KEYFACTOR

2024 | Keyfactor

Keyfactor SCEP 1.7

Installation and Configuration Guide

Table of Contents

List of Tables

List of Figures

1.0 Introduction

The Keyfactor implementation of the Simple Certificate Enrollment Protocol (SCEP) can be used wherever a SCEP server would be used (see https://datatracker.ietf.org/doc/id/draft-nourse-scep-23.txt for more information). Keyfactor's SCEP server implementation can function in an *Intune-gated* mode, where the SCEP server will validate every incoming enrollment against the customer's Intune instance, using a Microsoft-proprietary API and protocol. Microsoft Intune is a cloud-based service that supports policies to control applications and help keep employees productive and secure in either a corporate or bring-your-own-device (BYOD) scenario. Keyfactor customers routinely choose to utilize it with Microsoft Intune for mobile device management (MDM).

More information about Intune, including a brief overview in Microsoft's architectural document, is available on the Microsoft documentation site at:

https://docs.microsoft.com/en-us/mem/intune/fundamentals/what-is-intune

https://docs.microsoft.com/en-us/mem/intune/protect/certificate-authority-add-scep-overview

https://docs.microsoft.com/en-us/mem/intune/protect/scep-libraries-apis

The SCEP protocol allows devices to enroll for a certificate by using a URL and a shared secret to communicate with a PKI. The role runs under Microsoft IIS and requires at least one Microsoft CA on the back end. Although the Keyfactor SCEP implementation and the Microsoft implementation of SCEP (NDES) can be collocated on the same server, there is no need to install NDES to support the Keyfactor SCEP install. The Keyfactor SCEP server has no dependence on Microsoft's NDES.

1.1 SCEP Server Architecture & System Requirements

The components that make up a Keyfactor SCEP implementation include:

- Keyfactor SCEP Server

  This server runs under IIS and requires ASP.NET (4.5 or higher).
- Certificate authority

  For issuing certificates.
- Keyfactor Command

  For managing, monitoring, and reporting on certificates.
- Microsoft Intune

  Optionally, for managing delivery of certificates to devices.
- Web Reverse Proxy
- Optionally, if devices will be contacting the SCEP server from outside the enterprise firewall.

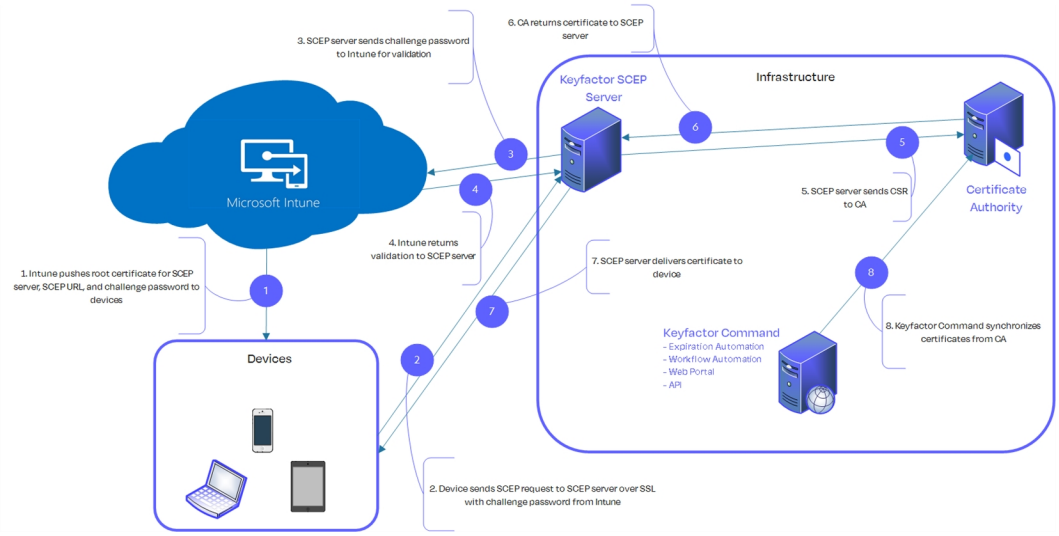

Figure 1: Certificate Request Flow with Intune

The system requirements for the Keyfactor SCEP server are:

- Window Server 2016, or Windows Server 2019
- Internet Information Services (IIS) with Windows Authentication (assuming you plan to use Windows authentication to authenticate for challenge passwords)
- ASP.NET 4.5 or greater
- Microsoft Windows Update KB 3118401 if applicable
- .NET Framework 4.6.2 or greater
- Minimum of 2 GB RAM
- Minimum of one 2 GHz CPU
- Minimum of 20 GB disk space

2.0 Preparing for the SCEP Server

This section describes the steps that need to be taken prior to the SCEP server installation to install the prerequisites, create the required supporting components, and gather the necessary information to complete the SCEP installation and configuration process.

2.1 Install IIS and .NET on the SCEP Server

On the server that will host the SCEP server, install Internet Information Services (IIS) and the .NET Framework version 4.6.2 or greater.

IIS is a role on all supported versions of Windows. In addition to the default components of IIS, the SCEP server requires these role services:

- Windows Authentication
- ASP.NET

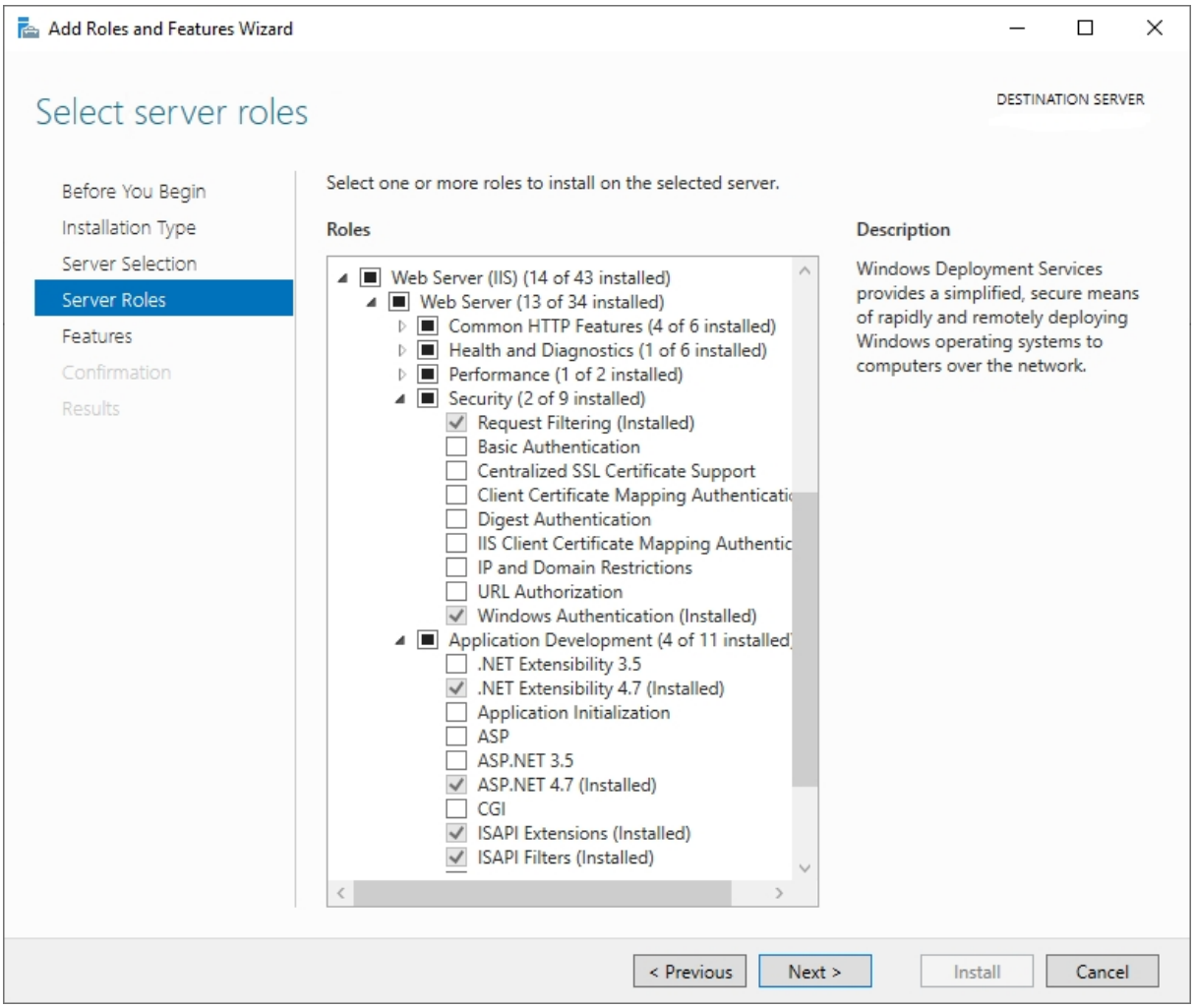

Figure 2: Required IIS Role Services for SCEP

Windows Server 2016, and Windows Server 2019, it is a standard Windows feature added through the Windows Server Manager tool.

To verify the version of .NET installed:

1. Open the Registry Editor:

   regedit

2. Navigate to:

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\Full

3. If the **Release** attribute value is not at least 394806 or 394802, then install .Net Framework

4.6.2 or higher.

