KEYFACTOR

2024 | Keyfactor

Remote CA Gateway 24.1

Installation & Reference Guide

Table of Contents

List of Figures

List of Tables

1.0 Introduction

The Keyfactor Remote CA Gateway solution by Keyfactor allows organizations to leverage existing on-premise CAs with an Azure-hosted, Keyfactor-managed instance of Keyfactor Command to issue and manage certificates across enterprise infrastructures. Out-of-the-box, Microsoft and EJBCA CAs are supported. Other CAs can be supported with the addition of a custom connector extension.

The Keyfactor Remote CA Gateway is made up of:

- The Keyfactor Remote CA Gateway Connector and a connector extension to allow communication with a specific type of CA (e.g. Microsoft), which are installed in the on-premise forest to provide a connection to an on-premise CA.

  Note: A single gateway connector can connect to more than one on-premise CA as long as all CAs are of the same type (e.g. Microsoft).

- The Keyfactor Remote CA Service, which is Azure-hosted and managed by Keyfactor. The service receives the connection from the connector and brokers it to Keyfactor Command.
- The Keyfactor Remote CA Configuration Portal, which is Azure-hosted and managed by Keyfactor. The portal is used to configure the gateway connectors and CAs that will be made available to Keyfactor Command.

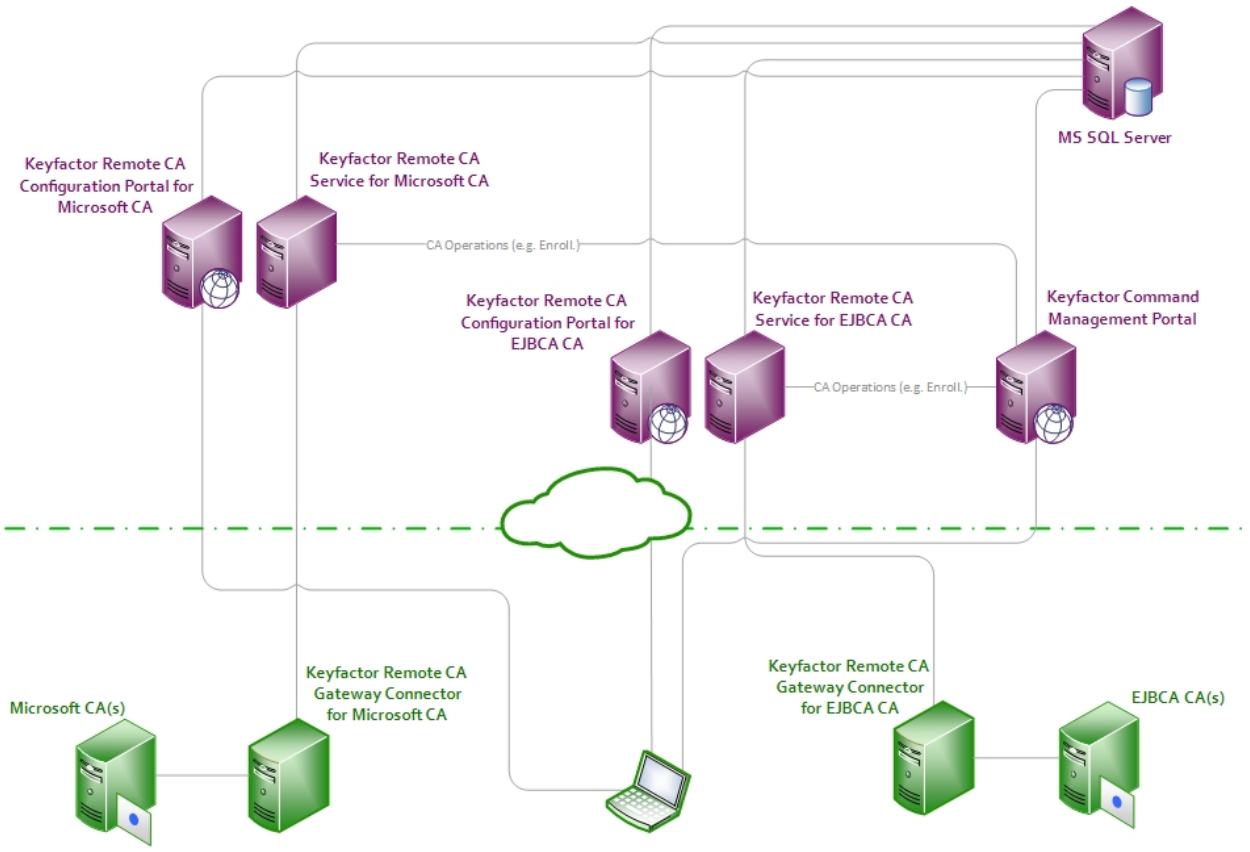

Figure 1: Keyfactor Remote CA Gateway Architecture

The Keyfactor Remote CA Gateway Connector runs on either Windows or Linux and can be installed either on the CA or on a separate machine on the same network. Connections to Microsoft CAs are only supported from gateway connectors running on Windows.

A given instance of the Keyfactor Remote CA Gateway Connector and associated Keyfactor Remote CA Service and Keyfactor Remote CA Configuration Portal can support only one type of CA. If you have more than one type of CA (e.g. both Microsoft and EJBCA), you will need more than one instance of these.
For a comprehensive description of the components that make up Keyfactor Command, please see the Installation and Reference Guides[1] for both the server and the orchestrators and gateways that enhance the server functionality.

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

2.0 Preparing

This section describes the steps that need to be taken prior to a Keyfactor Remote CA Gateway Connector installation to install the prerequisites, create the required supporting components, and gather the necessary information to complete the Keyfactor Remote CA Gateway Connector installation and configuration process.

2.1 System Requirements

The Keyfactor Remote CA Gateway Connector is supported on the following operating systems:

- Windows Server 2019 or higher
- Oracle Linux 7 or higher
- Red Hat Enterprise 7 or higher
- Ubuntu 16 or higher

Important: Connections to Microsoft CAs are only supported from instance of the Keyfactor Remote CA Gateway Connector running on Windows.

Note: The gateway connector may be installed *on* the CA as long as the system requirements are met.

The gateway has the following application requirements:

Windows Application Requirements

The following applications are required in order to install the Keyfactor Remote CA Gateway Connector on Windows servers.

Microsoft ASP.NET 6 Runtime

The gateway connector requires the ASP.NET version 6.

You can use the following PowerShell command to check the .NET core runtime version(s) installed on a server (if any):

```
dotnet --list-runtimes
```

Microsoft Visual C++ Redistributable

If you plan to connect to a Microsoft CA, the gateway connector requires the Microsoft Visual C++ 2019 (or later) redistributable for x64. This is available for download from Microsoft:

https://aka.ms/vs/16/release/vc_redist.x64.exe

The Microsoft Visual C++ Redistributable appears as an application in the Windows Apps & features.

Note: If this redistributable is missing, you may encounter an error similar to the following:

```
Could not load file or assembly 'Keyfactor.CAClient.Microsoft.DCOM,
Version=2.1.1.0, Culture=neutral, PublicKeyToken=0ed89d330114ab09'. An attempt
was made to load a program with an incorrect format.
```

Linux Application Requirements

The following applications are required in order to install the Keyfactor Remote CA Gateway Connector on Linux servers.

Microsoft ASP.NET 6 Runtime

The gateway connector requires ASP.NET version 6. In many cases, this can be installed via the OS package manager. The method to complete this varies depending on the Linux operating system.

You can use the following command to check the .NET core runtime version(s) installed on a server (if any):

```
dotnet --list-runtimes
```

jq

The gateway connector can only be installed on a Linux server that has jq installed. You can use the following command to check the jq version of a server:

```
jq --version
```

systemd

The gateway connector requires a Linux server that uses the systemd service manager. You can use the following command to test whether a system is running systemd:

```
ps -p 1
```

bash

The gateway connector can only be installed on a Linux server that is running bash version 4.3 or higher. You can use the following command to check the bash version of a server:

```
bash --version
```

A given instance of the Keyfactor Remote CA Gateway Connector and associated Keyfactor Remote CA Service and Keyfactor Remote CA Configuration Portal can support only one type of CA. If you have more than one type of CA (e.g. both Microsoft and EJBCA), you will need more than one instance of these.

2.2 Identify the Keyfactor Remote CA Gateway Connector Access Token

The Keyfactor Remote CA Gateway Connector uses an OAuth 2.0 access token to make a connection to the Keyfactor Remote CA Gateway. This access token consists of four components:

- Scope: The scope is the mechanism by which the gateway connector makes a request for the specific access that it needs.
- Authentication URL: The authentication URL is a URL to the service providing OAuth2 authentication.
- Client ID: The client ID is a string issued by the authentication provider that identifies the application authenticating. It is generally something long and not guessable.
- Client Secret: The client secret is a secret shared between the components of the Keyfactor Remote CA Gateway implementation and the authentication provider. This secret should be handled securely.

The person performing the Keyfactor Remote CA Gateway Connector installation will need these pieces of information, which should be provided by your Keyfactor representative.

2.3 Create Service Accounts for the Keyfactor Remote CA Gateway

The Keyfactor Remote CA Gateway makes use of up to two service accounts to allow it to communicate with the on-premise CA(s) and the Keyfactor Command server. The Keyfactor Gateway Connector Service on the Keyfactor Remote CA Gateway Connector server runs as one service account. A second service account may be configured in the Keyfactor Remote CA Configuration Portal to allow the Keyfactor Remote CA Gateway to make a connection to the CA to read certificate records, enroll for new certificates, and perform management functions such as revocation. Under some circumstances, the same service account may be used for both roles.

- **Keyfactor Remote CA Gateway Connector Service**
  - **Windows**

    When the Keyfactor Remote CA Gateway Connector is installed on Windows, you may use either the built-in Network Service account or a custom service account to run the service. The custom service account may be either an Active Directory service account or a local machine account. Of the custom service account choices, an Active Directory account is more typically used unless the machine is not domain-joined. If you use an Active Directory

service account, it needs to be a service account in the forest in which the Keyfactor Remote CA Gateway Connector is installed.

The Keyfactor Gateway Connector Service on the server on which the Keyfactor Remote CA Gateway Connector is installed runs as the service account you select for this role. The service account requires local "Log on as a service" permissions.

- **Linux**

  For the purposes of this documentation, it is assumed that Linux machines will be non-domain joined and will use a local account to run the Keyfactor Remote CA Gateway Connector.

