Deploying YubiHSM 2 with ADCS

Yubico

May 12, 2022

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 YubiHSM 2 with Microsoft Active Directory Certificate Services

This document is intended to enable systems administrators to deploy YubiHSM 2 with YubiHSM Key Storage Provider. The expected outcome is that the Active Directory Certificate Services Certificate Authority (ADCS CA) root key is created securely on the device and that a hardware-based backup copy of key materials has been produced.

As a guide for deployment, it covers basic topics. Instructions should be modified as required for your specific environment. It is assumed that installation is performed on a single server destined to become a production or lab Certificate Authority root. It is also assumed that you are familiar with the concepts and processes of working with Microsoft ADCS.

Plan a public key infrastructure (PKI) that is appropriate for your organization. For guidance on setting up a PKI, see Microsoft's TechNet article on Public Key Infrastructure Design Guidance

We recommend that you install and test the installation and setup of the YubiHSM 2 in a test or lab environment before deploying to production.

Scenario: In a Windows PKI environment, protect the CA root key in hardware.

Benefits: YubiHSM 2 guards the CA root key and protects all signing and verification services using the root key.

Note: Although the screenshots in this guide are specific to Windows Server 2016, Server 2019 is also supported.

CHAPTER

TWO

PREREQUISITES AND PREPARATIONS

The audience of this document is expected to be an experienced systems administrator with a good understanding of Windows Server management. In addition, it helps to be familiar with the terminology, software and tools specific to YubiHSM 2. As a primer for these, refer to the *Terminology Used* chapter in this guide.

In order to follow the steps provided in this guide, be sure you meet the following prerequisites, having:

- Access to Microsoft Windows Server 2012, R2/2016, 2019 with Active Directory in an offline, air-gapped environment, such as a secure computer network that is physically isolated from unsecured networks such as the internet. You must also have elevated system privileges.
- YubiHSM 2 software and tools for Windows downloaded from the YubiHSM 2 Release page and available on the system to be used.
- Two (2) factory-reset YubiHSM 2 devices, one for deployment and one for backup in hardware.
- Key custodians identified as per local requirements and available to participate. For more information about key custodians and the associated **'M of N'** key shares, see the next chapter in this guide.

CHAPTER

THREE

KEY SPLITTING AND KEY CUSTODIANS

The preferred method for backing up the YubiHSM 2 keys calls for key splitting and restoring or regenerating, often referred to as setting up an **'M of n'** scheme (Shamir's Secret Sharing (SSS)). This process ensures no individual can export key material from the YubiHSM 2, and provides a way to control the import of key material that has been exported under wrap from one device into other devices. For example, you would export and import objects for backup purposes, as described in *Backing Up Key Material*.

The key that is split among a predetermined number (**n**) of **key custodians** (also known as key shareholders) is known as the wrap key. Each custodian receives their own unique share. In order to use the key, a minimum number of shares (**m**) must be present so that the key can be regenerated (sometimes called "rejoined"). This minimum number of custodians is called the **privacy threshold**. If this threshold is not attained, the wrap key cannot be regenerated. This minimum number, **'n'**, should be larger than one.

The exact number of key shares and the privacy threshold are determined by the requirements of your organization. If your organization has policies in place that define how this procedure should be performed, be sure you know these policies before proceeding. You should also have a predetermined practice in place specifying both:

- How the key shares must be recorded (written on paper, photographed, locally printed, or some other means) and
- How they must be stored between uses (for example, offsite archive, safety deposit box, sealed envelope).

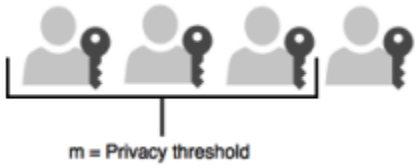

Figure: Privacy threshold

The YubiHSM Setup Tool enables you to perform the key splitting and assigning of shares to key custodians. To carry out the setup process, you need to know who the wrap key custodians will be. During setup, all key custodians must be physically present to record their shares. Exact instructions for key splitting and assigning of shares are given in *Configuring the Primary YubiHSM 2 Device*.

CHAPTER

FOUR

INSTALLING THE YUBIHSM 2 TOOLS AND SOFTWARE

To complete the procedures in this guide, install the YubiHSM 2 tools and software that will be needed for this.

4.1 About the YubiHSM Software

The following YubiHSM pieces of software are used in this guide. They are included as part of the archive file you downloaded.