Table 1: .NET Framework Release Values

.NET Framework	Release Value
.NET Framework 4.6.2	394806 or 394802
.NET Framework 4.7	460805
.NET Framework 4.7.1	461310
.NET Framework 4.7.2	461814
.NET Framework 4.8	528040

2.2 Create Service Accounts for the SCEP Server

The SCEP server makes use of one service account under which the IIS application pool for SCEP runs. An Active Directory service account is generally used. The service account needs to be created prior to installation of the SCEP server software, and the person installing the SCEP server software needs to know the domain, username and password of the service account.

2.3 Enable the Required Templates for SCEP Infrastructure

The SCEP server needs two certificates to allow it to operate—one for encryption and one for signing. During the configuration process, you may enroll for the required SCEP certificates automatically from the SCEP configuration tool. If you do this, the SCEP server requests one certificate based on the *CEP Encryption* Microsoft CA certificate template to be used for encryption and one certificate based on the *Exchange Enrollment Agent (Offline request)* certificate template to be used for signing. These standard Microsoft templates must be available for enrollment from at least one CA in the environment during the configuration if the automated certificate request option will be used.

Note: The CA used to request the SCEP infrastructure certificates does not need to be the same CA that will be used to issue the SCEP enrollment certificates.

The built-in Microsoft templates have a 1024-bit key size which cannot be modified through the standard template management interface. Certificates you acquire using these standard templates will therefore have 1024-bit keys. If you would prefer to use SCEP certificates with stronger keys, you will need to first create templates to allow you to enroll for those certificates, enroll for the certificates, and configure SCEP to use certificates.

In addition to the templates for infrastructure certificates required by the SCEP server, you also need to identify a template that you will configure for SCEP enrollment certificates. This template will be selected in the SCEP server configuration tool.

2.3.1 Enable the Built-in SCEP Templates

To enable the built-in *CEP Encryption* and *Exchange Enrollment Agent (Offline request)* certificate templates on a CA:

1. On the CA that will issue the SCEP infrastructure certificates, open the Certification Authority management tool.

2. In the Certification Authority management tool, right-click the **Certificate Templates** folder and choose **New->Certificate Template to Issue**. Select the **CEP Encryption** template from the list presented and click **OK**.

3. In the Certification Authority management tool, right-click the **Certificate Templates** folder and choose **New->Certificate Template to Issue**. Select the **Exchange Enrollment Agent (Offline request)** template from the list presented and click **OK**.

2.3.2 Create Custom SCEP Templates (Optional)

This step is optional. If you prefer to use the built-in *CEP Encryption* and *Exchange Enrollment Agent (Offline request)* certificate templates (which support a maximum key size of 1024 bit), you may skip this step.

To create custom templates as replacements for the built-in templates to support stronger keys:

1. On the CA that will issue the SCEP certificates, open the Certification Authority management tool.

2. In the Certification Authority management tool, drill down to locate the **Certificate Templates** folder. Right-click the Certificate Templates folder and choose **Manage**. This will open the Certificate Templates Console.

3. In the Certificate Templates Console, right-click the **CEP Encryption** template and choose **Duplicate Template**.

4. If prompted with a Duplicate Template dialog (some versions of Windows), choose Windows Server 2003 Enterprise and click **OK**.

5. General Tab: In the Properties of New Template dialog on the General tab, enter **Keyfactor SCEP Encryption** (or an alternate name of your choosing) in the **Template display name** field. The **Template name** will be auto-populated based on the text you enter in the **Template display name**. Select a validity period for the certificate that's appropriate for your environment.

6. Cryptography Tab: In the Properties of the New Template dialog on the Cryptography tab, set a **Minimum key size** that's appropriate for your environment (generally 2048).

7. Security Tab: In the Properties of New Template dialog on the Security tab, grant the SCEP server machine account **Read** and **Enroll** permissions on the template if you plan to enroll for the certificates using the Microsoft certificates MMC or the appropriate user or group if you plan to enroll for the certificates using the Keyfactor Command Management Portal.

8. Click **OK** to save the new template.
9. In the Certificate Templates Console, right-click the **Keyfactor SCEP Encryption** template you just created and choose **Duplicate Template**.
10. If prompted with a Duplicate Template dialog (some versions of Windows), choose Windows Server 2003 Enterprise and click **OK**.
11. General Tab: In the Properties of New Template dialog on the General tab, enter **Keyfactor SCEP Signing** (or an alternate name of your choosing) in the **Template display name** field. The **Template name** will be auto-populated based on the text you enter in the **Template display name**. Select a validity period for the certificate that's appropriate for your environment.
12. Request Handling Tab: In the Properties of the New Template dialog on the Request Handling tab, change the **Purpose** of the template from Encryption to Signature.
13. Extensions Tab: In the Properties of New Template dialog on the Extensions tab, review the configuration for Key Usage and confirm that both **Digital signature** and **Signature is proof of origin (nonrepudiation)** are checked and no other options are configured.
14. Click **OK** to save the new template.
15. Back in the Certification Authority management tool, right-click the **Certificate Templates** folder and choose **New->Certificate Template to Issue**. Select the **Keyfactor SCEP Encryption** and **Keyfactor SCEP Signing** templates from the list presented and click **OK**.

2.3.3 Create the SCEP Certificates

The Keyfactor SCEP server needs one encryption certificate with private key and one signing certificate with private key in the local machine store on the SCEP server. Once you have made the templates available for enrollment, you may go about acquiring certificates using these templates in whatever way is easiest for you.

This step assumes that you are using custom templates created as per Create Custom SCEP Templates (Optional) on the previous page. If you are using the built-in *CEP Encryption* and *Exchange Enrollment Agent (Offline request)* certificate templates, you can automatically acquire certificates as part of configuration process and may skip this step.

To acquire the certificates using the Microsoft certificates MMC:

1. On the SCEP server, do one of following:
    - Using the GUI:
        a. Open an empty instance of the Microsoft Management Console (MMC).
        b. Choose **File->Add/Remove Snap-in...**.
        c. In the *Available snap-ins* column, highlight **Certificates** and click **Add**.

d. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.

e. Click **OK** to close the Add or Remove Snap-ins dialog.

- Using the command line:
  a. Open a command prompt using the ”Run as administrator” option.

  b. Within the command prompt type the following to open the certificates MMC:

     ```
     certlm.msc
     ```

2. Drill down to the Personal folder under Certificates, right-click, and choose **All Tasks->Request New Certificate...**.

3. In the Certificate Enrollment Wizard, click **Next**.

4. On the Select Certificate Enrollment Policy page, accept the default and click **Next**.

5. On the Request Certificates page, scroll down to locate the **Keyfactor SCEP Encryption** template, and check the box for the template.

   If the template does not appear in the list, you may need to run a ”gpupdate /force” on the SCEP server to pick up the new template or you may need to verify that you granted the SCEP server machine account enroll permissions on the template.

6. On the Request Certificates page, click the link below the **Keyfactor SCEP Encryption** template name that says ”More information is required to enroll for this certificate...”. On the Subject tab of the Certificate Properties dialog, select **Common name** in the **Type** dropdown under **Subject name**, enter a name for the certificate in the **Value** field, and click the **Add** button. No specific text is required in the subject name. This name is for your reference and to clarify the purpose of the certificate—e.g. Keyfactor SCEP Server Encryption. Click **OK** at the bottom of the Certificate Properties dialog.

7. On the Request Certificates page, click **Enroll** to enroll for the certificate and Finish when the enrollment is complete.

8. Repeat steps 5 through 10 using the **Keyfactor SCEP Signing** template to acquire a second certificate.

9. In the Certificates MMC, drill down to the Certificates folder under Personal, right-click the Keyfactor SCEP Server Encryption certificate, and choose **Open**. On the Details tab, locate the **Serial number** and copy the serial number from the box at the bottom of the dialog to a text file, making note that this is the encryption certificate serial number. Remove any spaces from the serial number so that the serial number string looks something like this:

   ```
   69000016e1ffccf7521125122a0000000016e1
   ```

Important: As displayed in the certificates dialog, the serial number has a narrow leading space that is actually an unprintable control character. If you accidentally copy this character and paste it into the registry setting when you are following the instructions in Installing the SCEP Server on page 12, the serial numbers will fail to appear in the Keyfactor SCEP Configuration tool. Be sure to strip off any leading spaces on the copied text.

10. Repeat step 12 for the Keyfactor SCEP Server Signing certificate.

Once the certificates for SCEP encryption and signing have been acquired, the SCEP server needs to be configured to use these certificates. This is done by changing a couple of registry settings. The registry configuration area for SCEP is created by the SCEP server installation, so the SCEP server will be configured to use the appropriate certificates during the server installation.

Note: If you encounter an error with a message similar to "Invalid algorithm specified" when enrolling for one or both of your certificates, try setting your template(s) on the Cryptography tab to Requests must use one of the following providers and select the two Microsoft Enhanced... providers.

2.4 Using a SQL Database for SCEP (Optional)

The Keyfactor SCEP server uses a database to store challenge passwords and configuration information. The default database that is implemented as part of the standard installation is a Microsoft Jet database stored on the local SCEP server. For SCEP deployments that require load balancing either due to traffic or availability requirements, the option is available to use a Microsoft SQL database for this role instead. Configuration to use a SQL database rather than a Jet database is done separately from the primary installation and configuration process—either before or after it—and is not featured in the configuration wizard.

To configure the SCEP server to store data in a SQL database:

1. On the SQL server, create an empty database for SCEP.
2. Grant the service account under which the SCEP application pool will run (see Create Service Accounts for the SCEP Server on page 5) at least db_datareader and db_datawriter permissions on the database.

