration file `scep.ini`

3. Configuring SCEP for eLux

How to enable eLux devices to use SCEP for certificate deployment:

Note
The eLux package **Network Access Control** and the included feature package **SCEP** must be installed on the devices.This may require modifications of the image definition file on the web server via ELIAS.

1. To configure the SCEP agent, use the `scep.ini` file.Then select the Scout feature Files configured for transfer to transfer the configuration file to the devices.For further information, see Configuring SCEP ini file.
2. Ensure that the devices are running on the correct time.We recommend that you configure a time server.

   The local time must match the time of the certificate authority.
3. To activate the SCEP agent, add the following entry to the `terminal.ini` file on the devices.To do so, use the Scout feature **Advanced file entries**:

File	`/setup/terminal.ini`
Section	`Network`
Entry	`SCEP`
Value	`true`

   Please note case sensitivity.
   For further information, see Advanced file entriesin the **Scout** guide.
4. To use SCEP for 802.1X authentication for LAN, in the Scout Console, for the relevant OU, in the **Network** device configuration for the LAN profile, select **Advanced > IEEE 802.1X authentication > Activate**.For further information, see Configuring 802.1X for eLux in the **802.1X** guide.

   When using SCEP for other purposes, however, make sure that the above-mentioned option is not selected for the network profile **LAN**.

The client creates a WPA configuration file from the template `/setup/scep/wpa.conf.scep`.However, if you have transferred an individual WPA configuration file to `/setup/scep/wpa.conf`, it will take precedence over the template.

3.1. Certificates for SCEP

The SCEP agent stores the following certificates in `/setup/cacerts/scep` by default.

client.pem	Client certificate	
client.key	Private key of client certificate	`2048` or `4096`[1] RSA key may be stored TPM 2.0 chip
serverca.pem	Server CA certificate	
serverca.key	Server CA key	required renewal of certificates
serverca.sig	Server CA signature	required renewal of certificates
serverra.pem	RADIUS server certificate	

Devices with TPM 2.0

The authentication procedures via WPA supplicant for wired Ethernet connections and via IEEE 802.1X for WLAN components consider both storage options for the SCEP private key – the file system and the TPM 2.0 chip.

Client certificates continue to be stored in the file system. The roll-out procedure, with which the devices initially receive the client certificates, is carried out exclusively via wired Ethernet connections. Existing certificates can be renewed both via wired Ethernet connections and via WLAN.

Also the private key file of the SCEP client certificate is stored in the file system, by default. It is located under `/setup/cacerts/scep` or in the directory you have defined in the `scep.ini` file. When you perform an eLux firmware update to a version with TPM 2.0 support, existing private key files of the SCEP client certificate remain stored in the file system.

Private key in TPM 2.0 chip

- To use the TPM 2.0 module for the private key, make the following Advanced file entry:

File	`scep.ini`	
Section	`Certificate`	
Entry	`UseTPM2`	
Value	`true`	Default is `false`.

This parameter ensures that the private key is generated for each TPM 2.0 device in its TPM module.[2] It is stored there only and cannot be displayed or exported. The key never leaves the TPM chip of the device.

When the certificate is renewed, the private key is reused.

---

[1] from eLux RP 6 2204
[2] from eLux RP 6 2101

The SCEP agent creates a certificate signing request (CSR) via openssl with the public key of the TPM module. Each time a connection is set up to an 802.1X-secured port, the system checks for a valid certificate. The certificate must be signed with the private key of the TPM chip. If the certificate does not exist or is invalid, the connecting process is rejected.

The `client.key` file remains in the file system but only contains information on the public key. The retained file name allows devices without a TPM to use the same certificate.

To delete the key from the TPM 2.0 chip, delete the TPM in the BIOS. We recommend that you first revoke certificates that are still valid.

3.2. Configuring the SCEP ini file

Note
You can use the example file on the devices to configure the SCEP agent:
`/setup/scep/scep.ini.sample`

1. Create the text file `scep.ini`. Then insert the sections and make the entries described below.
2. To transfer the configuration file `scep.ini` to the devices under `/setup/scep/`, use the Scout feature Files configured for transfer. For further information, see Advanced device configuration > Files in the **Scout** guide.