4.2 Installation

A generic prompt, `*$*`, is used in command line examples in this document. Depending on your command line application, your prompt may be different.

Step 1 Unzip the downloaded archives of the SDK containing the YubiHSM libraries and tools and move the contents to an appropriate location.

Step 2 On your Windows system, run both installers:

- `yubihsm-cngprovider-windows-amd64.msi` (YubiHSM Key Storage Provider)
- `yubihsm-connector-windows-amd64.msi` (YubiHSM Connector for Windows)

4.3 Verifying the Default Configuration of the YubiHSM 2

The YubiHSM 2 device comes with a single factory-installed authentication key whose default password is `password`. As part of the configuration in this guide, this default authentication key will be destroyed. If the YubiHSM 2 is reset to its default configuration, any non factory-installed objects stored on it are also destroyed. Reset instructions can be found in Factory Reset.

We reiterate that you will need two YubiHSM 2 devices to complete all steps of this guide, because you will be deploying the first device and creating a backup of all key material on the second device.

To ensure that neither of the YubiHSM 2 devices have been tampered with, verify that they still have the default configuration by following the steps below:

Step 1 Do one of the following:

- If the application that calls the YubiHSM Connector is **running on a local host**, start the Connector with the command `yubihsm-connector` without additional parameters. In Windows Server 2012 SP2 or higher, `yubihsm-connector.exe` is located in `C:\Program Files\YubiHSM Connector\`.

- If the application is **running on a VM or a different server**, start the YubiHSM Connector on the host operating system in networking mode. For example, if the host machine's IP address is `192.168.100.252`, launch the Connector on the host OS with the command `yubihsm-connector -l 192.168.100.252:12345`

Tip: For testing or debugging the YubiHSM Connector, the flag `-d` can be set.

Step 2 To gain shell access to the YubiHSM 2, launch the YubiHSM Shell program by opening a Command Prompt and running the command `yubihsm-shell`. If a networked Connector is used, set the parameter `--connect <connector URL>`, for example:

```
$ yubihsm-shell --connector http://192.168.100.252:12345
```

Tip: For testing or debugging the YubiHSM Shell, the flag `-d` can be set.

Step 3 To connect to the YubiHSM 2, at the `yubihsm` command line, type `connect`. A message saying that you have a successful connection is displayed.

Step 4 To open a session with the YubiHSM 2, type `session open 1` (where 1 is the ID of the default authentication key pre-installed on the device).

Step 5 Type in the default password: `password`. A message confirming that the session has been set up successfully is displayed.

Step 6 You now have an administrative connection to the YubiHSM 2 and you can list the objects available by typing `list objects 0` and pressing **Enter**. Your results should be similar to the following:

```
id: 0x0001, type: authentication-key, sequence: 0
```

Step 7 To exit, type `quit`.

CHAPTER

FIVE

CONFIGURING THE PRIMARY YUBIHSM 2 DEVICE

The YubiHSM Setup program, which is part of the YubiHSM 2 toolset, is used to perform the initial configuration of the primary YubiHSM 2 device. This program configures the device with the requirements needed for deploying YubiHSM 2 to safely store the ADCS root CA key. Specifically, during the setup process the YubiHSM is configured so that:

- The necessary key material is generated on the device:
- One wrap key
- One application authentication key
- One audit key
- The wrap key is split among a determined number of key custodians, and each share is recorded by each custodian
- The authentication key and the audit key are exported under wrap to a file, located in the current working directory

To safeguard the integrity of the device, configuration must be performed in an air-gapped environment.

5.1 Summary of Configuration Steps

After you have inserted the primary device into the air-gapped system, the configuration steps are diagrammed in the following image, and listed below. They are described in detail in the next section, **Configuration Procedure**