  For Linux systems, Keyfactor recommends running the service as an account other than root. The default account of *keyfactor-gatewayconnector* will be created automatically during the install if the force option is used. If you prefer not to use the force option, you may create a local service account before running the installation script.

- **Keyfactor Remote CA Configuration Portal Application Pool**

  The IIS application pool for the Keyfactor Remote CA Configuration Portal runs in the context of an Active Directory user in the Keyfactor-managed forest. This application pool user needs to be granted permissions to manage the Keyfactor Remote CA Configuration Portal (see Security Tab on page 30); all management tasks in the portal are done in the context of this user. This service account information will be provided to you by your Keyfactor representative.

- **Keyfactor Remote CA Configuration Portal CA Connection Account**

  - **Microsoft CAs**

    When each Microsoft CA is configured in the Keyfactor Remote CA Configuration Portal, a service account from the on-premise forest must be configured to allow a connection to be made from the Keyfactor Remote CA Gateway (and thus Keyfactor Command) to the on-premise CA via the Keyfactor Remote CA Gateway Connector (see CA Connection Tab on page 37). If the Keyfactor Remote CA Gateway Connector has been installed to run using an Active Directory service account, this same account may be used for the CA connection role. If the Keyfactor Remote CA Gateway Connector is running as a local account, an Active Directory service account needs to be created for this role.

  - **EJBCA CAs**

    When each EJBCA CA is configured in the Keyfactor Remote CA Configuration Portal, a client certificate is selected to authenticate the Keyfactor Remote CA Gateway Connector to the Keyfactor Gateway Connector Service and Keyfactor Command (see CA Connection Tab on page 37). This certificate needs to be issued from the EJBCA CA and associated with an EJBCA end entity. An end entity may be created specifically for this role, or an existing end entity may be used. For Windows, the certificate needs to either be installed in the local computer personal store of the gateway connector server or hosted in a file path on the gateway connector server. For Linux, the certificate needs to be hosted in a file path on the gateway connector server.

The service accounts need to be created prior to installation of the Keyfactor Remote CA Gateway Connector software (except as noted above for installations on Linux), and the person installing the Keyfactor Remote CA Gateway Connector software and configuring the CA(s) in the Keyfactor

Remote CA Configuration Portal needs to know the domain (if applicable), username and password of each service account.

2.4 Configure a Certificate Root Trust for the Keyfactor Remote CA Gateway

The Keyfactor Remote CA Gateway requires the use of HTTPS to secure the channel between each Keyfactor Remote CA Gateway Connector and the Keyfactor Remote CA Service server(s). This requires an SSL certificate configured in IIS on the Keyfactor Remote CA Service server(s). This certificate can either be a publicly-rooted certificate (e.g. from DigiCert, Entrust, etc.), or one issued from a private certificate authority (CA). If your Keyfactor Remote CA Service server is using a publicly rooted certificate, the Keyfactor Remote CA Gateway Connector server may already trust the certificate root for this certificate. However, if you have opted to use an internally-generated certificate, your Keyfactor Remote CA Gateway Connector server may not trust this certificate. In order to use HTTPS for communications between the Keyfactor Remote CA Gateway Connector and the Keyfactor Remote CA Service server with a certificate generated from a private CA, you may need to import the certificate chain for the certificate into either the local machine certificate store on the Keyfactor Remote CA Gateway Connector server on Windows or the root certificate store on Linux.

Installations on Windows

If the public key infrastructure (PKI) that issued the certificate has only a root CA, the root certificate from this CA must be installed in the Trusted Root Certification Authorities store under Local Computer on the Keyfactor Remote CA Gateway Connector server. If the PKI that issued the certificate has both a root and issuing CA, the root certificate must be installed in the Trusted Root Certification Authorities store under Local Computer on the Keyfactor Remote CA Gateway Connector server and the issuing CA certificate must be installed in the Intermediate Certification Authorities store under Local Computer on the Keyfactor Remote CA Gateway Connector server.

Installations on Linux

The location of the OpenSSL trusted root store varies depending on your Linux implementation. The root certificate must be installed in the appropriate location for the operating system before beginning the installation.

2.5 Grant the Gateway Service Account Permissions on the CA

In order to allow the Keyfactor Remote CA Gateway to make a connection to the CA to read certificate records, enroll for new certificates, and perform management functions such as revocation, the service account configured in the Keyfactor Remote CA Configuration Portal (see Keyfactor Remote CA Configuration Portal CA Connection Account on the previous page) must be granted appropriate permissions to the CA database.

Microsoft CAs

In the Microsoft management console (MMC) for each CA that a Keyfactor Remote CA Gateway Connector will interact with, open the properties for the CA and grant the Keyfactor Remote CA Configuration Portal CA connection service account (see Keyfactor Remote CA Configuration Portal CA Connection Account on page 6):

- **Read** permissions to allow it to synchronize certificates from the CA
- **Request Certificates** to allow it to request certificates from the CA
- **Issue and Manage Certificates** to allow it to perform workflow tasks and revocation

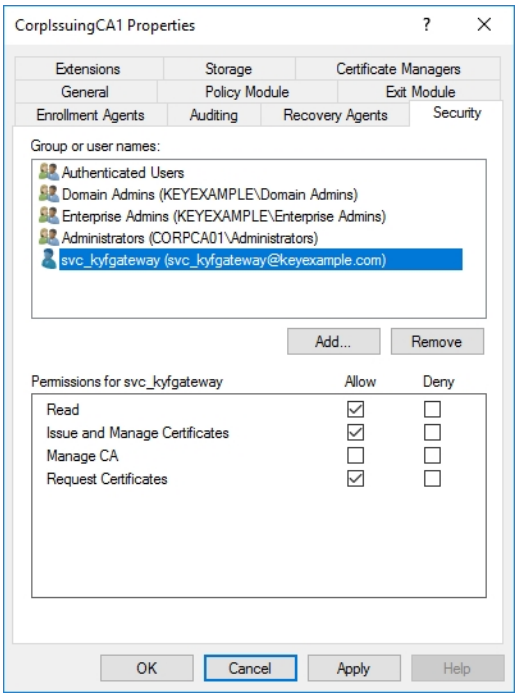

Figure 2: Grant CA Permissions for a Microsoft CA

EJBCA CAs

In the EJBCA administration portal for each CA that a Keyfactor Remote CA Gateway Connector will interact with, create or select a role that the end entity associated with the certificate that is used to make the connection from the gateway connector (see Keyfactor Remote CA Configuration Portal CA Connection Account on page 6) will hold and make certain that the access rules for that role include the permissions shown in Figure 3: Grant CA Permissions for an EJBCA CA. The end entity must hold all the roles shown for full functionality (where *Sample* is the name of your end entity profile). You may wish to grant the */ra_functionality/create_end_entity/* permission to allow the end entity to create new end entities, which is useful during configuration (see CA Connection Tab on page 37).

Figure 3: Grant CA Permissions for an EJBCA CA

2.6 Configure Firewall Settings

In order for the Keyfactor Remote CA Gateway Connector to be able to communicate with the Keyfactor Remote CA Service server and the local Active Directory, appropriate firewall ports need to be open on the Keyfactor Remote CA Gateway Connector server and throughout the environment. These ports may already be open or may need to be opened.

Table 1: Protocols the Keyfactor Remote CA Gateway Connector Uses for Communication

Type	Protocols and Ports	Source/Target
Inbound	RPC (TCP 135)	Keyfactor Remote CA Service, for enrollment
Inbound	DCOM (Random high ports typically in the range TCP 49152 – 65535)	Keyfactor Remote CA Service, for enrollment
Outbound	Active Directory Web Services (TCP 9389)	Active Directory domain controllers, for template retrieval
Outbound	HTTPS (TCP 443)	Keyfactor Remote CA Service

On the Keyfactor Remote CA Gateway Connector server:

1. Verify that the current ephemeral port range is open by opening an administrative command prompt and running the following command:

```
netsh interface ipv4 show dynamic protocol=tcp
```

The output from this command should look like this:

```
Protocol tcp Dynamic Port Range
---------------------------------
Start Port: 49152
Number of Ports: 16384
```

2. If the range is not open, it needs to be opened to allow RPC communication via TCP. Keyfactor provides a PowerShell script for this purpose for use on Windows servers (see Appendix A—Firewall Rules for Windows on page 53).
   No rules are included in this script for HTTP/HTTPS or ADWS traffic, since outbound traffic is generally open on servers in most environments. If this is not the case in your environment, you will need to update the script or manually add a rule.

3. After running the firewall script to open the inbound ports, check the firewall rules to confirm that the new Keyfactor rule has been added by opening an administrative command prompt and running the following command:

   ```
   wf.msc
   ```

4. Click **Inbound Rules** and verify that the new rule "Keyfactor Gateway Connector RPC-IN" exists and is enabled. Verify that the existing rule: "COM+ Network Access (DCOM-In)" is enabled.