The certificate request is created from the `scep.ini` data. You can optionally extend the certificate request with additional attributes. For further information, see "Extended SCEP certificate request" on page 15.

Sections and entries for `scep.ini`

Section	Entry	Description	Example
Admin	URI	SCEP server address (URI)	`URI=http://ca.w2k12.sampletec-01.com/certsrv/mscep/`
	PROXY	Proxy server (optional)	`PROXY=proxy.sampletec-01.com:3800`
	ReNew	Number of days before the certificate expires and the device requests a new certificate (time span for renewal) From this day on, the client tries to renew the certificate. The value must be less than the validity period of the certificate.	`ReNew=30` (default)

Section	Entry	Description	Example
	ReNewCheckOnlyClient[1]	Alternative option to `ReNew`: Unlike `ReNew`, only the client certificate is checked, but not CA and RA certificates	`ReNew=30` (Standard)
	ExpireCheck	Time interval in days, how often the ReNew date is checked	`ExpireCheck=1` (default)
	challengePassword	One-time password for the request Valid for 60 minutes once only The password is deleted after the certificates have been successfully transferred.	(A): `challengePassword=12345` (B): `challengePassword=<password requested via http://CA_FQDN/certsrv/mscep_admin/>`

[1] from eLux RP 6 2302.1000

Certificate	CNTYPE	Type (`autoip|ip|dns|autodns|dnsfqdn|email`) `autoip` The IP address of the device is used as **CN**. `ip` The IP address you specify as **CN** in the `scep.ini` (see next option) is used. `dns` The name you specify as **CN** in the `scep.ini` (see next option) is used. `autodns` The host name specified in the `terminal.ini` is used as **CN**. `dnsfqdn` The host name specified in the `terminal.ini` with the domain name appended is used as **CN**. `email` The email address you specify as **CN** in the `scep.ini` (see next option) is used.	`CNTYPE=email`
	CN	Certificate/Name Is filled by the system for `CNTYPE=autoip|autodns|dnsfqdn`	`CN=userxxx@sampletec-01.com`

	OU
	ORGANIZATION
	LOCALITY
	STATE
		Certificate/Attribute (optional)	`OU=TestLab`
			`ORGANIZATION=SampleTec`
			`LOCALITY=Karlsruhe`
			`STATE=BW`

	OU1 OU2 OU3 OU4 OU5 OU6	Further attributes for up to 6 OUs (optional) The OUs specified here can be used for a certificate request.	OU1=Testlab OU2=KA_QA OU3=KA_DEV

	COUNTRY	Certificate/Country	`COUNTRY=DE`
	KEYLEN	Key length (from certificate template) Allowed values: `2048` or `4096` An incorrect key length leads to the event log entry **Incorrect Challenge Password** If a 4096 key cannot be generated in the TPM (for TPM 2.0 and **UseTPM2**), as a fallback a 4k RSA key is generated in the file system.	`KEYLEN=2048`
	CertStore	The certificate store can be freely selected. This parameter allows you to specify a new directory that the SCEP agent creates on the device. If you specify the default path `/setup/cacerts/scep`, the private key for the SCEP client certificate for TPM 2.0 devices is stored in the TPM 2.0 module.	`CertStore=/setup/cacerts/scep`

Additional information

- The storage location can be defined individually for each certificate file.

  The three file storage parameters have higher priority than the **CertStore** parameter.
  The specified directories must exist.
  Client certificate files can be merged.

- Access rights can be defined for the private key (if in file system).
- Certificates can be checked for validity according to the certificate revocation list (CRL)[1]

Certificate	CA_Path	Path and file name of the server CA certificate	`CA_Path=/setup/cacerts/scep/serverca.pem` (default)
	Client_Path	Path and file name of the client certificate	`Client_Path=/setup/cacerts/scep/client.pem` (default)
	ClientKey_Path	Path and file name of the private key file of the client certificate	`ClientKey_Path=/setup/cacerts/scep/client.key` (default)

Note
Specify the same value for **Client_Path** and **ClientKey_Path**, for example `/setup/cacerts/scep/client.pem`, if you want to merge the two files.