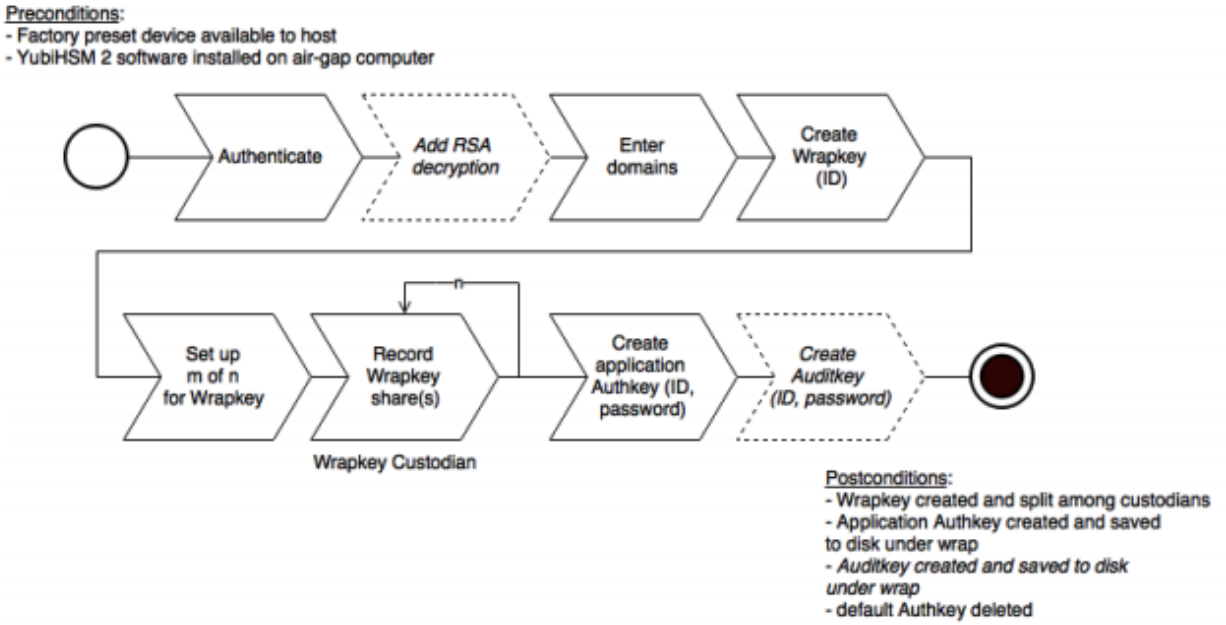

Figure: Pre- and Post-Conditions

5.1.1 Configuration Steps

1. Set up communication between the YubiHSM 2 tools and the device.
2. Start the configuration process.
3. Authenticate to the YubiHSM device.
4. Specify if you need to add RSA decryption (not required if you are using YubiHSM 2 exclusively with ADCS).
5. Enter the domains you need the application authentication key and audit key to be available in.
6. Create the wrap key.
7. Split the wrap key into shares, and record the number of shares required to rejoin the wrap key.
8. Create the application authentication key, which is used to authenticate to the device by the KSP in Windows so the KSP can perform operations in YubiHSM 2.
9. Create the audit key, to access the internal audit log of the device which holds information about the last 62 operations performed and so you can reset the audit log.

5.2 Configuration Procedure

Step 1 Enable communication with the YubiHSM 2 device by ensuring that the YubiHSM Connector service (`yhconsrv` in Windows) is running on the system where the device is inserted. You can validate that the connector is running properly by typing the following URI into your browser: http://127.0.0.1:12345/connector/status. Output should be similar to:

```
status=OK
serial=*
```

(continues on next page)

(continued from previous page)

```
version=1.0.0
pid=*
address=127.0.0.1
port=12345
```

Step 2 In your command line application (where `$` is the generic prompt used in this document; depending on your command line application, your prompt may be different), run YubiHSM Setup with the argument `ksp`. To do this, launch your command line application, navigate to the directory containing the YubiHSM Setup program, run `yubihsm-setup ksp` and press **Enter**.

Tip: The setup tool also has a help argument that you can call to learn more about its usage.

Step 3 To start the YubiHSM Setup process, type the default authentication key password: `password` and press **Enter**. A message appears, confirming that the default authentication key was used and that you have successfully authenticated to the device: `Using authentication key 0x0001.` Object IDs are displayed in the YubiHSM Setup Tool using hexadecimal numbers, in this case the default authentication key has ID `1`, or `0x0001` in hexadecimal.

Step 4 You are prompted to add RSA decryption capabilities. Do one of the following:

- If you plan to use your YubiHSM 2 with ADCS exclusively, you will not need the RSA decryption capabilities, you will only need signing capabilities. Type `n`.
- If you plan on using the same YubiHSM 2 device for purposes that do require decrypting RSA, type `y`.

Tip: If you are unsure what selection to make, type `n`.