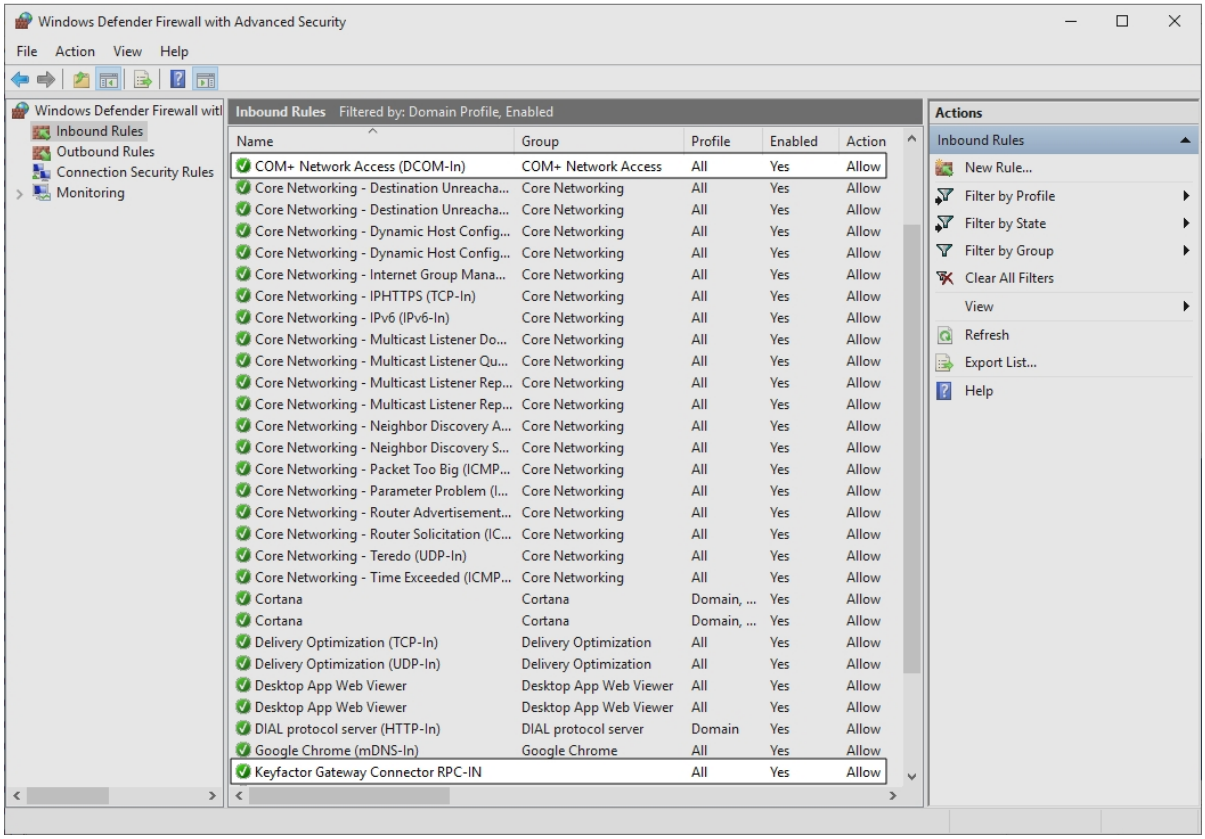

Figure 4: Firewall Rules

2.7 Create or Identify Certificate Templates or Profiles

The Keyfactor Remote CA Gateway uses certificate templates or profiles from the on-premise CA that match templates hosted in the managed forest to support enrollment for certificates through the Keyfactor Remote CA Gateway. When you enroll for a certificate in the managed instance of Keyfactor Command, you make a request using the managed forest template and the corresponding on-premise certificate template or profile is used. Template mappings are configured in the Keyfactor Remote CA Configuration Portal. No synchronization of templates occurs between the on-premise and managed environments. Before you configure the Keyfactor Remote CA Gateway Connector in the configuration portal, you need to create new or identify existing templates or profiles that will be used for enrollment in the managed environment.

Microsoft CAs

For Microsoft CAs, templates are stored in Active Directory. You will need a list of the *template names* (vs template display names) of the templates that will be used when configuring the Keyfactor Remote CA Configuration Portal.

EJBCA CAs

For EJBCA CAs, you will need a list of the certificate profiles configured as available on the profile of the end entity that you will use to make the connection from the gateway connector (see Keyfactor Remote CA Configuration Portal CA Connection Account on page 6).

Tip: If Keyfactor has upgraded your hosted instance to be compatible with Keyfactor Command version 10 or higher, you will need both the *certificate profile name* and the *end entity profile name* handy.

3.0 Installing and Configuring

The Keyfactor Remote CA Gateway Connector may be installed on either Windows or Linux:

- Install the Keyfactor Remote CA Gateway Connector on Windows below
- Install the Keyfactor Remote CA Gateway Connector on Linux on page 17

Important: Connections to Microsoft CAs are only supported from instances of the Keyfactor Remote CA Gateway Connector running on Windows.

3.1 Install the Keyfactor Remote CA Gateway Connector on Windows

To begin the Keyfactor Remote CA Gateway Connector installation on Windows, place the installation files in a temporary working directory on the Windows server and:

1. On the Windows machine on which you wish to install the gateway connector, open a PowerShell window using the "Run as Administrator" option and change to the temporary directory where you placed the installation files.

2. In the PowerShell window, run the following commands to populate a variable with the credentials for the service account, if you plan to run the gateway connector as a custom service account rather than the default of Network Service (see Keyfactor Remote CA Gateway Connector Service on page 5), and populate a variable with the client secret for the access token (see Identify the Keyfactor Remote CA Gateway Connector Access Token on page 5):

   ```
   $credService = Get-Credential
   $clientSecret = ConvertTo-SecureString -Force -AsPlainText 'MySecret'
   ```

   Enter the appropriate username and password when prompted and enter the appropriate client secret in place of *MySecret*. Usernames should be given in DOMAIN\username format.

   Or, to avoid being prompted for credentials:

   ```
   $serviceUser = "DOMAIN\myserviceusername"
   $servicePassword = "MySecurePassword"
   $secServicePassword = ConvertTo-SecureString $servicePassword -AsPlainText -Force
   $credService = New-Object System.Management.Automation.PSCredential ($serviceUser,
   $secServicePassword)
   $clientSecret = ConvertTo-SecureString -Force -AsPlainText 'MySecret'
   ```

3. In the PowerShell window, run the Install-GatewayConnector.ps1 script using the following syntax to begin the installation:

-URL

This is the URL to the Keyfactor Remote CA Service server. SSL is required to connect to the Keyfactor Remote CA Service server. This URL should be provided to you by your Keyfactor representative.

This parameter is **required**.

-Scope

This is the mechanism by which the gateway connector makes a request for the specific access that it needs. This value should have been provided to you by your Keyfactor representative.

This parameter is **required**.

-AuthURL

This is the URL to the service providing OAuth2 authentication. This value should have been provided to you by your Keyfactor representative.

This parameter is **required**.

-ClientId

This is a string issued by the authentication provider that identifies the application authenticating. This value should have been provided to you by your Keyfactor representative.

This parameter is **required**.

-ClientSecret

This is the secret shared between the components of the Keyfactor Remote CA Gateway implementation and the authentication provider. This value should have been provided to you by your Keyfactor representative.

This parameter is **required**.

-Source

Specify this parameter to point to a directory containing the installation files other than the directory in which the Install-GatewayConnector.ps1 file is found. This parameter is used primarily if a copy of the Install-GatewayConnector.ps1 file is made in an alternate directory, updated with some customizations, and then used for installation without being copied back to the directory where the remaining installation files are located.

-Destination

This parameter specifies a location in which to install the gateway connector that is other than the default. The default installation location is:

```
C:\Program Files\Keyfactor\Keyfactor Gateway Connector
```

This parameter cannot be used in conjunction with the *InPlace* parameter.

-InPlace

This parameter is used to indicate that the installation should occur in the current directory where the install files are located and no files should be copied to another location on the machine.

This parameter cannot be used in conjunction with the *Destination* parameter.

-ServiceSuffix

This parameter is used to add a suffix to the root service name of *KeyfactorGatewayConnector* (e.g. *Instance1* for a resulting service name of *KeyfactorGatewayConnector-Instance1*). This is used primarily for implementations where the gateway connector will be installed multiple times on the same server.

If this parameter is not specified, the default service name of *KeyfactorGatewayConnector-Default* will be used—with a display name of *Keyfactor Gateway Connector Service (Default)*.

-ServiceUser

This is the credential object of the service account the gateway connector service will run as (see Keyfactor Remote CA Gateway Connector Service on page 5). It is provided as a PSCredential object.

If this parameter is not specified, the built-in Network Service account will be used.

-Name

Specifying this parameter allows you to override the name the gateway connector would by default use to register itself in the Keyfactor Remote CA Configuration Portal.

By default, the gateway connector uses the value of the machine hostname without the domain.

-Force

Specifying this parameter causes the installation to warn and continue on certain potential problems, including:

- A service with either the default service name or the service name specified with the *ServiceSuffix* parameter already exists. The service will be overwritten if *Force* is specified.
- Either the default installation location or the location specified with the *Location* parameter is not empty. The install will occur to the specified or default location anyway and files may be overwritten if *Force* is specified.

If this parameter is not specified and any of these problems are encountered, the installation will terminate prematurely.

The output from the command should look similar to the following, given the example commands shown.

```
$serviceUser = "KEYEXAMPLE\svc_kyfconnect"
$servicePassword = "MySecurePassword"
$secServicePassword = ConvertTo-SecureString $servicePassword -AsPlainText -Force
$credService = New-Object System.Management.Automation.PSCredential ($serviceUser, $secSer-
vicePassword)
$mySecret = ConvertTo-SecureString -Force -AsPlainText 'h-TM53PDzM9w+!56Gk.z4#TmME=G4=h4r$'

.\Install-GatewayConnector.ps1 -URL https://kyf101.keyfactorpki.com/RemoteCAManagement -Scope
api://a12b345c-1234-8qqe-7521-1d91e647f7bg/.default -AuthURL https://-
login.microsoftonline.com/mycred.onmicrosoft.com/oauth2/v2.0/token -ClientID 1a234567-8b90-123c-
d456-7e89f0123ghi -Name websrvr12.keyexample.com -ServiceUser $credService -clientSecret
$mySecret

    Directory: C:\Program Files\Keyfactor\Keyfactor Gateway Connector\logs

Mode                 LastWriteTime         Length   Name
----                 -------------         ------   ----
-a----          6/23/2021 8:21 AM            0     Gateway_Connector_Log.txt

Created new file at C:\Program Files\Keyfactor\Keyfactor Gateway Connector\logs\Gateway_
Connector_Log.txt
```

4. Review the output from the installation to confirm that no errors have occurred.

The script creates a directory, C:\Program Files\Keyfactor\Keyfactor Gateway Connector by default, and places the gateway connector files in this directory. Log files are found in C:\Program Files\Keyfactor\Keyfactor Gateway Connector\Logs by default, though this is configurable (see Configure Logging on page 21).

The gateway connector service, by default given a display name of *Keyfactor Gateway Connector Service (Default)*, should be automatically started at the conclusion of the install and configured to restart on reboot.

Tip: Once the installation of the gateway connector is complete, you need to use the Keyfactor Remote CA Configuration Portal to approve the gateway connector (see Gateway Connectors on page 42) and configure CAs (see Certificate Authorities on page 28). You can then add these CAs in Keyfactor Command as per the the *Keyfactor Command Reference Guide* instructions.

3.2 Install the Keyfactor Remote CA Gateway Connector on Linux

To begin the Keyfactor Remote CA Gateway Connector installation on Linux, place the installation files in a temporary working directory on the Linux server and:

1. On the Linux machine on which you wish to install the gateway connector, in a command shell change to the temporary directory where you placed the installation files.
2. Use the chmod command to make the install.sh script file executable. The file ships in a non-executable state to avoid accidental execution. For example:

   ```
   sudo chmod +x install.sh
   ```

3. In the command shell, run the install.sh script as root using the following syntax to begin the installation:

   ### --url

   This is the URL to the Keyfactor Remote CA Service server. SSL is required to connect to the Keyfactor Remote CA Service server. This URL should be provided to you by your Keyfactor representative.