	RA_Path	Path and file name of the RADIUS server certificate	`/setup/cacerts/scep/serverra.pem` (default)
	MODE	For additional use of a RADIUS server certificate: `RA` Allowed values: `CA` (default), `RA`	`CA`
	ClientKey_Permission	Access rights for the private key file `0400` Can only be read with root rights (default) `0444` Can also be read by users	`ClientKey_Permission=0400`

[1] from eLux RP 6 2101

UseTPM2	For devices with TPM 2.0 chip From eLux RP 6 2101: The private key is generated in the TPM 2.0 module and remains there. eLux RP 6.10 and 11: The private key file is stored in the TPM 2.0 chip. Allowed values: `false` (default), `true` For further information, see "Certificates for SCEP" on page 6.	`UseTPM2=true`
TPM2Fallback[1]	If the private key cannot be stored in the TPM 2.0 module (for example, on devices with other TPM versions), it is stored in the file system. Allowed values: `true` (default), `false` `false`: There is no fallback. On devices without TPM 2.0, the key is not stored at all.	`TPM2Fallback=true`
CrlCheck[2]	Optionally, specify up to five certificates (in addition to CA_Path, Client_Path and RA_Path) you want the SCEP agent to check for validity according to CRLs. The check is done each time a connection is set up. If the connection remains active for longer, the check is carried out after the time interval you set under `ExpireCheck`.	`CrlCheck1=/setup/cacerts/myrootca.pem` `CrlCheck2=/setup/cacerts/myintermediate.pem`

[1] from eLux RP 6 2204
[2] from eLux RP 6 2101

CrlCheckEnabled[1]	`true` (default): All certificates are checked for validity according to CRLs `false`: No revocation check is done.	`CrlCheckEnabled=true`
FirstCrlOnly[2]	`true`: If multiple CRLs are registered, only the first one is downloaded and checked. This reduces the load on the PKI infrastructure. `false` (default): All existing CRLs are downloaded and checked.	`FirstCrlOnly=true`

[1] from eLux RP 6 2110
[2] from eLux RP 6 2204

3.3. Extended SCEP certificate request

Note
Use this function to configure the SCEP agent with additional attributes that are not included with the `scep.ini` file.

The certificate request is created from the values in the `scep.ini` file and is available on the device in the file `/setup/scep/clientreq.conf.in`. This file is installed with SCEP as a template and contains the following entries by default:

```
[req]
prompt=no
distinguished_name=req_distinguished_name
string_mask=nombstr
attributes=req_attributes

[req_attributes]
challengePassword=__CHALLENGEPASSWORD__

[req_distinguished_name]
C=__COUNTRY__
ST=__STATE__
L=__LOCALITY__
O=__ORGANIZATION__
OU=__OU__
1.OU=__OU1__
2.OU=__OU2__
3.OU=__OU3__
4.OU=__OU4__
5.OU=__OU5__
6.OU=__OU6__
CN=__CN__

[__X509V3__]
subjectAltName=critical,__CNTYPE__:__ALTNAME__
```

The fields marked with underscores are parameters and are replaced by the relevant values in `scep.ini`.

If the attributes provided in the `scep.ini` are sufficient for your purposes, there is no need to edit the `clientreq.conf.in` file.

Extending certificate request with additional attributes

1. Retrieve the `/setup/scep/clientreq.conf.in` file installed with SCEP from a device, for example by using the **Diagnostics** feature.

2. Edit the file. Add any other openSSL sections and attributes. For further information, see https://www.openssl.org/docs/man1.0.2/man1/openssl-req.html
3. Transfer the file to the relevant devices under `/setup/scep/clientreq.conf.in` by using the Scout feature **Files configured for transfer**.

Note
This file - not the `scep.ini` - is the source for the certificate request. Only delete fields that you do not need!.

---

4. Diagnosis for SCEP

Show client certificate in a shell

- Use the following command:
  `openssl x509 -in /setup/cacerts/scep/client.pem -noout -text`

All information about the certificate is displayed.

View log files

1. For the relevant OU, in the device configuration, enable enhanced logging.
2. Request the diagnostic files. For further information, see Requesting diagnostic files in the **Scout** guide.

`/tmp/systemd-journal.log`	Network log file
`/setup/logs/scepagent.log`[1]	Log file of the SCEP agent, contains last certificate transfer This file is not included in the template `#System`, but must be defined in an individual template. For further information, see Device diagnostics in the **Scout** guide.

[1] temporary version under `/var/log/scepagent.log`