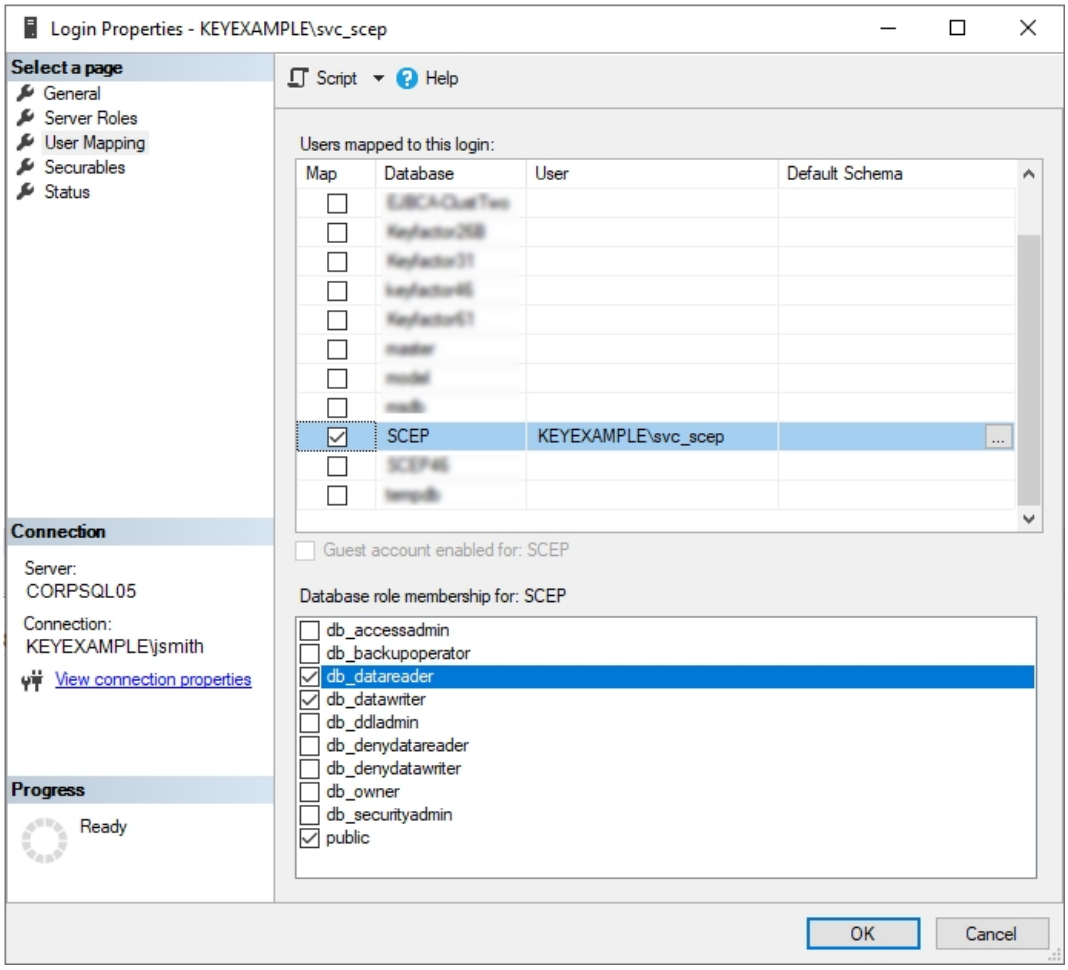

Figure 3: Set SQL Permissions

3. Install the SCEP server software but do not configure it.

4. Locate the *PopulateDatabase.sql* script in the Configuration directory under the installed SCEP server directory. By default, this file is located in the following directory:

   C:\Program Files\Keyfactor\Keyfactor SCEP Server\Configuration

5. Copy the *PopulateDatabase.sql* script to the SQL server and run it on the SCEP database you created above.

6. On the SCEP server, open a text editor (e.g. Notepad) using the ”Run as administrator” option.

7. In the text editor, browse to open the Web.config file for the SCEP server. By default, this file is located in the following directory:

   C:\Program Files\Keyfactor\Keyfactor SCEP Server\SCEP Server

8. In the Web.config file near the top of the file, find the connectionStrings section and in the ChallengeDB item, add a connection string that's appropriate for your environment, referencing the SQL server name, the SCEP database name , and the authentication method of *integrated Windows authentication*.

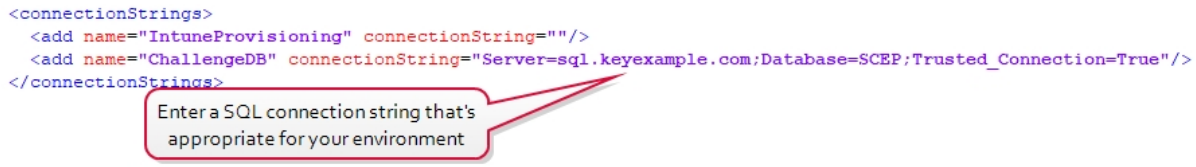

Figure 4: Enter a SQL Connection String

9. In the Web.config file in the appSettings section just below the connectionStrings section, find the *Keyfactor.SCEP.ChallengePassword.RepostoryType* key. Change the Value for this key from *ESENT* to *SQL*.

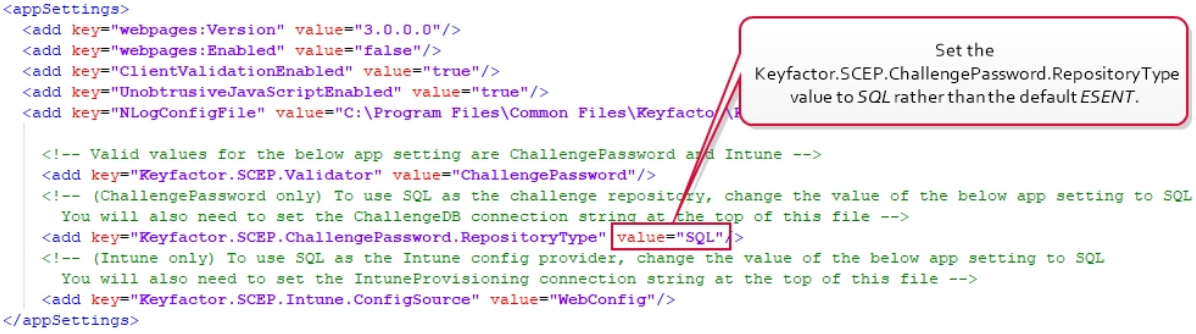

Figure 5: Select a Challenge Password Repository Type

10. Save the Web.config changes.

11. Return to the SCEP server installation (see Installing the SCEP Server on page 12) and finish the configuration.

If you prefer, you can fully complete the SCEP server installation and configuration using the built-in database and then switch over to using the SQL database at a later time. The SQL database configuration does not need to be done before the SCEP server is configured. However, configuring the SQL database settings prior to configuring the SCEP server will prevent the creation of the local Jet database.

3.0 Installing the SCEP Server

To begin the Keyfactor SCEP installation, execute the SCEPInstaller.msi file from the installation media and install as follows.

1. On the first installation page, click **Next** to begin the setup wizard.
2. On the next page, read and accept the license agreement and click **Next**.
3. On the next page, select the destination folder for the install. The default installation location is:

   C:\Program Files\Keyfactor\Keyfactor SCEP Server

4. On the next screen, click **Install**.
5. On the final installation wizard page, click **Finish**. The configuration tool should start automatically. This can take several seconds.
6. Optional: If you've opted to store SCEP challenges in a Microsoft SQL database, pause at this step and configure the SQL database (see Using a SQL Database for SCEP (Optional) on page 9) and then continue with the configuration steps.
7. Optional: If you opted to create custom templates so that you could generate SCEP encryption and signing certificates with keys greater than 1024-bit as per Enable the Required Templates for SCEP Infrastructure on page 5, you will need to configure SCEP to use those certificates using the registry before continuing with the rest of the standard configuration steps. To update the SCEP server to use the manually acquired encryption and signing certificates:
   a. Use the Registry Editor (regedit) to open the following configuration area:

      HKEY_LOCAL_MACHINE\SOFTWARE\Certified Security Solutions\SCEP Server-\Configuration

   b. In the Configuration key, create a **String Value** field and name it **EncryptionSerial**.
   c. Double-click to edit the **EncryptionSerial** configuration setting and paste in the serial number for the Keyfactor SCEP Server Encryption certificate that you made note of in Create the SCEP Certificates on page 7. Click **OK** to save.
   d. In the Configuration key, create a **String Value** field and name it **SigningSerial**.
   e. Double-click to edit the **SigningSerial** configuration setting and paste in the serial number for the Keyfactor SCEP Server Signing certificate that you made note of in Create the SCEP Certificates on page 7. Click **OK** to save.
8. In the Keyfactor SCEP Configuration tool in the SCEP Enrollment section of the page, select the CA from which certificates will be issued via SCEP in the **Enrollment CA** dropdown. Select the

certificate template in the **Enrollment Template** dropdown. Only templates available on the selected CA will be shown in the dropdown. In the **Challenge Type** dropdown, select Unique for best security. The option to use the same challenge password for every request (Single Challenge) or no challenge password at all (No Challenge) should only be used in environments where this functionality is absolutely necessary and where access to the SCEP server is strictly controlled. In the **Concurrent Challenges** box, enter a number that reflects the number of SCEP challenge requests you expect to receive, and challenges issue, in the **Challenge Lifetime**. In the **Challenge Lifetime (minutes)** box, enter a number of minutes for which SCEP challenges will be valid.