   This parameter is **required**.

   ### --scope

   This is the mechanism by which the gateway connector makes a request for the specific access that it needs. This value should have been provided to you by your Keyfactor representative.

   This parameter is **required**.

   ### --auth-url

   This is the URL to the service providing OAuth2 authentication. This value should have been provided to you by your Keyfactor representative.

   This parameter is **required**.

--client-id

This is a string issued by the authentication provider that identifies the application authenticating. This value should have been provided to you by your Keyfactor representative.

This parameter is **required**.

-client-secret

This is the secret shared between the components of the Keyfactor Remote CA Gateway implementation and the authentication provider. This value should have been provided to you by your Keyfactor representative.

This parameter is **required**.

Note: If you prefer to avoid providing the client secret at the command line (and storing it in command history), use an input file instead as follows:

a. Create a file that contains just the client secret. For example:

```
vi my_secret_file
```

b. When using the client secret parameter, reference the file. For example:

```
--client-secret $(cat my_secret_file)
```

c. Delete the client secret file after the install is complete. For example:

```
rm my_secret_file
```

--name

Specifying this parameter allows you to override the name the gateway connector would by default use to register itself in the Keyfactor Remote CA Configuration Portal.

By default, the gateway connector uses the results from a hostname lookup for the server's name.

--destination

This parameter specifies a location in which to install the gateway connector that is other than the default. The default installation location is:

```
/opt/keyfactor-gateway-connector
```

This parameter cannot be used in conjunction with the *in-place* parameter.

--log-location

This parameter specifies a path for the output log directory. The default location is the logs directory under the installed directory, which by default is:

```
/opt/keyfactor-gateway-connector/logs
```

--service-user

This is the local Linux service account that the service will run as (see Keyfactor Remote CA Gateway Connector Service on page 5). It should be entered as just the user name. Entry of a password for this service account is not required. You may either create this account prior to running the installation script (or use an existing account) or use the *force* parameter to generate the account automatically during the installation process.

If this parameter is not specified, the default service account name of *keyfactor-gatewayconnector* will be used.

--service-suffix

This parameter is used to add a suffix to the root service name of *keyfactor-gateway-connector* (e.g. *instance1* for a resulting service name of *keyfactor-gateway-connector-instance1*). This is used primarily for implementations where the gateway connector will be installed multiple times on the same server.

If this parameter is not specified, the default service name of *keyfactor-gateway-connector-default* will be used.

-v, --verbose

Specifying this parameter causes verbose messages to be output during installation.

-f, --force

Specifying this parameter causes the installation to warn and continue on certain potential problems, including:

- The local service account does not exist. The default user will be created if *force* is specified.
- The appsettings.json file does not exist or is invalid. A new one will be created if *force* is specified.
- The secretAppsettings.json file does not exist or is invalid. A new one will be created if *force* is specified.
- A service with either the default service name or the service name specified with the *service-suffix* parameter already exists. The service will be overwritten if *force* is specified.

- Either the default installation location or the location specified with the *destination* parameter is not empty. The install will occur to the specified or default location anyway and files may be overwritten if *force* is specified.

If this parameter is not specified and any of these problems are encountered, the installation will terminate prematurely. See also the *what-if* parameter.

--in-place

This parameter is used to indicate that the installation should occur in the current directory where the install files are located and no files should be copied to another location on the machine.

This parameter cannot be used in conjunction with the *destination* parameter.

--what-if

This parameter is used to test the installation command without actually installing in order to see any errors that might arise and correct them before installing.

The output from the command should look similar to the following, given the example commands shown.

```
vi my_secret_file

sudo ./install.sh --url https://kyf101.keyfactorpki.com/RemoteCAManagement --scope
api://a12b345c-1234-8qqe-7521-1d91e647f7bg/.default --auth-url https://-
login.microsoftonline.com/mycred.onmicrosoft.com/oauth2/v2.0/token --client-id 1a234567-8b90-
123c-d456-7e89f0123ghi --name appsrvr162.keyexample.com --client-secret $(cat my_secret_file) --
force
Gateway connector installation directory does not exist and will be created.
Gateway connector log directory does not exist and will be created.
Creating user keyfactor-gatewayconnector
Creating install directory...
Creating gateway connector log location...
Installing Keyfactor Gateway Connector...
Saving app settings
Saving secret app settings
Setting file permissions
Creating service unit file...
Created symlink /etc/systemd/system/multi-user.target.wants/keyfactor-gateway-connector-
default.service → /etc/systemd/system/keyfactor-gateway-connector-default.service.
Starting Gateway Connector...
```

4. Review the output from the installation to confirm that no errors have occurred.

The script creates a directory, /opt/keyfactor-gateway-connector by default, and places the gateway connector files in this directory. Log files are found in /opt/keyfactor-gateway-connector/logs by default, though this is configurable (see Configure Logging below).

The gateway connector service, by default named keyfactor-gateway-connector-default.service, should be automatically started at the conclusion of the install and configured to restart on reboot.

Note: Once the installation of the gateway connector is complete, you need to use the Keyfactor Remote CA Configuration Portal to approve the gateway connector (see Gateway Connectors on page 42) and configure CAs (see Certificate Authorities on page 28). You can then add these CAs in Keyfactor Command as per the the *Keyfactor Command Reference Guide* instructions.

3.3 Optional Configuration

Once the installation is complete, the Keyfactor Remote CA Gateway Connector should be running and ready to communicate with the Keyfactor Remote CA Service. The initial installation allows the gateway connector to register itself with the portal and, once approved and configured appropriately in the portal (see Using the Remote CA Configuration Portal on page 27), provide the connection from the portal to the CA to allow for certificate synchronization, enrollment and management.

This section details some post-install configuration steps that may be helpful during ongoing use or troubleshooting.

3.3.1 Configure Logging

By default, the Keyfactor Remote CA Gateway Connector places its log files in the logs directory under the installed directory, generates logs at the "Info" logging level and stores logs for two days before deleting them. If you wish to change these defaults, follow the directions below for your installation type.

Windows Installations

1. On the Windows 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 Keyfactor Remote CA Gateway Connector. The file is located in the configuration directory within the install directory, which is the following directory by default:

   ```
   C:\Program Files\Keyfactor\Keyfactor Gateway Connector\configuration
   ```

3. Your Nlog.config file may have a slightly different layout than shown here, but it will contain the four fields highlighted in Figure 5: Keyfactor Remote CA Gateway Connector on Windows NLog.config File. The fields you may wish to edit are:

- fileName="C:\Program Files\Keyfactor\Keyfactor Gateway Connector\logs\Gateway_Connector_Log.txt"

  The path and file name of the active gateway connector log file.

  > **Important:** If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\ConnectorLogs) and grant the service account under which the Keyfactor Remote CA Gateway Connector is running full control permissions on this directory.

- archiveFileName="C:\Program Files\Keyfactor\Keyfactor Gateway Connector\logs\Log_Archive_{#}.txt"

  The path and file name of previous days' gateway connectorr log files. The gateway connector 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" writeTo="logfile"

  The level of log detail that should be generated and output to the log file. 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:\Program Files\Keyfactor\Keyfactor Gateway Connector\logs\Gateway_Connector_Log.txt"
    layout="${longdate} ${logger} [${level}] - ${message} - ${exception:format=StackTrace}${newline}"
    archiveFileName="C:\Program Files\Keyfactor\Keyfactor Gateway Connector\logs\Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling"
    maxArchiveFiles="2" archiveAboveSize="2147483648" />
  </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="${logger} ${message}" />
  <target xsi:type="EventLog" name="eventLog" source="CoreTest2" eventId="${event-properties:item=eventID}" category="${event-properties:item=categoryID}"
  layout="${event-properties:item=message}" />
</targets>
<rules>
  <logger name="*-EVENT" minlevel="Info" writeTo="eventLog" final="true" />
  <logger name="*" minlevel="Info" writeTo="logfile" />
</rules>
```

Figure 5: Keyfactor Remote CA Gateway Connector on Windows NLog.config File

Linux Installations

1. On the Keyfactor Remote CA Gateway Connector machine where you wish to adjust logging, open a command shell and change to the directory in which the gateway connector is installed. By default this is /opt/keyfactor-gateway-connector.

2. In the command shell in the directory in which the gateway connector is installed, change to the configuration directory.

3. Using a text editor, open the nlog.config file in the configuration directory. Your nlog.config file may have a slightly different layout than shown here, but it will contain the five fields highlighted in the below figure. The fields you may wish to edit are:

    - fileName=”/opt/keyfactor-gateway-connector/logs/Gateway_Connector_Log.txt”

      The path and file name of the active gateway connector log file.

      > ⚠ **Important:** If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. /opt/kyflogs) and grant the service account under which the keyfactor-gateway-connector-default service is running full control permissions on this directory.

    - archiveFileName=”/opt/keyfactor-gateway-connector/logs/Log_Archive_{#}.txt”

      The path and file name of previous days’ gateway connector log files. The gateway connector 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” writeTo=”logfile”

      The level of log detail that should be generated and output to the log file. 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="/opt/keyfactor-gateway-connector/logs/Gateway_Connector_Log.txt"
      layout="${longdate} ${logger} [${level}] - ${message} - ${exception:format=StackTrace}${newline}"
      archiveFileName="/opt/keyfactor-gateway-connector/logs/Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling"
      maxArchiveFiles="2" archiveAboveSize="2147483648"/>
  </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="${logger} ${message}"/>
  <target xsi:type="EventLog" name="eventLog" source="CoreTest2"
      eventId="${event-properties:item=eventID}" category="${event-properties:item=categoryID}" layout="${event-properties:item=message}" />
</targets>
<rules>
  <logger name="*-EVENT" minlevel="Info" writeTo="eventLog" final="true" />
  <logger name="*" minlevel="Info" writeTo="logfile" />
</rules>
```

Figure 6: Keyfactor Remote CA Gateway Connector on Linux NLog.config File

3.3.2 Start the Service

The Keyfactor Remote CA Gateway Connector service runs on the gateway connector server and controls gateway connector communications with the Keyfactor Remote CA Service server and the on-premise CA(s). During the configuration process you set the service account under which the gateway connector service will run. The service should start automatically at the conclusion of the installation. To check to see if it’s running and start it if necessary, follow the directions below for your installation type.

Windows Installations

The service on Windows is added with a display name of Keyfactor Gateway Connector Service (Default) by default.