Step 5 The next question to be answered is what domain(s) you need the application authentication key and audit key to be available in (the authentication and audit keys are generated after you create the wrap key). You will only need one domain for the purposes of completing this guide. Do the following:

Unless you have a requirement to assign more than one domain, type a single number between 1 to 16 and press **Enter**. In this guide, we assume that domain `1` was entered. Confirmation will look like the following:

```
got domains [
One
]
```

Step 6 In this step you will create a wrap key. The wrap key is very important as it allows you to export and import objects from and to the device. For example, you would export and import objects for backup purposes, as described in the section Backup Key Material. Do one of the following:

- To manually assign a wrap key ID, type the number and press Enter. As object ID `1` is already in use by the default application authentication key, we recommend you assign id `2` to the wrap key. Type `2` and press **Enter**.
- To allow the system to assign a wrap key ID automatically, type `0` and press **Enter**. A confirmation message is displayed:

```
Stored wrap key with ID 0x0002 on the device
```

Step 7 Now you will split the wrap key among a number of key custodians. For this example, we will assume that the wrap key is split into three shares, out of which at least two shares must be present in order to use the key. If there are not two key custodians present, the wrap key cannot be rejoined. When prompted, do the following:

- Enter the number of shares. In this example, enter `3`.
- Enter the privacy threshold. In this example, enter `2`.

When defined, the three wrap key custodians will each take their turn in front of the screen to record their respective share. A warning notice appears advising you that the shares are not stored anywhere.

Note: Be sure to record the shares and store them safely if you want to re-use the wrap key forthis device in the future.

To start recording the key shares, press **Enter**.

The first custodian records his share and leaves the screen. The next one enters and repeats the key share recording for the second share, and so on. Each custodian confirms by pressing y that the share was recorded before handing over to the next. The screen buffer is cleared before each share is presented. Following is an example of a share presented on the screen:

```
2-1-WWmTQj5PHGJQ4H9Y2ouURm8m75QkDOeYzFzOX1VyMpAOeF3YKYZyAVd
   M0WY4GErclVuAC
Have you recorded the key share? (y/n)
```

It is important to record the whole string presented, including the prefix (in this case) `2-1-` which indicates the number of shares determined to be required to rejoin (or the privacy threshold) and the number of the share itself out of the total number of shares being created.

Tip: For non-production purposes, such as in a lab scenario, it is not necessary to specify that the wrap key should be shared between key custodians but instead just use one solitary key. To do this, when configuring the device using YubiHSM Setup, indicate the number of shares to be 1 and the privacy threshold to also be 1.

Step 8 The setup configuration continues by creating an application authentication key. This key is used to authenticate to the device by the Key Storage Provider (KSP) in Windows, allowing the KSP to perform operations in YubiHSM 2. Since object ID `1` and `2` are already in use by the default authentication key and the wrap key respectively, the example in this guide assumes that the application authentication key to be created next gets ID `3`. Do one of the following:

- To manually assign an application authentication key ID, type `3` and press **Enter**.
- To instead allow the system to assign a wrap key ID automatically, type `0` and press **Enter**.

You also need to choose a password for the application authentication key. Be sure to store the password of the application authentication key that you will use in a way so that it cannot be compromised. You will need this information later to configure the KSP for use with ADCS. Enter the application authentication key password and press Enter. A confirmation message appears.

```
Stored application authentication key with ID 0x0003 on the device
Saved wrapped application authentication key to {path} 0x0003.yhw
```

The wrapped application authentication key (`0x0003.yhw`) has been saved to the same path as the location of the YubiHSM Setup program. Although encrypted using the wrap key, we recommend that you do not store keys under wrap on a network-accessible or otherwise potentially comprisable

storage media. Leave the file where it was saved for now, as it will be used later to create a backup. You can remove the application authentication key afterwards.

Step 9 The final step of the YubiHSM 2 setup process is to decide whether to create an audit key. The audit key is used to access the internal audit log of the device which holds information about the last 62 operations performed. It is also used to reset the log if needed. Depending on your local requirements, you may not need to create an audit key. If you are unsure of your requirements, we suggest you create an audit key.

When prompted to create an audit key, type y. You are then prompted to assign a key ID to the audit key. Be sure to make a note of the ID you enter (for example, key ID 4). You are also prompted to enter the audit key password. Be sure to store this password as well, so that it cannot be compromised. Finally, the audit key will be exported under wrap to the current working directory. Using our example of key ID 4, the file will be named `0x0004.yhw`.