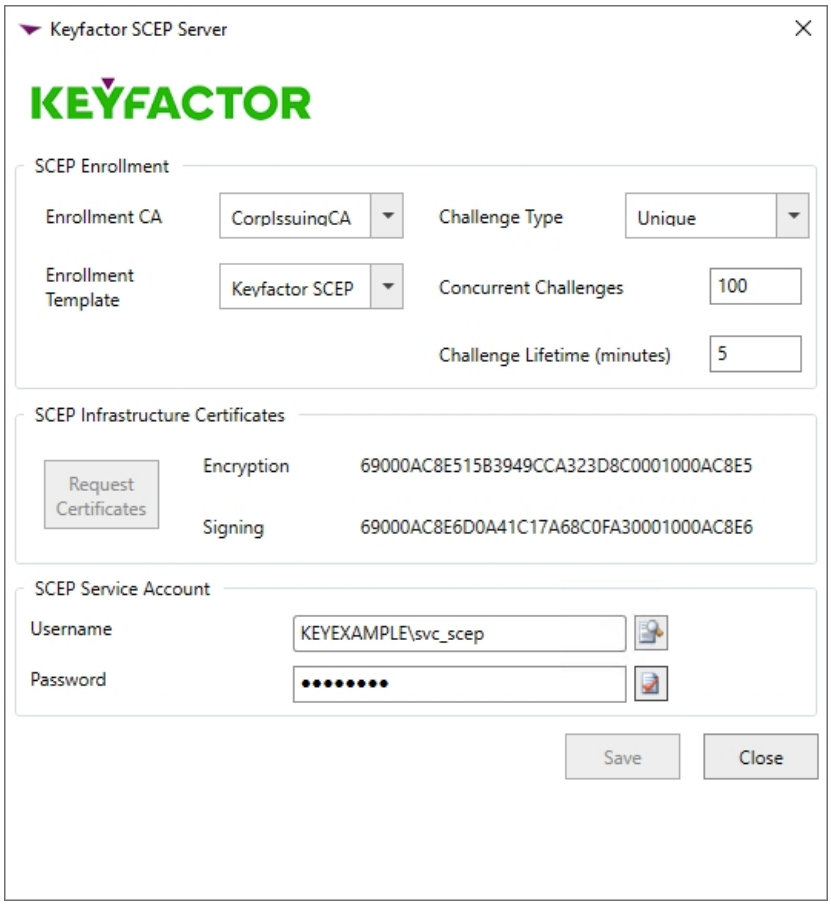

Figure 6: SCEP Configuration Tool

9. If you did not manually configure certificates as per step 7, in the Keyfactor SCEP Configuration tool in the SCEP Infrastructure Certificates section of the page, click the **Request Certificates** button to automatically request certificates for the SCEP server. If you manually configured certificates in step 6, you should see the serial numbers for your certificates in the SCEP Infrastructure Certificates section of the page. The SCEP server requests certificates using the *CEP Encryption* and *Exchange Enrollment Agent (Offline request)* templates and will scan through any available CA in the environment for a CA that is able to issue certificates based on

these templates, beginning with the CA you selected for SCEP enrollment. If no CA is available to issue certificates based on these templates, an error will occur. You will need to make the templates available for issuing on a CA in the environment and try the *Request Certificates* step again.

> **Tip:** If you have more than one CA with the SCEP infrastructure certificates templates available on it, wish to specify which of these CAs to request the certificates from, and have not selected this CA in the previous step, you can temporarily select the CA for the SCEP infrastructure CAs in the SCEP enrollment section of the page, request the infrastructure certificates, and then change the CA in the SCEP enrollment section of the page.

10. In the Keyfactor SCEP Configuration tool in the SCEP Service Account section of the page, enter the user name (DOMAIN\User format) and password of the Active Directory service account under which the SCEP application pool will run. You may use the people picker button ( ) to browse for the account. Click the verify button ( ) to confirm that the username and password entered are valid.

11. At the bottom of the configuration tool, click **Save** and then close the dialog.

4.0 Initial Configuration

Once the installation and configuration is complete, only a few configuration tasks remain before the SCEP server is ready to use. This section details the basic post-install configuration steps that need to be completed to get SCEP up and running.

4.1 Configure Kerberos Authentication

By default, the SCEP server uses integrated Windows authentication. Integrated authentication consists of both NTLM and Kerberos authentication types. In some environments, NTLM will work for integrated authentication and users will be able to acquire a SCEP challenge without further configuration. In other environments, NTLM will not work, so only Kerberos will be supported. Further configuration is required to make Kerberos authentication work correctly. Even if NTLM is supported, Kerberos is generally preferred for best security practice.

Common scenarios in which NTLM will not work are multi-domain forests and authentication attempts between domains and servers that support only NTLM2 using clients attempting NTLM.

Configuring the environment to support Kerberos includes these topics:

- Configure browsers to support Integrated Windows Authentication (for testing purposes)
- Configure the service principal name (SPN) for the Keyfactor Command server

4.1.1 Configure Browsers for Integrated Windows Authentication

Note that this step is only necessary for browsers where you will be testing the SCEP functionality from Windows. This does not need to be done on iOS devices.

To support integrated Windows authentication using either NTLM or Kerberos, the browser must be configured correctly to support this integration. This becomes particularly important when only Kerberos is used, as the browser won't allow the user to continue if Kerberos authentication fails, whereas with NTLM authentication, the integration won't work (the user will be prompted to enter a password), but the user will be allowed to continue to the SCEP server. Many modern browsers support integrated authentication. Some will work with Kerberos without any additional client side configuration (e.g. Chrome). Others require additional configuration to work (e.g. Internet Explorer and Firefox). The following instructions cover adding a trust for the SCEP server to Internet Explorer's local intranet zone to support integrated authentication. Configuring Firefox to support integrated authentication is beyond the scope of this guide.

To configure Internet Explorer to support integrated authentication:

1. In Internet Explorer, open Internet Options and go to the Security tab.
2. On the Security tab, highlight **Local intranet** and click **Sites**.
3. On the Local intranet sites popup, click **Advanced**.

4. On the Local intranet dialog, enter the fully qualified domain name of your Keyfactor Command server and click **Add**.

5. Click **Close** and **OK** until you have closed all the dialogs.

6. Exit Internet Explorer and open it again to attempt your authentication.

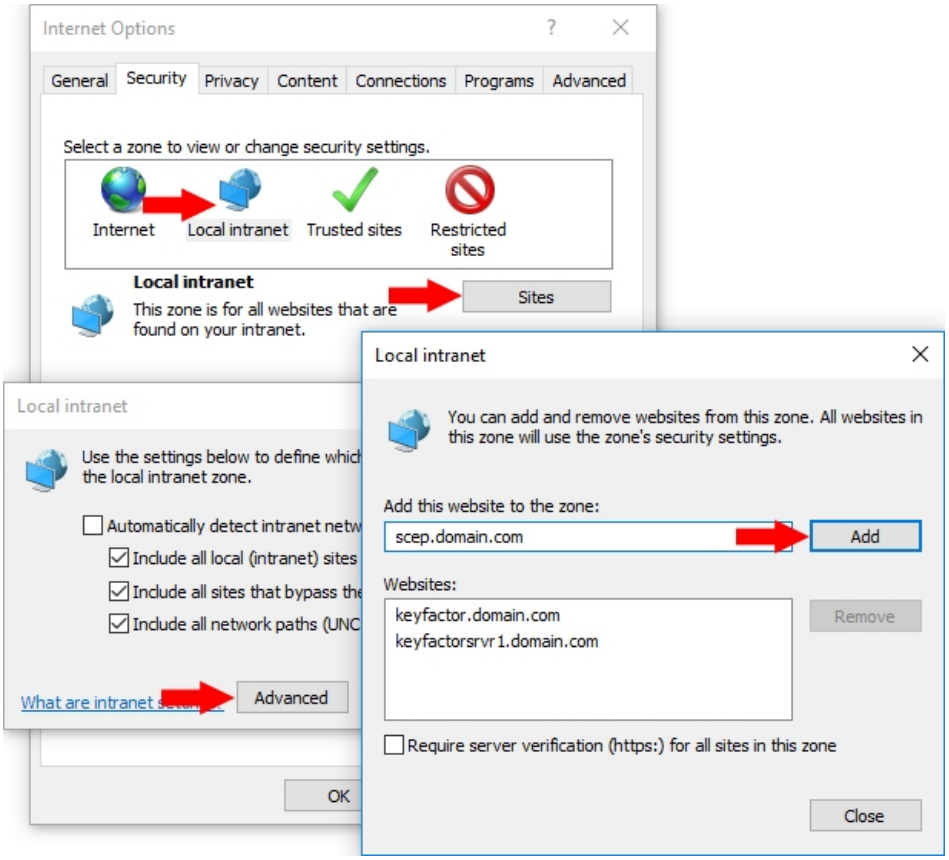

Figure 7: Configure Local Intranet Zone in Internet Explorer

Important: It is not sufficient to put the SCEP server in the **Trusted sites** zone. The server needs to be in the **Local intranet** zone for proper integrated authentication functionality (assuming these zones are still configured as per the default configuration).

4.1.2 Configure the Service Principal Name for the SCEP Server

On a server that has the setspn command available (typically it is available on domain controllers, as it installs as part of the Active Directory Domain Services role), open a command prompt using the "Run as administrator" option and run the following command (where *scepserver.keyexample.com* is the fully qualified domain name of your SCEP server or the DNS alias you are using to reference your SCEP server, if applicable, and *KEYEXAMPLE\svc_scep* is the domain name and service account name of the service account under which the SCEP application pool is running):

```
setspn -s HTTP/scepserver.keyexample.com KEYEXAMPLE\svc_scep
```

Important: If you are running the Keyfactor SCEP server on the Keyfactor Command server, wish to configure Kerberos authentication for both, and have chosen to run the two application pools with different service accounts, you will need to use a DNS alias to reference one or the other of these applications (or both) so that you can set the SPNs separately for the different service accounts. Setting the same SPN (e.g. HTTP/keyfactorserver.keyexample.com) on two different service accounts (e.g. KEYEXAMPLE\svc_keyfactorpool and KEYEXAMPLE\svc_scep) is not supported.