1. On the Keyfactor Remote CA Gateway Connector server, open the Services MMC.
2. In the Services MMC confirm that the Keyfactor Gateway Connector Service is set to a Startup Type of Automatic (if desired). If the service is not running, click the green arrow to start it.

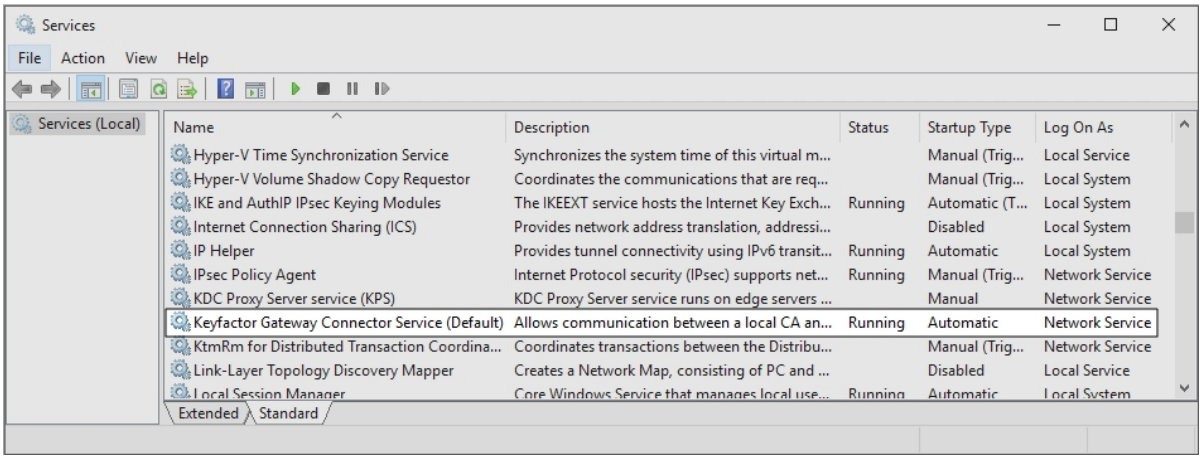

Figure 7: Keyfactor Gateway Connector Service

Note: Your service will have a name other than *(Default)* following *Keyfactor Gateway Connector Service* if you opted to use the *ServiceSuffix* installation parameter.

Linux Installations

The service on Linux is added as keyfactor-gateway-connector-default by default, so when referencing it in startup commands, it should be referenced by this name, including case. For example:

```
systemctl start [stop] [restart] [status] keyfactor-gateway-connector-default.service
```

Note: Your service will have a name other than *default* following *keyfactor-gateway-connector-* if you opted to use the *service-suffix* installation parameter.

3.3.3 Install a Custom Connector Extension

Connections to specific types of CAs are made with connector extensions for the Keyfactor Remote CA Gateway Connector. Out-of-the-box, extensions for Microsoft and EJBCA are provided. In addition, custom extensions may be built to support a variety of other types of CAs. The installation media for the gateway connectors for Microsoft and EJBCA includes the extensions. If you have a custom extension, you will need to install the standard gateway connector (without either Microsoft or EJBCA extensions) and then install your custom extension. Once you have your extension ready, install it as follows.

1. On the Keyfactor Remote CA Gateway Connector server, locate the extensions directory within the install directory. By default, this is:

   ```
   Windows: C:\Program Files\Keyfactor\Keyfactor Gateway Connector\extensions
   ```

```
Linux: /opt/keyfactor-gateway-connector/extensions
```

2. Under the extensions directory, create a new directory with an appropriate name for the extension (e.g. MyExtension). This name is for reference only and does not need to match any names used elsewhere.
3. Place the DLLs for the extension along with any other supporting files needed for the extension in the new directory.
4. Restart the Keyfactor Remote CA Gateway Connector service (see Start the Service on page 24).
5. In the Keyfactor Remote CA Configuration Portal, approve the gateway connector and wait for the connector to show *Connected* before continuing on with configuration (see Certificate Authorities on page 28).

Contact your Keyfactor representative for more information about custom solutions.

Note: A given instance of the Keyfactor Remote CA Gateway Connector can support only one connector extension. If you have more than one type of CA (e.g. both Microsoft and a type requiring a custom extension), you will need more than one instance of the Keyfactor Remote CA Gateway Connector.

4.0 Using the Remote CA Configuration Portal

The Keyfactor Remote CA Configuration Portal is a web-based application that you can open in any supported browser. The default URL for the portal is (where PORTAL_SERVER_FQDN is the FQDN of the Keyfactor Remote CA Configuration Portal server in the managed environment):

```
https://PORTAL_SERVER_FQDN/RemoteCAManagement
```

Within the portal, you:

- Approve Keyfactor Remote CA Gateway Connectors to allow them to act as the communication link between your on-premise CA(s) and the Keyfactor Command instance in the managed environment.
- Configure CA records that:
  - Link the gateway connector(s) to the CA.
  - Grant permissions for on-premise users to the CA for enrollment and management via Keyfactor Command.
  - Define templates/profiles that will be available for enrollment from the CA via Keyfactor Command.
  - Configure a service account user that allows the Keyfactor Remote CA Gateway access to read records from the on-premise CA, enroll for certificates, and perform management tasks.
  - Configure other supporting settings.

Once the gateway connectors are approved and appropriate CA records created, the remaining configuration is completed through the Keyfactor Command Management Portal. The first step in the Management Portal is to configure a CA as per the the *Keyfactor Command Reference Guide*[1].

The Keyfactor Remote CA Configuration Portal is supported in the following browsers:

- Chrome version 65.0.3325 or higher
- Firefox version 59.0 or higher
- Microsoft Edge version 42.17134 or higher

A given instance of the Keyfactor Remote CA Gateway Connector and associated Keyfactor Remote CA Service and Keyfactor Remote CA Configuration Portal can support only one type of CA. If you have more than one type of CA (e.g. both Microsoft and EJBCA), you will need more than one instance of these.

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

4.1 Certificate Authorities

On the Certificate Authorities page of the Keyfactor Remote CA Configuration Portal you create CA records that:

- Link the Keyfactor Remote CA Gateway Connector(s) to the CA.
- Grant permissions for on-premise users to the CA for enrollment and management via Keyfactor Command.
- Define templates/profiles that will be available for enrollment from the CA via Keyfactor Command.
- Configure a service account user that allows the Keyfactor Remote CA Gateway access to read records from the on-premise CA, enroll for certificates, and perform management tasks.
- Configure other settings to fully enable the CA, which vary depending in the type of CA.

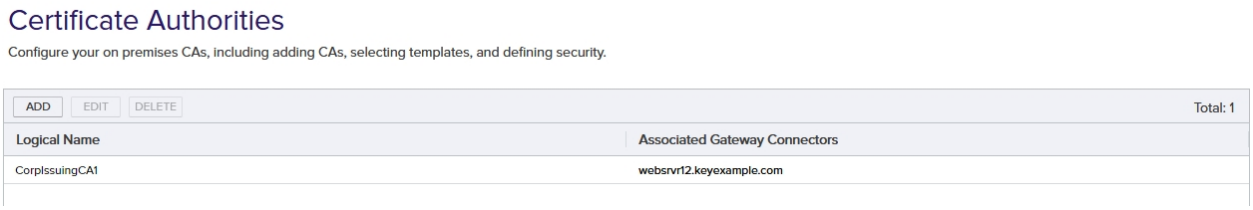

Figure 8: Certificate Authorities Grid

4.1.1 Add a New Certificate Authority

Before you can add a new CA record, you must have at least one Keyfactor Remote CA Gateway Connector that appears on the Gateway Connector page and shows as approved and connected (see Gateway Connectors on page 42).

Note:
- For the given EJBCA instance, validation will occur to ensure that the CA name provided is valid. An invalid name will prevent the CA from being saved in the Remote CA Portal.
- For EJBCA, validation will occur to ensure that the version is 7.8.1 or greater and if not, an error message is displayed.

To create a new CA record:

1. In the Hosted Configuration Portal select the Certificate Authorities page.
2. On the Certificate Authorities grid, click **Add** to create a new CA record.
3. When you open the Certificate Authorities dialog, you will see several tabs. Complete the dialog using the following instructions:

Basic Tab

On the Basic tab:

a. Enter the logical name of the CA you wish to connect to using the gateway connector(s) in the *Logical Name* field and click **Add**/**Edit**. If desired, click **Import** and select from CAs pre-configured on the Keyfactor Remote CA Configuration Portal server, if available.

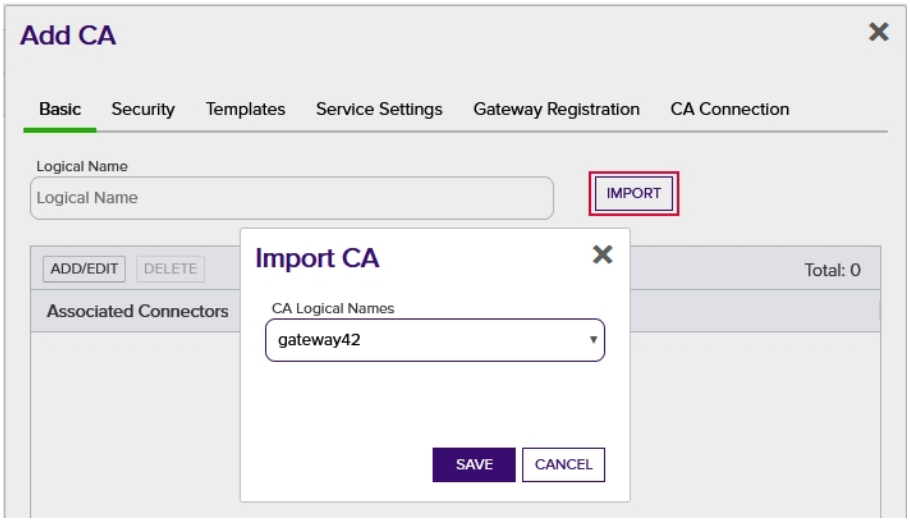

Figure 9: Select Gateway Connectors

b. In the Gateway Connectors dialog, check the box next to one or more gateway connector(s) that should be associated with the CA you entered. You may choose to select more than one gateway connector for load balancing or redundancy purposes.

   Connections to Microsoft CAs are only supported from gateway connectors running on Windows.