Step 10 The setup tool finishes by letting you know that the default, factory-installed authentication key has been deleted.

```
Previous authentication key 0x0001 deleted
All done
```

Finally, the YubiHSM Setup application exits.

5.3 Verifying the Setup

You can verify the results of the YubiHSM Setup program by using the YubiHSM Shell program, and logging in using the application authentication key (we used object ID 3 in this guide). To verify the YubiHSM Setup:

Step 1 In your command line application (where $ is the prompt), run YubiHSM Shell program. To do this, if you haven't already, launch your command line application and navigate to the directory containing the YubiHSM Shell program. Then run the following command and press **Enter**. `$ yubihsm-shell`

Step 2 To connect to the YubiHSM, at the yubihsm prompt, type `connect` and press **Enter**. A message verifying that you have a successful connection is displayed.

Step 3 To open a session with the YubiHSM 2, type `session open 3` and press **Enter**.

Step 4 Type in the password for the application authentication key. You will receive a confirmation message that the session has been set up successfully.

Step 5 You now have an administrative connection to the YubiHSM 2 and can list the objects available. To list the objects, type `list objects 0` and press **Enter**. Your results should be similar to the following:

```
Found 3 object(s)
id: 0x0002, type: wrapkey, sequence: 0
id: 0x0003, type: authkey, sequence: 0
id: 0x0004, type: authkey, sequence: 0
```

As you can see by looking at their IDs, these objects correspond to the wrap key, the application authentication key and the audit key that were just created.

To obtain more information about any one of the objects, for example, the application authentication key (object ID 3), including its capabilities, type the following command and press **Enter**:

```
yubihsm> get objectinfo 0 3 authentication-key
```

The response you receive should look similar to the following:

```
id: 0x0003, type: authkey, algorithm: yubico-aes-auth,
   label: "Application auth key", length: 40, domains: 1,
   sequence: 0, origin: imported, capabilities:
   asymmetric_gen:asymmetric_sign_pkcs:asymmetric_sign_pss:
   export_wrapped: import_wrapped:export_under_wrap,
   delegated_capabilities:
   asymmetric_gen:asymmetric_sign_pkcs:asymmetric_sign_pss:
   export_under_wrap
```

This indicates that YubiHSM 2 as it has now been configured will later on allow the KSP to leverage the device to:

- Generate asymmetric objects
- Compute signatures using RSA-PKCS1v1.5
- Compute signatures using RSA-PSS
- Export other objects under wrap
- Import wrapped objects
- Mark an object as exportable under wrap

In addition, this object (the application authentication key, object ID 3) also has so-called delegated capabilities. Delegated capabilities define the set of capabilities that can be set or "bestowed" onto other objects that are created by it.

Step 6 To exit, type `quit`.

CHAPTER

SIX

DEPLOYING YUBIHSM 2 WITH ACTIVE DIRECTORY CERTIFICATE SERVICES

With a YubiHSM 2 device now configured for use with YubiHSM Key Storage Provider and Microsoft Active Directory Certificate Services, the next set of steps covers the deployment in the ADCS environment. Note that YubiHSM Key Storage Provider software must be installed on the system before proceeding.

Deploying YubiHSM consists of three steps as follows. These steps are described in detail in the following procedure.

1. Configuring the Windows Registry for the YubiHSM Key Storage Provider for the primary YubiHSM 2 device that was configured earlier
2. Configuring ADCS (if not already present)
3. Configuring a new ADCS CA with a root CA key being generated on the device