4.2 Configure Logging

By default, the SCEP server places its log files in the C:\CMS\logs directory, generates logs at the ”Info” logging level and stores logs for two days before deleting them. If you wish to change these defaults:

1. On the SCEP server where you wish to adjust logging, open a text editor (e.g. Notepad) using the ”Run as administrator” option.
2. In the text editor, browse to open the Nlog.config file for the SCEP server. The file is located in the following directory:

   ```
   C:\Program Files\Common Files\Keyfactor\Keyfactor SCEP Server
   ```

3. Your Nlog.config file may have a slightly different layout than shown here, but it will contain the four fields highlighted in the below figure. The fields you may wish to edit are:

   - fileName=”C:\CMS\logs\SCEP_Log.txt”

     The path and file name of the active SCEP server log file.

     If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\CMSLogs) and grant the service account under which the SCEP server application pool runs full control permissions on this directory.
   - archiveFileName=”C:\CMS\logs\SCEP_Log_Archive_{#}.txt”

     The path and file name of previous days’ SCEP server log files. The SCEP server rotates log files daily and names the previous files using this naming convention.
   - maxArchiveFiles=”2”

     The number of archive files to retain before deletion.
   - name=”*” minlevel=”Info”

     The level of log detail that should be generated. The default ”Info” level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to ”Debug” or ”Trace”. Available log levels (in order of increasing verbosity) are:

- OFF – No logging
- FATAL – Log severe errors that cause early termination
- ERROR – Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN – Log errors and use of deprecated APIs, poor use of APIs, ”almost” errors, and other runtime situations that are undesirable or unexpected but not necessarily ”wrong”
- INFO – Log all of the above plus runtime events (startup/shutdown)
- DEBUG – Log all of the above plus detailed information on the flow through the system
- TRACE – Maximum log information—this option can generate VERY large log files

```
<targets>
  <target name="buffered_wrapper" xsi:type="BufferingWrapper" slidingTimeout="true" bufferSize="500" flushTimeout="500">
    <target xsi:type="File" name="logfile" fileName="C:\CMS\logs\SCEP_Log.txt" layout="${longdate} ${logger} [${level}] - ${message}"
            archiveFileName="c:\CMS\logs\SCEP_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="2"/>
  </target>
  <target xsi:type="OutputDebugString" name="String" layout="${longdate} ${logger}::${message}"/>
  <target xsi:type="Debugger" name="debugger" layout="${longdate} ${logger}::${message}"/>
  <target xsi:type="Console" name="console" layout="${longdate} ${logger} ${message}"/>
  <target xsi:type="EventLog" name="eventLog" source="Certificate Management System"
          eventId="${query-string:item=eventID}" category="${query-string:item=categoryID}" layout="${query-string:item=eventMessage}" />
</targets>
<rules>
  <!-- Don't write events to the log file (log file should contain different, more verbose, logging) -->
  <logger name="*-EVENT" minlevel="Trace" writeTo="eventLog" final="true" />
  <logger name="*" minlevel="Info" writeTo="logfile" />
</rules>
```

Figure 8: NLog.config File

4.3 Test the SCEP Server

To test the SCEP server to confirm that it is correctly issuing SCEP challenges, visit the following URL from a browser that has been configured to support integrated authentication while logged in as a user with request permissions on the template configured for the SCEP server (where SCEP_SERVER_FQDN is the FQDN of your SCEP server):

https://SCEP_SERVER_FQDN/scep/challenge

Your SCEP server may have been configured to use HTTP rather than HTTPS.

4.4 Configure CA Redundancy (Optional)

As initially configured, the Keyfactor SCEP Server will direct SCEP certificate enrollment requests to a single CA. In many environments, this will be sufficient. For SCEP deployments that require load balancing either due to traffic or availability requirements, the option is available to configure additional CAs against which enrollment requests will be directed in the event that one or more CAs is unavailable. Configuration of multiple CAs for redundancy is done separately from the primary installation and configuration process and is not featured in the configuration wizard.

To configure the SCEP server to enable two or more CAs to respond to SCEP certificate enrollment requests:

1. On the SCEP server where you wish to add CAs, open the registry editor and browse to:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Certified Security Solutions\SCEP Server\Configuration
   ```

2. Edit the **CAConfiguration** value and add each additional CA to which enrollments should be directed in the format *hostname\logical name*.

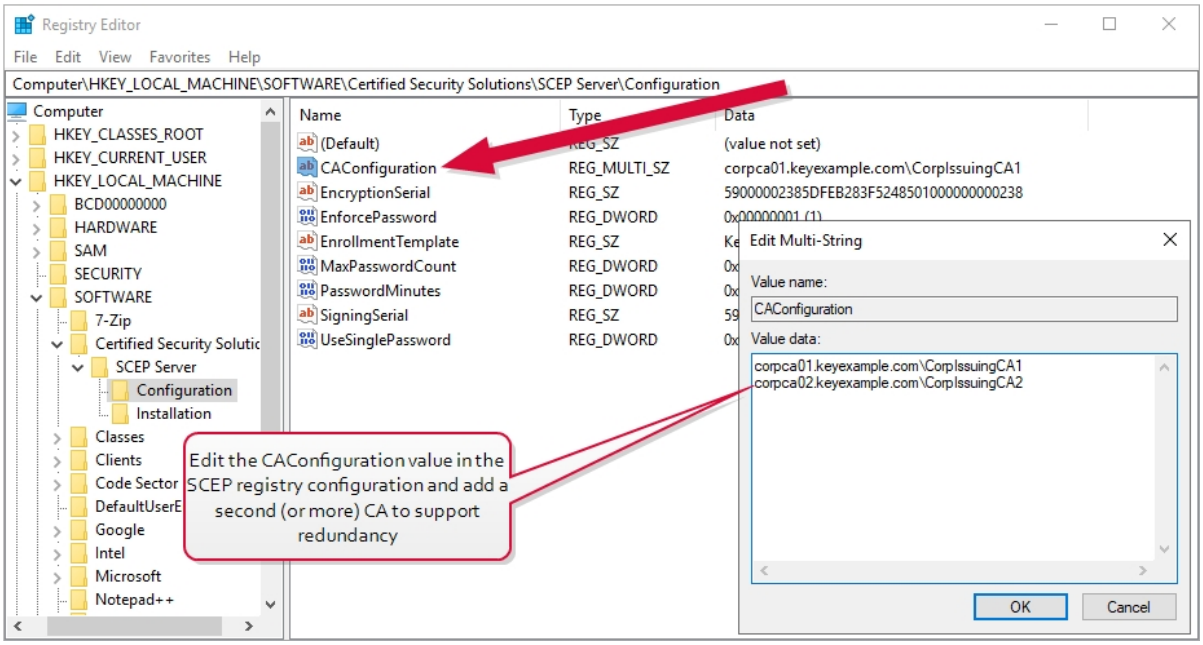

Figure 9: Edit the Registry to Add CAs for Redundancy

3. Make the template referenced in the **EnrollmentTemplate** value available for enrollment on each of the CAs.

4. Grant the service account under which the SCEP application pool runs (see Create Service Accounts for the SCEP Server on page 5) enrollment permissions on the template and on each of the CAs.

4.5 Optional Configuration Settings for Intune

The web.config file for the SCEP server contains two optional configuration settings that may be useful in some environments. Both settings relate to using Microsoft Intune for validation rather than traditional challenge-based SCEP validation.

Keyfactor.SCEP.Validator

Use this setting to configure the SCEP server to use Microsoft Intune for validation rather than a challenge-based SCEP validation. Supported values are:

- ChallengePassword
- Intune

The default is *ChallengePassword* (traditional SCEP challenge).

Keyfactor.SCEP.Intune.ConfigSource

Use this setting to configure where the Microsoft Intune connection information is stored. Supported values are:

- WebConfig
- SQL

If the default value of *WebConfig* is selected, the connection information is stored in the <intune> section of the web.config file. Consider using aspnet_regiis to encrypt the <intune> section after setting the values.

To configure optional Intune settings:

1. On the SCEP server, open a text editor (e.g. Notepad) using the ”Run as administrator” option.
2. In the text editor, browse to open the Web.config file for the SCEP server. By default, this file is located in the following directory:

   C:\Program Files\Keyfactor\Keyfactor SCEP Server\SCEP Server

3. In the web.config file in the appSettings section just below the connectionStrings section, find the *Keyfactor.SCEP.Validator* key. Change the Value for this key from *ChallengePassword* to *Intune* to use Intune as your validator. In the intune section just below the appSettings section, configure appropriate information to connect to your Intune account if you've opted to store the connection information in the web.config file (leaving the value of *Keyfactor.SCEP.Intune.ConfigSource* at the default of *WebConfig*). For best security practice, consider encrypting the intune section of the web.config file following configuration. For more information, see:

   https://docs.microsoft.com/en-us/previous-versions/aspnet/zhhddkxy(v=vs.100)

```
<appSettings>
  <add key="webpages:Version" value="3.0.0.0"/>
  <add key="webpages:Enabled" value="false"/>
  <add key="ClientValidationEnabled" value="true"/>
  <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
  <add key="NLogConfigFile" value="C:\Program Files\Common Files\Keyfactor\Keyfactor SCEP Server\NLog.config"/>