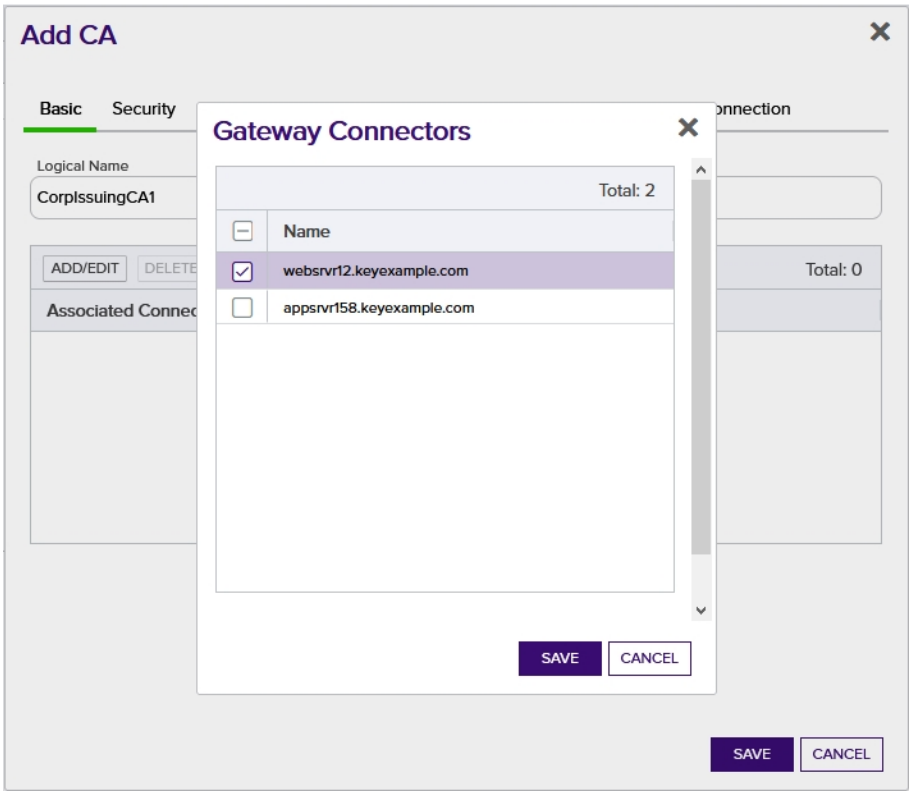

Figure 10: Select Gateway Connectors

c. Click **Save** to save the gateway connector selection.

Security Tab

On the Security tab you define which domain accounts can access the on-premise CA via the Keyfactor Remote CA Gateway and what level of permissions those users will have. The permission levels can be understood in the context of the standard Microsoft CA permission levels:

- READ
  Enumerate and read contents of certificates. This is the equivalent of the Microsoft CA *Read* permission.
- ENROLL
  Request certificates from the CA. This is the equivalent of the Microsoft CA *Request Certificates* permission.
- OFFICER
  Perform certificate functions such as issuance and revocation. This is the equivalent of the Microsoft CA *Issue and Manage Certificates* permission.

- ADMINISTRATOR
  Configure/reconfigure settings in the Keyfactor Remote CA Configuration Portal. This is similar to the Microsoft CA *Manage CA* permission.

Valid permission settings are "Allow", "None", and "Deny". Security can be defined at the user or group level. "Deny" overrides any "Allow" permissions. "None" does not grant permissions, but it also does not deny them.

Permission for users and groups needs to be set as follows:

- The Keyfactor Remote CA Configuration Portal application pool service account (see Keyfactor Remote CA Configuration Portal Application Pool on page 6) needs **"ADMINISTRATOR": "Allow"** permissions. All management of the configuration of the Keyfactor Remote CA Gateway is done in the context of this user.
- Users who will enroll for certificates from the CA through the Keyfactor Remote CA Gateway need **"READ": "Allow"** and **"ENROLL": "Allow"** permissions.
- Users who will revoke certificates or approve or deny pending certificates outside of Keyfactor Command but connecting to the CA through the Keyfactor Remote CA Gateway need **"READ": "Allow"** and **"OFFICER": "Allow"** permissions.
- If you're using Keyfactor Command, the service account under which the application pool on the Keyfactor Command Management Portal server is running needs **"READ": "Allow"** permission and the service account under which the Keyfactor Command service (a.k.a. the timer service) is running needs **"READ": "Allow"** permission. **"READ": "Allow"** permission is needed for the application pool account to allow the Keyfactor Command Management Portal to display the status of the Keyfactor Remote CA Gateway as "online" on the Keyfactor Command Management Portal dashboard.
- If you're using Keyfactor Command and plan to use the renewal event handler in expiration alerts, the service account under which the Keyfactor Command service (a.k.a. the timer service) is running needs **"ENROLL": "Allow"** permission and the service account under which the application pool on the Keyfactor Command Management Portal server is running needs **"ENROLL": "Allow"** permission if you want to test alerts.
- Depending on your configuration choice for Kerberos delegation for the Keyfactor Remote CA Gateway, you need one of these:
  - If you're using Keyfactor Command and you *are not* planning to use Kerberos delegation for the Keyfactor Remote CA Gateway, the service account under which the application pool on the Keyfactor Command Management Portal server is running needs **"READ": "Allow"** and **"OFFICER": "Allow"** permissions.
  - If you're using Keyfactor Command and you *are* planning to use Kerberos delegation for the Keyfactor Remote CA Gateway, the users who will perform revocation and workflow (approve/deny pending certificates) through the Keyfactor Command Management Portal need **"READ": "Allow"** and **"OFFICER": "Allow"** permissions.

To configure security identities:

a. Click **Add**.

b. In the Add Security Identity dialog, enter an *Identity Name* (user or group) in DOMAIN\username format at the top and select the appropriate radio buttons for the access permissions that identity should have.

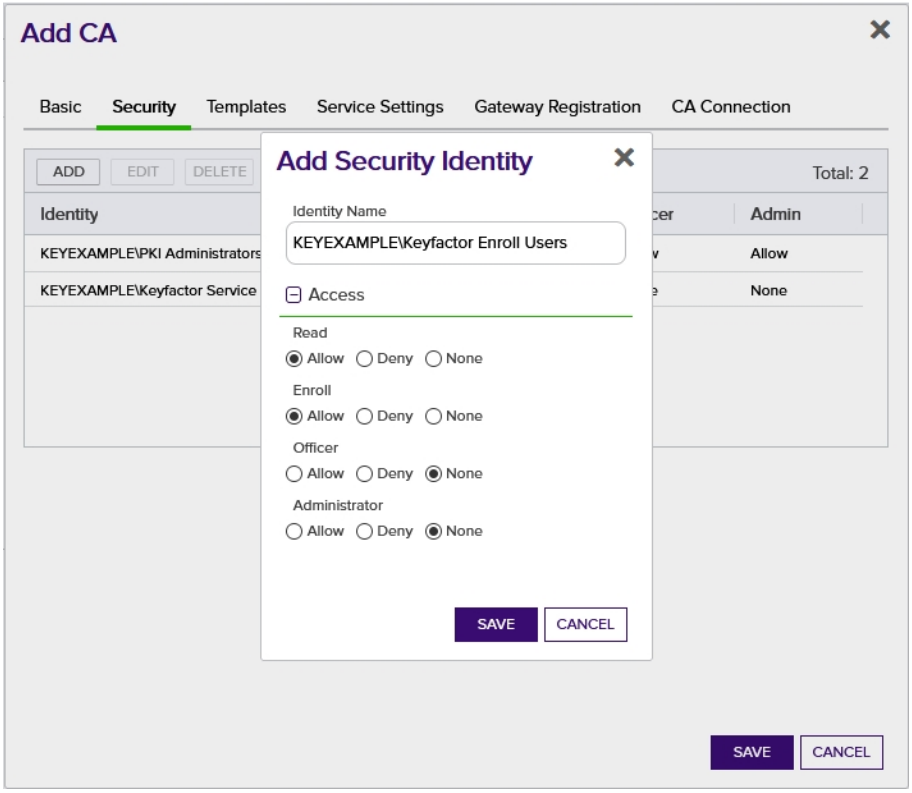

Figure 11: Configure Security Permissions

c. Click **Save** to save the security identity.

d. Repeat for each additional security identity.

Templates Tab

On the Templates tab, you configure the templates from the managed forest that will be available for enrollment through the Keyfactor Remote CA Gateway and associate them with product types from the on-premise CA. In the case of an on-premise Microsoft CA, the product type is also a template. Optionally, parameters can be configured for the templates. Neither the Microsoft CA nor the EJBCA CA make use of optional parameters for templates.

Tip: No synchronization of templates occurs between the managed forest and the on-premise forest.

To add a new template record:

a. Click **Add**.

b. In the Add Template dialog, enter a *Template Short Name* for a template that exists in the managed forest. You can find the list of existing templates for the managed forest in Keyfactor Command on the Certificate Templates page. For more information about certificate templates in Keyfactor Command, see the the *Keyfactor Command Reference Guide*[1].

c. Enter a *Product ID* for a product that exists in the on-premise CA. For a Microsoft CA, this will be another *Template Short Name*—in this case for a template available for enrollment from the on-premise CA. The template names in the managed and on-premise forest do not need to match, though a close approximation helps to keep track of things. For an EJBCA CA, these will be certificate profiles configured for support on the end entity profile that you configure on the *CA Connection Tab on page 37* (e.g. *SERVER*). Product IDs will vary for custom CA extensions.

   The Product ID field is case sensitive—SERVER is not equal to Server.
   If Keyfactor has upgraded your hosted instance and advised you that you need to make template configuration changes to the EJBCA CA, the changes are as follows:

   For each template, modify the template **ProductID** from the current value of the certificate profile name (CertificateProfileName) to include the end entity profile name (EndEntityProfileName) in this format: " *EndEntityProfileName_CertificateProfileName*".