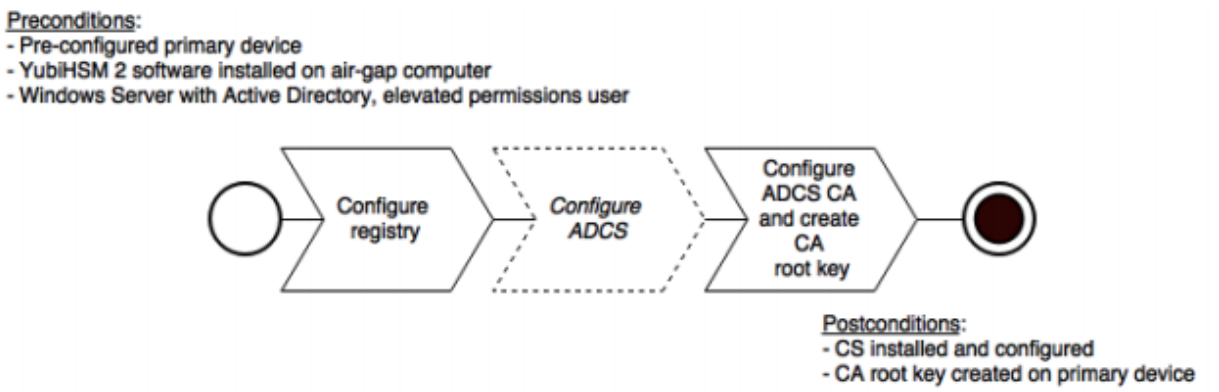

Figure: Pre and Post Conditions

The host that these steps are performed on is assumed to be a member server in the Active Directory domain (domain-joined, not a Domain Controller).

These instructions include steps for a basic configuration and should be performed by an experienced system administrator.

6.1 Configuring the Windows Registry

For ADCS to use the YubiHSM 2, the following registry entries need to be changed from their default values. The `HKEY_LOCAL_MACHINE\SOFTWARE\Yubico\YubiHSM` subkey was created during installation. Be sure to make a backup of your Registry before you make any changes. To configure the Windows Registry

Step 1 Click **Start > Run**, type `regedit` in the Run dialog box, and click **OK**.

Step 2 Locate and then click the registry subkey for YubiHSM (`HKEY_LOCAL_MACHINE\SOFTWARE\Yubico\YubiHSM`).

Step 3 To change the URI where the connector is listening, change the following entry: `"ConnectorURL"=http://127.0.0.1:12345`

Step 4 To change the ID of the application authentication key (object ID `3` was used as an example in this guide; if you used another object ID be sure to enter the correct information). For our example, because the hexadecimal value of `0x00000003` resolves to `3` in the Windows Registry, change the entry as follows: `"AuthKeysetID"=3`

Step 5 To change the password for the application authentication key that is stored in the registry change the entry for: `"AuthKeysetPassword"={password}` The password is stored here for the Key Storage Provider to use when authenticating to the device.

Step 6 To save your changes, exit the Windows Registry.

The YubiHSM Connector service reads the configuration file, `yubihsm-connector-config.yaml`.

Depending on your local setup, for instance if you are running multiple instances of the software on the same host, you may need to edit this configuration file to make sure that parameters are consistent between the configuration file and the Windows Registry. On Windows, the `yubihsmconnector.config.yaml` file is available at `C:\programdata\yubiHSM\yubihsmconnector.yaml` - you will need administrator rights to modify the file.

6.2 Setting Up Your Enterprise Certificate Authority

6.2.1 To Configure ADCS

If you already have Certification Services installed, you can skip these steps.

Step 1 On a Windows Server host, joined to an existing Active Directory domain, log on into the server as a domain administrator.

Step 2 Click **Start > Administrative Tools**, then click **Server Manager**.

Step 3 Under Roles Summary, click **Add roles and features**.

Step 4 Use the Add Roles and Features Wizard to add the Active Directory Certificate Services role, and click **Next**

Step 5 In the Select role services wizard page, select the option for **Certification Authority**, then click **Next**.

Step 6 Complete the wizard and reboot the host if prompted.

6.2.2 To Configure the ADCS CA and Create the Root Key

After you have completed the feature installation, you need to create the Enterprise CA instance.

Step 1 If you haven't already, do the following:

- On a Windows Server host, joined to an existing Active Directory domain, log into the server as a `domain administrator`.
- Click **Start > Administrative Tools**, then click **Server Manager**.

Step 2 In Server Manager, start the **Add Roles and Features Wizard** and select **Role-based** or **feature based** installation. Click **Next**.

Step 3 In the Credentials page, confirm that you are logged in as a `domain administrator`. If you are not, you will not be able to create an Enterprise CA in the subsequent steps. Click **Next**.

Step 4 In the Role Services page, select the option for **Certification Authority**, and then click **Next**.

Step 5 In the Setup Type page, select the option for **Enterprise CA**, and then click **Next**.

Step 6 In the CA Type page, select the option for **Root CA**, and then click **Next**.

Step 7 In the Private Key page, select the option for **Create a new private key**, and then click **Next**.