    <!-- Valid values for the below app setting are ChallengePassword and Intune -->
    <add key="Keyfactor.SCEP.Validator" value="Intune"/>
    <!-- (ChallengePassword only) To use SQL as the challenge repository, change the value of the below app setting to SQL
      You will also need to set the ChallengeDB connection string at the top of this file -->
    <add key="Keyfactor.SCEP.ChallengePassword.RepositoryType" value="ESENT"/>
    <!-- (Intune only) To use SQL as the Intune config provider, change the value of the below app setting to SQL
      You will also need to set the IntuneProvisioning connection string at the top of this file -->
    <add key="Keyfactor.SCEP.Intune.ConfigSource" value="WebConfig"/>
</appSettings>

  <!-- If storing Intune auth keys here, consider encrypting the section after setup
   See https://docs.microsoft.com/en-us/previous-versions/aspnet/zhhddkxy(v=vs.100) for more information -->
<intune ApplicationId="" ApplicationKey="" TenantId=""/>
```

Figure 10: Use Microsoft Intune as Validator

4. If desired, the connection information for Intune may be stored in a SQL database rather than the web.config file. To set this configuration, in the web.config file in the appSettings section, find the *Keyfactor.SCEP.Intune.ConfigSource* key. Change the Value for this key from *WebConfig* to *SQL*. Do not configure the intune section below the appSettings section. In the connectionStrings section above the appSettings section, configure a connection string appropriate for Intune provisioning in your environment.

```
<connectionStrings>
  <add name="IntuneProvisioning" connectionString=""/>
  <add name="ChallengeDB" connectionString="Server=sql.keyexample.com;Database=SCEP;Trusted_Connection=True"/>
</connectionStrings>
<appSettings>
  <add key="webpages:Version" value="3.0.0.0"/>
  <add key="webpages:Enabled" value="false"/>
  <add key="ClientValidationEnabled" value="true"/>
  <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
  <add key="NLogConfigFile" value="C:\Program Files\Common Files\Keyfactor\Keyfactor SCEP Server\NLog.config"/>

    <!-- Valid values for the below app setting are ChallengePassword and Intune -->
    <add key="Keyfactor.SCEP.Validator" value="Intune"/>
    <!-- (ChallengePassword only) To use SQL as the challenge repository, change the value of the below app setting to SQL
      You will also need to set the ChallengeDB connection string at the top of this file -->
    <add key="Keyfactor.SCEP.ChallengePassword.RepositoryType" value="SQL"/>
    <!-- (Intune only) To use SQL as the Intune config provider, change the value of the below app setting to SQL
      You will also need to set the IntuneProvisioning connection string at the top of this file -->
    <add key="Keyfactor.SCEP.Intune.ConfigSource" value="SQL"/>
</appSettings>