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

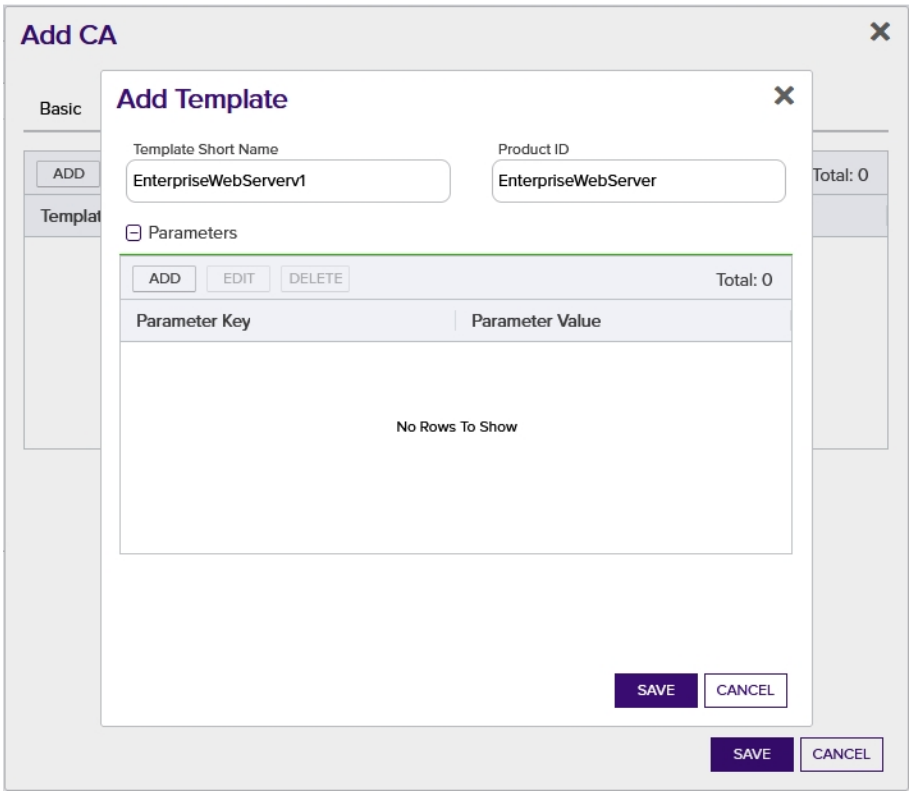

Figure 12: Add a Template for a Microsoft CA

d. To add an optional parameter, click **Add**. Enter a *Parameter Key* and *Parameter Value* and **Save**. Parameters are configured on a template-by-template basis, so if you are entering more than one template with the same set of parameters, the parameters must be configured for each template.

> **Note:** Neither the Microsoft CA nor the EJBCA CA makes use of optional parameters for templates.

e. Click **Save** to save the template.

f. Repeat for each additional template.

Service Settings Tab

On the Service Settings tab, configure the settings that control synchronization periods and timeouts for communications between the Keyfactor Remote CA Gateway and the on-premise CA. The settings include:

- View Idle (Minutes)

  A timeout interval setting, defined in minutes, for certificate synchronization jobs between the on-premise CA and the Keyfactor Remote CA Gateway. The suggested setting is 8 minutes.

- Full Scan Period (Hours)

  A scan period to specify how often, in hours, the Keyfactor Remote CA Gateway will perform a full synchronization from the on-premise CA to the gateway. The suggested setting is 24 hours.

- Partial Scan Period (Minutes)

  A scan period to specify how often, in minutes, the Keyfactor Remote CA Gateway will perform a partial or incremental synchronization from the on-premise CA to the gateway. The suggested setting is 10 minutes.

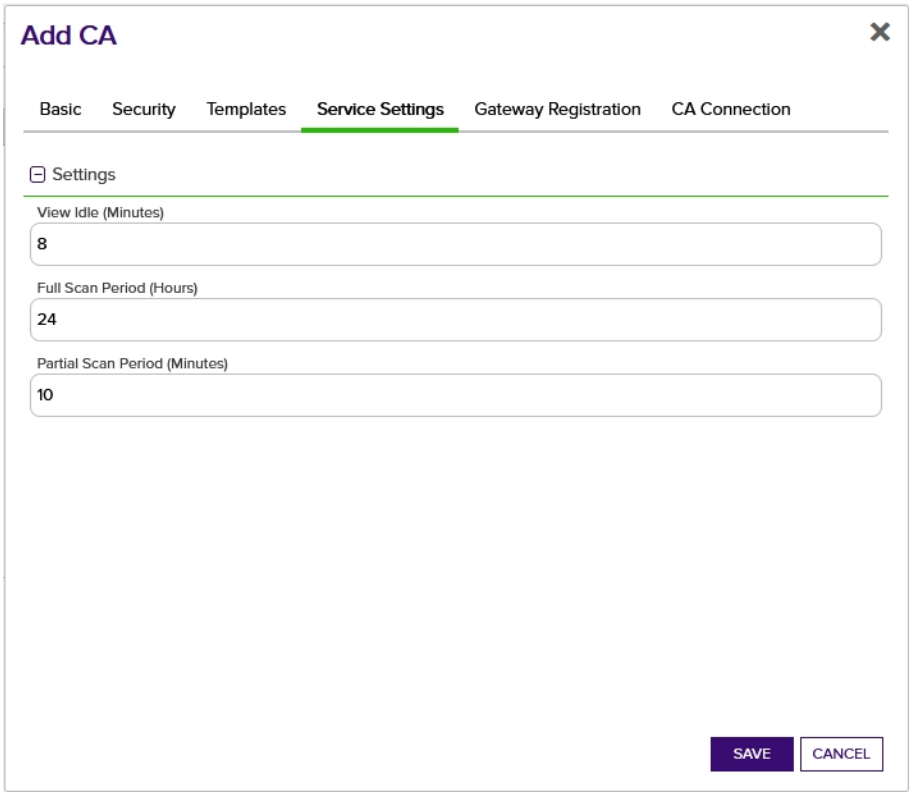

Figure 13: Configure Service Settings

Note: These settings control synchronization only between the on-premise CA and the Keyfactor Remote CA Gateway. Synchronization between the Keyfactor Remote CA

Gateway and Keyfactor Command is configured separately in the Keyfactor Command Management Portal.

Gateway Registration Tab

On the Gateway Registration tab, configure information regarding the intermediate or root chain certificate for the SSL certificate installed on the Keyfactor Remote CA Gateway server (see Configure a Certificate Root Trust for the Keyfactor Remote CA Gateway on page 7). The settings include:

- Store Name

  The store name where the chain certificate from the CA is located (e.g. *CA* for the *Intermediate Certification Authorities* store or *Root* for the *Trusted Root Certification Authorities* store).

- Store Location

  The store location where the chain certificate from the CA is located (e.g. *LocalMachine* for local computer).

- Thumbprint

  The thumbprint of the intermediate or root chain certificate from the CA.

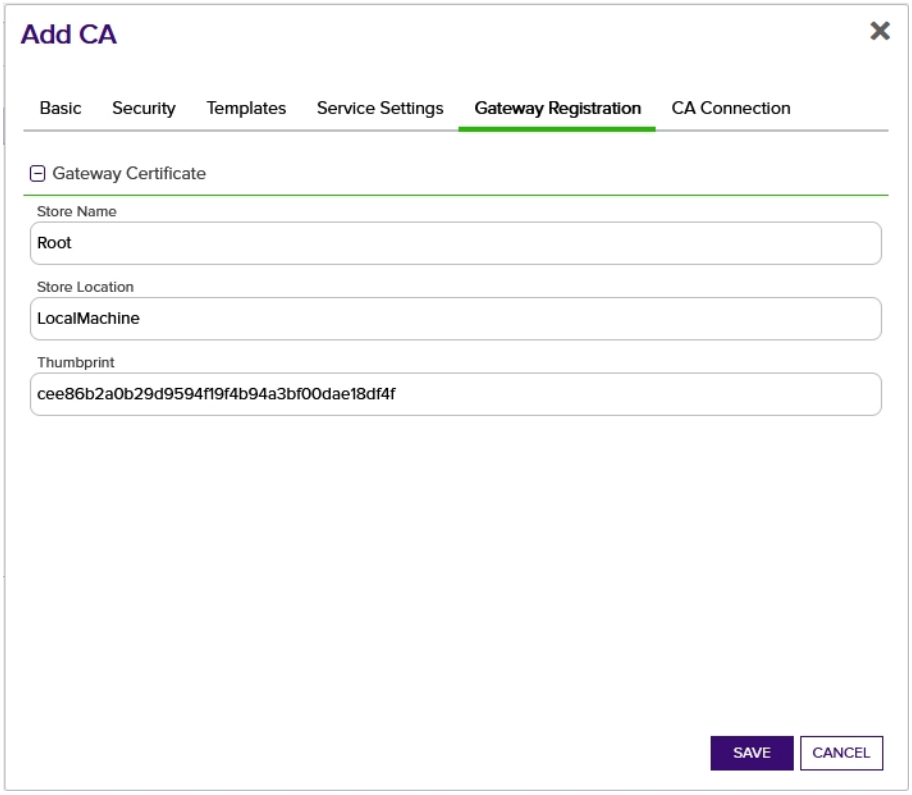

Figure 14: Configure Gateway Registration

Note: Contact your Keyfactor representative for assistance with these settings.

CA Connection Tab

On the CA Connection tab, configure connection settings specific to the on-premise CA. These will vary depending on the type of CA. Out-of-the-box, Microsoft and EJBCA CAs are supported.

Microsoft CAs

- Hostname

  The FQDN of the on-premise CA.

- Logical Name

  The logical name of the on-premise CA.

- Use Default Credentials

  A value of *True* or *False* that indicates whether the service account that the gateway connector is running as (see Keyfactor Remote CA Gateway Connector Service on

page 5) should also be used to make the connection to the CA for certificate synchronization (see Keyfactor Remote CA Configuration Portal CA Connection Account on page 6), enrollment and management. The gateway connector must be running as a domain account in order to set this value to *True*. If this value is set to *False*, the connection is made using the *Username* and *Password* provided in the CA connection information.

- Username

  The domain service account (e.g. DOMAIN\username) from the on-premise Active Directory used to make the connection to the CA if UseDefaultCredentials is *False* (see Keyfactor Remote CA Configuration Portal CA Connection Account on page 6). This account must have Read, Enroll, and Manage CA permissions on the CA and will need Read and Enroll permissions for any on-premise templates that will be used for enrollment.

- Password

  The password for the service account specified in *Username*.

- Standalone

  A value of *True* or *False* that indicates whether the on-premise CA is a standalone CA.

- Forest

  The Active Directory forest where the on-premise CA resides.

- Max Errors

  A numeric value that indicates the number of times an attempt should be made to read records from the on-premise CA before the job ends with a failure (e.g. 5).

- Batch Size

  A numeric value that indicates the number of records to retrieve from the on-premise CA in each batch of records during synchronization jobs (e.g. 100). If certificate records are especially large and errors are occurring on synchronization, it may help to reduce the batch size.

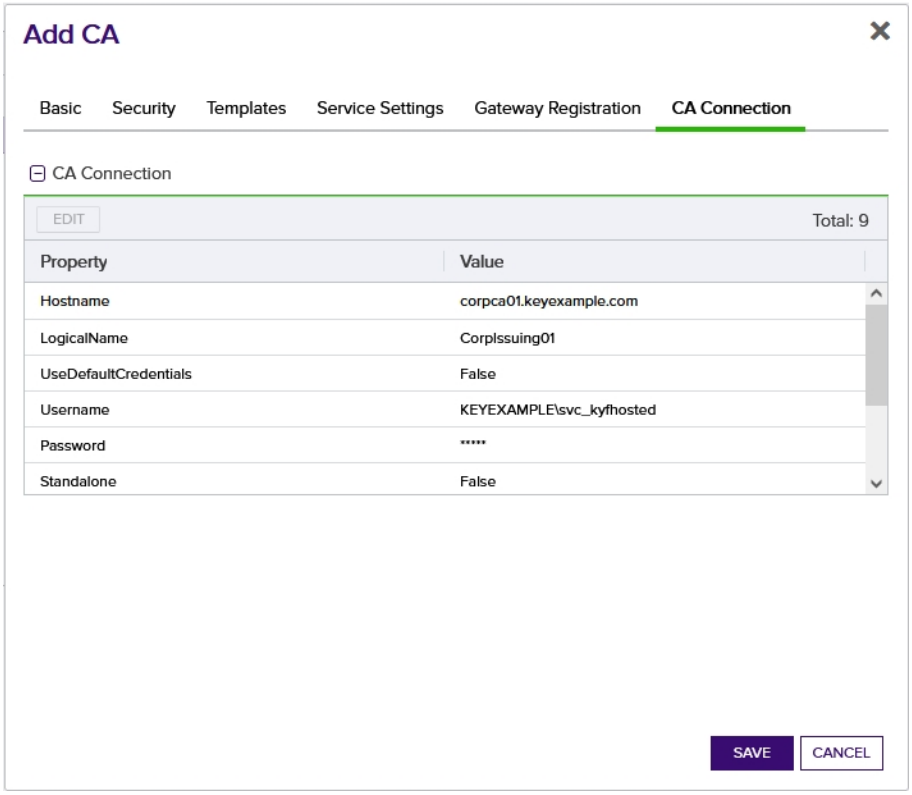

Figure 15: Configure CA Connection for a Microsoft CA

EJBCA CAs

- End Entity Profile Name

  The name of the end entity profile in the EJBCA database that should be used for certificate synchronization and enrollment.

  Once the remote EJBCA CA portal has been upgraded to be compatible with Keyfactor Command version 10, or higher, this field will not be used. Multiple end entity profiles will be supported, which will be configured in the templates section (see Templates Tab on page 32).
- Certificate Authority Name

  The logical name of the EJBCA CA.
- Username

  The end entity in the EJBCA database that should be associated with newly enrolled certificates.

Tip: An end entity record will automatically be created in the EJBCA database for this end entity if the end entity associated with the certificate specified in the *Client Certificate* setting has sufficient permissions to create end entities.

- Password

  The password for the end entity specified by *Username*.

- Client Certificate

  Certificate information for the certificate used to authenticate to the EJBCA database (see Keyfactor Remote CA Configuration Portal CA Connection Account on page 6) for synchronization, enrollment and management of certificates. Enter either the *StoreName* (e.g. *My* for the local computer personal store on Windows), *StoreLocation*, and *Thumbprint* for the certificate on the gateway connector server or the *CertificatePath* and *CertificatePassword* for a PKCS12 file containing the certificate on the gateway connector server.

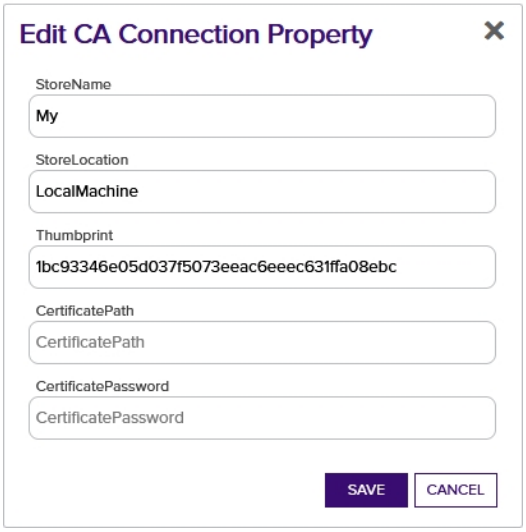

Figure 16: Configure Client Certificate for an EJBCA CA

  The end entity associated with this certificate does not need to be the same end entity specified by the *Username* setting.

- URL

  The URL pointing to your EJBCA server. If the URL provided does not have a virtual directory (/ejbca or otherwise) the /ejbca will be provided, otherwise it will use what is supplied in the URL.

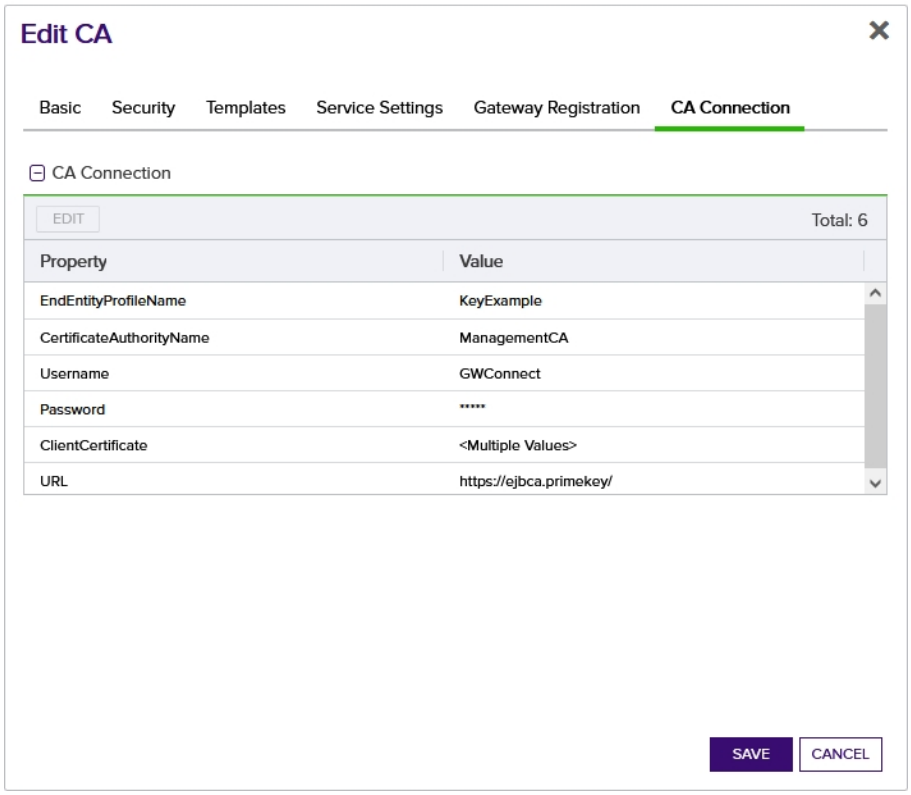

Figure 17: Configure CA Connection for an EJBCA CA

4. Click **Save** to save the configuration.

4.1.2 Edit or Delete a Certificate Authority

To edit an existing certificate authority record:

1. In the Keyfactor Remote CA Configuration Portal select the Certificate Authorities page.
2. On the Certificate Authorities grid, click **Edit**.
3. When you open the Certificate Authorities dialog, you will see several tabs. Update the dialog using the instructions shown for adding a certificate authority (see Add a New Certificate Authority on page 28).

   Note: Some fields in the CA record cannot be edited (e.g. the Logical Name on the Basic tab). If you need to modify a field that is non-editable, delete the record and re-create it.

4. Click **Save** to save the configuration.

To delete a certificate authority record, highlight a record in the Certificates Authorities grid and click **Delete**.

4.2 Gateway Connectors

On the Gateway Connectors page of the Keyfactor Remote CA Configuration Portal you can approve or deny Keyfactor Remote CA Gateway Connectors that have connected to the portal server and view the status of the connections.

- The *Last Seen* column indicates the last time the gateway connector contacted the server. Each gateway connector attempts to contact the portal server every 5 minutes.
- The *Approval Status* column indicates whether the gateway connector is new, has been approved, or has been denied. A gateway connector must have a status of approved before a CA record can be linked to it.
- The *Connected* column indicates whether the gateway connector is actively communicating with the portal server. A gateway connector must have a connected state of yes before a CA record can be linked to it.

Figure 18: Gateway Connectors Grid

To approve a gateway connector, highlight it in the Gateway Connectors grid and click **Approve**. Only one gateway connector may be approved at a time.

To deny a gateway connector, highlight it in the Gateway Connectors grid and click **Deny**. Only one gateway connector may be denied at a time.

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.

6.0 Copyright Notice

7.0 Appendices

7.1 Appendix A—Firewall Rules for Windows

This script configures the firewall appropriately to allow communication between the Keyfactor Remote CA Gateway Connector and the Azure-hosted Keyfactor Remote CA Service server. It should be run as an Enterprise Admin in an administrative PowerShell window on the Keyfactor Remote CA Gateway Connector machine.

Usage: **KeyfactorAnyGatewayFirewallRules.ps1**

Customizations:

- RemoteAddress
  This value needs to be customized to match the IP address of the Azure-based server hosting your Keyfactor Remote CA Configuration Portal. Your Keyfactor representative can provide this to you.
- Program
  If you installed the Keyfactor Remote CA Gateway Connector software in a non-default path, modify the script to change the *-Program* reference before executing it.

KeyfactorAnyGatewayFirewallRules.ps1 script contents:

```
Write-host Enabling "Keyfactor Gateway Connector RPC-IN"
New-NetFirewallRule -DisplayName "Keyfactor Gateway Connector RPC-IN" `
  -Description "An inbound rule to allow traffic to the Keyfactor Gateway Connector for enterprise
certificate enrollment." `
  -Direction Inbound `
  -RemoteAddress "10.9.8.7" `
  -LocalPort "RPC" `
  -RemotePort "49152-65535" `
  -Program "C:\Program Files\Keyfactor\Keyfactor Gateway Connector\GatewayConnector.exe" `
  -Protocol TCP `
  -Action Allow
Write-host Enabling "COM+ Network Access (DCOM-In)"
Set-NetFirewallRule -DisplayName "COM+ Network Access (DCOM-In)" -Enabled True
Get-NetFirewallRule -DisplayName "COM+ Network Access (DCOM-In)"
```

If your Keyfactor Remote CA Service and Keyfactor Remote CA Configuration Portal are on different servers, you will need to run the script for both servers.