  <!-- If storing Intune auth keys here, consider encrypting the section after setup
   See https://docs.microsoft.com/en-us/previous-versions/aspnet/zhhddkxy(v=vs.100) for more information -->
<intune ApplicationId="" ApplicationKey="" TenantId=""/>
```

Figure 11: Store Microsoft Intune Settings in SQL

5.0 Operations

Once your Keyfactor SCEP server is up and running, it does not need much regular maintenance to keep it running. You may wish to monitor the logs (see Configure Logging on page 17) to identify potential issues early and have an overall sense of the health of your SCEP server. You may wish to add CAs for enrollment redundancy (see Configure CA Redundancy (Optional) on page 18). Depending on the configuration of the templates you used to acquire the SCEP encryption and signing certificates, the certificates will be valid for typically either one or two years. When the certificates are approaching expiration, you will need to acquire new certificates.

If you acquired the certificates for SCEP encryption and signing automatically by using the Request Certificates button in the Keyfactor SCEP Configuration tool, the certificates will most likely have a lifetime of 2 years because the certificates will have been requested using the built-in Microsoft *CEP Encryption* and *Exchange Enrollment Agent (Offline request)* templates, both of which have a 2 year lifetime that cannot be altered through the standard template management interface. If you wish to continue to use these templates, you can renew the certificates through the SCEP configuration tool as per Renew Certificates Using the Built-in Templates below.

If you acquired the certificates for SCEP encryption and signing manually as per Enable the Required Templates for SCEP Infrastructure on page 5 to allow you to use stronger keys than the built-in Microsoft templates support, the certificates may have a different lifetime than the 2 year lifetime supported by the built-in templates. If you wish to continue to use your custom templates, you should renew the certificates using the method described in Renew Certificates Using Custom Templates on the next page.

5.1 Renew Certificates Using the Built-in Templates

To renew the certificates using the built-in templates and the automatic request process:

1. On the SCEP server, use the Registry Editor (regedit) to open the following configuration area:

   HKEY_LOCAL_MACHINE\SOFTWARE\Certified Security Solutions\SCEP Server-\Configuration

2. In the Configuration key, double-click to edit the **EncryptionSerial** configuration setting and copy the value to a saved location in case you need to revert back to the old value, making note that this is the serial number for the old Keyfactor SCEP Server Encryption certificate. Clear out the value so the field is blank and click **OK** to save.

3. In the Configuration key, double-click to edit the **SigningSerial** configuration setting and copy the value to a saved location in case you need to revert back to the old value, making note that this is the serial number for the old Keyfactor SCEP Server Signing certificate. Clear out the value so the field is blank and click **OK** to save.

4. Close the Registry Editor.

5. Open the Keyfactor SCEP Configuration tool, which can be found on the Windows menus under Certified Security Solutions.

6. In the Keyfactor SCEP Configuration tool in the SCEP Infrastructure Certificates section of the page, click the **Request Certificates** button to request certificates for the SCEP server. The SCEP server requests certificates using the *CEP Encryption* and *Exchange Enrollment Agent (Offline request)* templates and will scan through any available CA in the environment for a CA that is able to issue certificates based on these templates, beginning with the CA configured in the SCEP Enrollment section of the page. If no CA is available to issue certificates based on these templates, an error will occur. You will need to make the templates available for issuing on a CA in the environment and try the Request Certificates step again.

   Note: If you have more than one CA with the SCEP infrastructure certificates templates available on it, wish to specify which of these CAs to request the certificates from, and have not selected this CA in the SCEP Enrollment section of the page, you can temporarily select the CA for the SCEP infrastructure CAs in the SCEP Enrollment section of the page, request the infrastructure certificates, and then change the CA in the SCEP Enrollment section of the page back to the CA you wish to use for SCEP enrollment.

7. At the bottom of the configuration tool, click **Save** and then close the dialog.

5.2 Renew Certificates Using Custom Templates

To renew the certificates using your custom templates:

1. On the Keyfactor SCEP server, use the Registry Editor (regedit) to open the following configuration area:

   HKEY_LOCAL_MACHINE\SOFTWARE\Certified Security Solutions\SCEP Server-\Configuration

2. Double-click to edit the **EncryptionSerial** configuration setting and copy the serial number for the Keyfactor SCEP Server Encryption certificate to a saved location in case you need to revert back to the old value, making note that this is the serial number for the old Keyfactor SCEP Server Encryption certificate. Click **Cancel** to close the dialog.

3. Double-click to edit the **SigningSerial** configuration setting and copy the serial number for the Keyfactor SCEP Server Signing certificate to a saved location in case you need to revert back to the old value, making note that this is the serial number for the old Keyfactor SCEP Server Signing certificate. Click **Cancel** to close the dialog.

4. On the Keyfactor SCEP server, open an empty instance of the Microsoft Management Console (MMC).

5. Choose **File->Add/Remove Snap-in...**.

6. In the Available snap-ins column, highlight **Certificates** and click **Add**.

7. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.

8. In the Certificates MMC, drill down to the Certificates folder under Personal, right-click the Keyfactor SCEP Server Encryption certificate (your certificate may have a different name), and choose **Open**. On the Details tab, locate the **Serial number** and confirm that it matches the serial number you copied in step two. On the Details tab, locate the **Certificate Template Information** and make a note of the template used to acquire the certificate.

9. In the Certificates folder under Personal, right-click the Keyfactor SCEP Server Signing certificate (your certificate may have a different name), and choose **Open**. On the Details tab, locate the **Serial number** and confirm that it matches the serial number you copied in step three. On the Details tab, locate the **Certificate Template Information** and make a note of the template used to acquire the certificate.

10. Under the Personal folder, right-click **Certificates** and choose **All Tasks->Request New Certificate...**.

11. In the Certificate Enrollment Wizard, click **Next**.

12. On the Select Certificate Enrollment Policy page, accept the default and click **Next**.

13. On the Request Certificates page, scroll down to locate the template you identified in step eight (or another template you have created for the SCEP encryption certificate as per Enable the Required Templates for SCEP Infrastructure on page 5), and check the box for the template. If the template does not appear in the list, you may need to verify that the SCEP server machine account has been granted enroll permissions on the template.

14. On the Request Certificates page, click the link below the **Keyfactor SCEP Encryption** template name that says "More information is required to enroll for this certificate...". On the Subject tab of the Certificate Properties dialog, select **Common name** in the **Type** dropdown under **Subject name**, enter a name for the certificate in the **Value** field, and click the **Add** button. No specific text is required in the subject name. This name is for your reference and to clarify the purpose of the certificate—e.g. Keyfactor SCEP Server Encryption. Click **OK** at the bottom of the Certificate Properties dialog.

15. On the Request Certificates page, click **Enroll** to enroll for the certificate and click Finish when the enrollment is complete.

16. Repeat steps 10 through 15 using the **Keyfactor SCEP Signing** template identified in step nine (or another template you have created for the SCEP signing certificate as per section Enable the Required Templates for SCEP Infrastructure on page 5) to acquire a second certificate.

17. In the Certificates MMC in the Certificates folder under Personal, right-click the Keyfactor SCEP Server Encryption certificate, and choose **Open**. On the Details tab, locate the **Serial number** and copy the serial number from the box at the bottom of the dialog to a text file, making note that this is the encryption certificate serial number. Remove the spaces from the serial number so that the serial number string looks something like this:

69000016e1ffccf7521125122a0000000016e1

Important: As displayed in the certificates dialog, the serial number has a narrow leading space that is actually an unprintable control character. If you accidentally copy this character and paste it into the registry setting when you are following the instructions in steps 21 and 22, the serial numbers will fail to appear in the Keyfactor SCEP Configuration tool. Be sure to strip off any leading spaces on the copied text.

18. Repeat step 17 for the Keyfactor SCEP Server Signing certificate.

19. Return to the Registry Editor (regedit) and the following configuration area:

    HKEY_LOCAL_MACHINE\SOFTWARE\Certified Security Solutions\SCEP Server-\Configuration

20. Double-click to edit the **EncryptionSerial** configuration setting and paste in the serial number for the Keyfactor SCEP Server Encryption certificate that you made note of in step 17, replacing the existing value. Click **OK** to save.

21. Double-click to edit the **SigningSerial** configuration setting and paste in the serial number for the Keyfactor SCEP Server Signing certificate that you made note of in step 18, replacing the existing value. Click **OK** to save.

22. Open the Keyfactor SCEP Configuration tool, which can be found on the Windows menus under Certified Security Solutions.

23. In the Keyfactor SCEP Configuration tool in the SCEP Infrastructure Certificates section of the page, confirm that the serial numbers listed are the new serial numbers you made note of in steps 17 and 18.

24. In the Keyfactor SCEP Configuration tool in the SCEP Service Account section of the page, check the **Change Account** box and re-enter the password for the SCEP service account. Click the verify button () to confirm that the password entered is valid.

    Note: If the password for the SCEP service account is not immediately available, you can skip this step and instead manually grant the SCEP service account permissions to manage the private keys of the certificates as follows:

    a. In the Certificates MMC in the Certificates folder under Personal, right-click the Keyfactor SCEP Server Encryption certificate and choose **All Tasks->Manage Private Keys...**.

    b. In the Permissions for private keys dialog, click **Add**, add the SCEP service account (configured in the Keyfactor SCEP Configuration tool), and grant that service account **Read** but not **Full control** permissions. Click **OK** to save.

    c. Repeat these steps for the Keyfactor SCEP Server Signing certificate.

25. At the bottom of the configuration tool, click **Save** and then close the dialog.

5.0 Release Notes

SCEP v1.7.0

- Update: The Intune integration now uses the Microsoft Authentication Library (MSAL) rather than the Active Directory Authentication Library (ADAL). Customers using the Intune integration should upgrade before Microsoft disables support for ADAL—scheduled for June 2023 as of this writing.

SCEP v1.6.0

- Change: Supported OSs for install are now Windows Server 2016 and Windows Server 2019.

SCEP v1.5.2

- Update: Improvements to logging when used in conjunction with Intune.

SCEP v1.5.0

- Update: The product now includes the option to use a SQL database rather than a local database to store challenge passwords. This is useful in environments wishing to deploy multiple SCEP servers with load balancing. When a SQL database is used, a SCEP challenge can be requested from one SCEP server and subsequently submitted to another.
- Update: SCEP requests can now be directed to multiple CAs so that if the first CA cannot be reached, the request will automatically be submitted to a second CA, and so on. As long as at least one CA responds and is a success, the request will be a success.

A

AIA

The authority information access (AIA) is included in a certificate--if configured--and identifies a location from which the chain certificates for that certificate may be retrieved.

AnyAgent

The AnyAgent, one of Keyfactor's suite of orchestrators, is used to allow management of certificates regardless of source or location by allowing customers to implement custom agent functionality via an API.

AnyGateway

The Keyfactor AnyGateway is a generic third party CA gateway framework that allows existing CA gateways and custom CA connections to share the same overall product framework.

API

A set of functions to allow creation of applications. Keyfactor offers the Keyfactor API, which allows third-party software to integrate with the advanced certificate enrollment and management features of Keyfactor Command.

Argument

A parameter or argument is a value that is passed into a function in an application.

Authority Information Access

The authority information access (AIA) is included in a certificate--if configured--and identifies a location from which the chain certificates for that certificate may be retrieved.

B

Bash Orchestrator

The Bash Orchestrator, one of Keyfactor's suite of orchestrators, is used to discover and manage SSH keys across an enterprise.

Blueprint

A snapshot of the certificate stores and scheduled jobs on one orchestrator, which can be used to create matching certificate stores and jobs on another orchestrator with just a few clicks.

C

CA

A certificate authority (CA) is an entity that issues digital certificates. Within Keyfactor Command, a CA may be a Microsoft CA or a Keyfactor gateway to a cloud-based or remote CA.

Certificate Authority

A certificate authority (CA) is an entity that issues digital certificates. Within Keyfactor Command, a CA may be a Microsoft CA or a Keyfactor gateway to a cloud-based or remote CA.

Certificate Revocation List

A Certificate Revocation List (CRL) is a list of digital certificates that have been revoked by the issuing Certificate Authority (CA) before their scheduled expiration date and should no longer be trusted.

Certificate Signing Request

A CSR or certificate signing request is a block of encoded text that is submitted to a CA when enrolling for a certificate. When

you generate a CSR within Keyfactor Command, the matching private key for it is stored in Keyfactor Command in encrypted format and will be married with the certificate once returned from the CA.

CN

A common name (CN) is the component of a distinguished name (DN) that represents the primary name of the object. The value varies depending on the type of object. For a user object, this would be the user's name (e.g. CN=John Smith). For SSL certificates, the CN is typically the fully qualified domain name (FQDN) of the host where the SSL certificate will reside (e.g. servername.keyexample.com or www.keyexample.com).

Collection

The certificate search function allows you to query the Keyfactor Command database for certificates from any available source based on any criteria of the certificates and save the results as a collection that will be availble in other places in the Management Portal (e.g. expiration alerts and certain reports).

Common Name

A common name (CN) is the component of a distinguished name (DN) that represents the primary name of the object. The value varies depending on the type of object. For a user object, this would be the user's name (e.g. CN=John Smith). For SSL certificates, the CN is typically the fully qualified domain name (FQDN) of the host where the SSL certificate will reside (e.g. servername.keyexample.com or www.keyexample.com).

Configuration Tenant

A grouping of CAs. The Microsoft concept of forests is not used in EJBCA so to accommodate the new EJBCA functionality, and to avoid confusion, the term forest needed to be renamed. The new name is configuration tenant. For EJBCA, there would be one configuration tenant per EJBCA server install. For Microsoft, there would be one per forest. Note that configuration tenants cannot be mixed, so Microsoft and EJBCA cannot exist on the same configuration tenant.

CRL

A Certificate Revocation List (CRL) is a list of digital certificates that have been revoked by the issuing Certificate Authority (CA) before their scheduled expiration date and should no longer be trusted.

CSR

A CSR or certificate signing request is a block of encoded text that is submitted to a CA when enrolling for a certificate. When you generate a CSR within Keyfactor Command, the matching private key for it is stored in Keyfactor Command in encrypted format and will be married with the certificate once returned from the CA.

D

DER

A DER format certificate file is a DER-encoded binary certificate. It contains a single certificate and does not support storage of private keys. It sometimes has an extension of .der but is often seen with .cer or .crt.

Distinquished Name

A distinguished name (DN) is the name that uniquely identifies an object in a directory. In the context of Keyfactor Command, this directory is generally Active Directory. A DN is made up of attribute=value pairs, separated by commas. Any of the attributes defined in the directory schema can be used to make up a DN.

DN

A distinguished name (DN) is the name that uniquely identifies an object in a directory. In the context of Keyfactor Command, this directory is generally Active Directory. A DN is made up of attribute=value pairs, separated by commas. Any of the attributes defined in the directory schema can be used to make up a DN.

DNS

The Domain Name System is a service that translates names into IP addresses.

E

ECC

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers.

Endpoint

An endpoint is a URL that enables the API to gain access to resources on a server.

Enrollment

Certificate enrollment refers to the process by which a user requests a digital certificate. The user must submit the request to a certificate authority (CA).

EOBO

A user with an enrollment agent certificate can enroll for a certificate on behalf of another user. This is often used when provisioning technology such as smart cards.

F

Forest

An Active Directory forest (AD forest) is the top most logical container in an Active Directory configuration that contains domains, and objects such as users and computers.

G

Gateway Connector

The Keyfactor Gateway Connector is installed in the customer forest to provide a connection between the on-premise CA and the Azure-hosted, Keyfactor managed Hosted Configuration Portal to provide support for synchronization, enrollment and management of certificates through the Azure-hosted instance of Keyfactor Command for the on-premise CA. It is supported on both Windows and Linux.

H

Host Name

The unique identifier that serves as name of a computer. It is sometimes presented as a fully qualified domain name (e.g. server-

name.keyexample.com) and sometimes just as a short name (e.g. servername).

Hosted Config Portal

The Keyfactor Hosted Configuration Portal is used to configure connections between on-premise instances of the Keyfactor Gateway Connector and and on-premise CAs to make them available to Azure-hosted instance of Keyfactor Command.The portal is Azure-hosted and managed by Keyfactor.

Hosted Configuration Portal

The Keyfactor Hosted Configuration Portal is used to configure connections between on-premise instances of the Keyfactor Gateway Connector and and on-premise CAs to make them available to Azure-hosted instance of Keyfactor Command.The portal is Azure-hosted and managed by Keyfactor.

Hostname

The unique identifier that serves as name of a computer. It is sometimes presented as a fully qualified domain name (e.g. servername.keyexample.com) and sometimes just as a short name (e.g. servername).

J

Java Agent

The Java Agent, one of Keyfactor's suite of orchestrators, is used to perform discovery of Java keystores and PEM certificate stores, to inventory discovered stores, and to push certificates out to stores as needed.

Java Keystore

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based applications for authentication and encryption.

JKS

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based applications for authentication and encryption.

K

Key Length

The key size or key length is the number of bits in a key used by a cryptographic algorithm.

Key Pair

In asymmetric cryptography, public keys are used together in a key pair with a private key. The private key is retained by the key's creator while the public key is widely distributed to any user or target needing to interact with the holder of the private key.

Key Size

The key size or key length is the number of bits in a key used by a cryptographic algorithm.

Key Type

The key type identifies the type of key to create when creating a symmetric or asymmetric key. It references the signing algorithm and often key size (e.g. AES-256, RSA-2048, Ed25519).

Keyfactor CA Management Gateway

The Keyfactor CA Management Gateway is made up of the Keyfactor Gateway Connector, installed in the customer forest to provide a connection to the local CA, and

the Azure-hosted and Keyfactor managed Hosted Configuration Portal. The solution is used to provide a connection between a customer's on-premise CA and an Azure-hosted instance of Keyfactor Command for synchronization, enrollment, and management of certificates.

Keyfactor Gateway Connector

The Keyfactor Gateway Connector is installed in the customer forest to provide a connection between the on-premise CA and the Azure-hosted, Keyfactor managed Hosted Configuration Portal to provide support for synchronization, enrollment and management of certificates through the Azure-hosted instance of Keyfactor Command for the on-premise CA. It is supported on both Windows and Linux.

Keyfactor Universal Orchestrator

The Keyfactor Universal Orchestrator, one of Keyfactor's suite of orchestrators, is used to interact with servers and devices for certificate management, run SSL discovery and management tasks, and manage synchronization of certificate authorities in remote forests. With the addition of custom extensions, it can provide certificate management capabilities on a variety of platforms and devices (e.g. Amazon Web Services (AWS) resources, Citrix\NetScaler devices, F5 devices, IIS stores, JKS keystores, PEM stores, and PKCS#12 stores) and execute tasks outside the standard list of certificate management functions. It runs on either Windows or Linux servers or Linux containers.

Keystore

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based applications for authentication and encryption.

L

Logical Name

The logical name of a CA is the common name given to the CA at the time it is created. For Microsoft CAs, this name can be seen at the top of the Certificate Authority MMC snap-in. It is part of the FQDN\Logical Name string that is used to refer to CAs when using command-line tools and in some Keyfactor Command configuration settings (e.g. ca2.keyexample.-com\Corp Issuing CA Two).

M

MAC Agent

The MAC Agent, one of Keyfactor's suite of orchestrators, is used to manage certificates on any keychains on the Mac on which the Keyfactor MAC Agent is installed.

Metadata

Metadata provides information about a piece of data. It is used to summarize basic information about data, which can make working with the data easier. In the context of Keyfactor Command, the certificate metadata feature allows you to create custom metadata fields that allow you to tag certificates with tracking information about certificates.

O

Object Identifier

Object identifiers or OIDs are a standardized system for identifying any object, concept, or "thing" with a globally unambiguous persistent name.

OID

Object identifiers or OIDs are a standardized system for identifying any object, concept, or ”thing” with a globally unambiguous persistent name.

Orchestrator

Keyfactor orchestrators perform a variety of functions, including managing certificate stores and SSH key stores.

P

P12

A PFX file (personal information exchange format), also known as a PKCS#12 archive, is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

P7B

A PKCS #7 format certificate file is a base64-encoded certificate. Since it’s presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. Unlike PEM files, PKCS #7 files can contain only a certificate and its certifiate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

P7C

A PKCS #7 format certificate file is a base64-encoded certificate. Since it’s presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. Unlike PEM files, PKCS #7 files can contain only a certificate and its certifiate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

Parameter

A parameter or argument is a value that is passed into a function in an application.

PEM

A PEM format certificate file is a base64-encoded certificate. Since it’s presented in ASCII, you can open it in any text editor. PEM certificates always begin and end with entries like ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. PEM certificates can contain a single certificate or a full certifiate chain and may contain a private key. Usually, extensions of .cer and .crt are certificate files with no private key, .key is a separate private key file, and .pem is both a certificate and private key.

PFX

A PFX file (personal information exchange format), also known as a PKCS#12 archive, is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

PKCS #7

A PKCS #7 format certificate file is a base64-encoded certificate. Since it’s presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. Unlike PEM files, PKCS #7 files can contain only a certificate and its certifiate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

PKCS#12

A PFX file (personal information exchange format), also known as a PKCS#12 archive, is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

PKI

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

Private Key

Private keys are used in cryptography (symmetric and asymmetric) to encrypt or sign content. In asymmetric cryptography, they are used together in a key pair with a public key. The private or secret key is retained by the key's creator, making it highly secure.

Public Key

In asymmetric cryptography, public keys are used together in a key pair with a private key. The private key is retained by the key's creator while the public key is widely distributed to any user or target needing to interact with the holder of the private key.

Public Key Infrastructure

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

R

Rogue Key

A rogue key, in the context of Keyfactor Command, is an SSH public key that appears in an authorized_keys file on a server managed by the SSH orchestrator without authorization.

Root of Trust

A root of trust (RoT) is a source within a cryptographic system that can always be trusted. It is typically a hardened hardware module. HSMs (hardware security modules) and TPMs (trusted platform modules) are examples of RoTs.

RoT

A root of trust (RoT) is a source within a cryptographic system that can always be trusted. It is typically a hardened hardware module. HSMs (hardware security modules) and TPMs (trusted platform modules) are examples of RoTs.

RPC

Remote procedure call (RPC) allows one program to call a function from a program located on another computer on a network without specifying network details. In the context of Keyfactor Command, RPC errors often indicate Kerberos authentication or delegation issues.

rsyslog

Rsyslog is an open-source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network.

S

SAN

The subject alternative name (SAN) is an extension to the X.509 specification that allows you to specify additional values when enrolling for a digital certificate. A variety of SAN formats are supported, with DNS name being the most common.

server name indication

Server name indication (SNI) is an extension to TLS that provides for including the host-name of the target server in the initial hand-shake request to allow the server to respond with the correct SSL certificate or allow a proxy to forward the request to the appropriate target.

SMTP

Short for simple mail transfer protocol, SMTP is a protocol for sending email messages between servers.

SNI

Server name indication (SNI) is an extension to TLS that provides for including the host-name of the target server in the initial hand-shake request to allow the server to respond with the correct SSL certificate or allow a proxy to forward the request to the appropriate target.

SSH

The SSH (secure shell) protocol provides for secure connections between computers. It provides several options for authen-tication, including public key, and protects the communications with strong encryption.

SSL

TLS (Transport Layer Security) and its predecessor SSL (Secure Sockets Layer) are protocols for establishing authenticated and encrypted links between networked computers.

Subject Alternative Name

The subject alternative name (SAN) is an extension to the X.509 specification that allows you to specify additional values when enrolling for a digital certificate. A variety of SAN formats are supported, with DNS name being the most common.

T

Template

A certificate template defines the policies and rules that a CA uses when a request for a certificate is received.

TLS

TLS (Transport Layer Security) and its predecessor SSL (Secure Sockets Layer) are protocols for establishing authenticated and encrypted links between networked computers.

Trusted CA

A certificate authority in the forest in which Keyfactor Command is installed or in a forest in a two-way trust with the forest in which Keyfactor Command is installed.

U

Untrusted CA

A certificate authority in a forest in a one-way trust with the forest in which Keyfactor Command is installed or in a forest that is

untrusted by the forest in which Keyfactor Command is installed. Non-domain-joined standalone CAs also fall into this category.

W

Web API

A set of functions to allow creation of applications. Keyfactor offers the Keyfactor API, which allows third-party software to integrate with the advanced certificate enrollment and management features of Keyfactor Command.

Windows Orchestrator

The Windows Orchestrator, one of Keyfactor's suite of orchestrators, is used to manage synchronization of certificate authorities in remote forests, run SSL discovery and management tasks, and interact with Windows servers as well as F5 devices, NetScaler devices, Amazon Web Services (AWS) resources, and FTP capable devices, for certificate management. In addition, the AnyAgent capability of the Windows Orchestrator allows it to be extended to create custom certificate store types and management capabilities regardless of source platform or location.

Workflow

A workflow is a series of steps necessary to complete a process. In the context of Keyfactor Command, it refers to the workflow builder, which allows you automate event-driven tasks when a certificate is requested or revoked.

X

x.509

In cryptography, X.509 is a standard defining the format of public key certificates. An X.509 certificate contains a public key and an identity (e.g. a host name or an organization or individual name), and is either signed by a certificate authority or self-signed. When a certificate is signed by a trusted certificate authority it can be used to establish trusted secure communications with the owner of the corresponding private key. It can also be used to verify digitally signed documents and emails.

7.0 Copyright Notice

User guides and related documentation from Keyfactor are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. This documentation may not be disclosed, transferred, modified, or reproduced in any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of Keyfactor and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by Keyfactor. Keyfactor assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from Keyfactor.