Step 8 In the Cryptography for CA page, do the following:

a. Click **Select a cryptographic provider** and select **RSA#YubiHSM Key Storage Provider** from the list displayed. This indicates that the root key should be generated on the device.

b. Click **Key Length** and select the key size you want from the list displayed. Options for key size `2048-bit` or `4096-bit`. The default setting is `2048`.

c. For Select the hash algorithm for signing certificates issued by this CA, select a desired hash algorithm, such as `SHA256`. The default setting is `SHA256`.

d. Select the option to **Allow administrator interaction when the private key is accessed by the CA**. This allows the private key to be exported for backup purposes (so it can be restored to another server). Click **Next**.

Step 9 In the CA Name page, accept the defaults. Click **Next**.

Step 10 In the Validity Period page, accept the default or set another validity period appropriate for your purposes. Click **Next**.

Step 11 In the CA Database page, accept the default location for logs. Click **Next**.

Step 12 In the Confirmation page, the important detail is that the `YubiHSM Key Storage Provider` is being used to store the CA private key. Click **Configure**.

The Progress page appears, briefly, as the local CA database is created, and changes are written to Active Directory.

Step 13 Finally, confirm the presence of the `Configuration succeeded` message in the Results page. Click **Close**.

CHAPTER

SEVEN

ALTERNATIVE SCENARIOS

This guide covers only basic setup and use of the YubiHSM 2 with ADCS. Some alternative scenarios include migrating an existing CA root key to YubiHSM 2, or leveraging the YubiHSM 2 and YubiHSM Key Storage Provider in larger PKI installations using multiple hosts to serve the CA including Subordinate CAs. Since conditions can vary a great deal between organizations on these topics, the following contains some references that might be useful when deploying YubiHSM 2 under such circumstances.

7.1 Migrating an Existing CA Root Key to YubiHSM 2

One potential circumstance when deploying YubiHSM 2 to secure ADCS is the fact that a CA root key already exists, either in software or secured by hardware such as another Hardware Security Module. It is normally possible to migrate the CA root key over to the YubiHSM 2, however depending on the pre-existing setup, the steps to take may vary. For more information, see the information on the Yubico developers' website at Move Software Keys to Key Storage Provider.

7.2 Subordinate CAs

In order to improve security and scalability of your Certification Authority, consider installing the Root CA on a standalone (offline) server, and use a Subordinate CA for all certificate signing. For additional information about implementing advanced configurations, see the relevant Microsoft documentation on the subject (such as AD CS Step by Step Guide: Two Tier PKI Hierarchy Deployment).

7.3 Alternative Backup and Restore Procedures

In more advanced installations, such as when the YubiHSM Setup program was not used to set up YubiHSM 2 for ADCS, or when moving the YubiHSM 2 device containing the root CA key from one instance of ADCS to another, see the information on the Yubico developers' website at Backup and Restore.

CHAPTER

EIGHT

BACKING UP KEY MATERIAL

We strongly recommend you make a backup copy of all production objects residing on your primary device, particularly once the CA root key has been generated on the device. If there is an unforeseen hardware failure of the primary device, having a backup ensures that you can resume operations quickly. In addition, this provides a means to back up all objects contained on a device to reside in secure hardware offline. Specific recommendations for governance of your critical key material is out of scope for this guide. Ensure that you design and document these security procedures to fit the requirements of your organization.

8.1 Backing Up the YubiHSM 2

The backup of the primary YubiHSM 2 is a duplicate of all of the objects stored on the primary device, to be exported under wrap and that are available using the application authentication key used. For instance, when following this guide, the wrap key (created with ID 2 previously), the application authentication key (ID 3), the audit key (ID 4) (if created previously), and the CA root key will be duplicated onto the secondary device. The factory-installed authentication key (ID 1) on the secondary device will be destroyed. You will need assistance from the wrap key custodians to provide their respective wrap key shares, if applicable. In the example we used in this guide, 2 out of the 3 shares must be available. When you create a backup, you create a duplicate of the objects on your primary YubiHSM 2 onto a secondary device. The actual backup procedure consists of steps as follows. These steps are described in detail in the following procedure.

1. Set up communication between the YubiHSM 2 tools and the device.
2. Start the configuration process, and authenticate to the YubiHSM 2 device.
3. Identify the CA root key ID.
4. Export the CA root key.
5. Verify the key material under wrap.
6. Restore the key material onto a secondary (backup) device.
7. Verify the objects on the secondary device are correct.

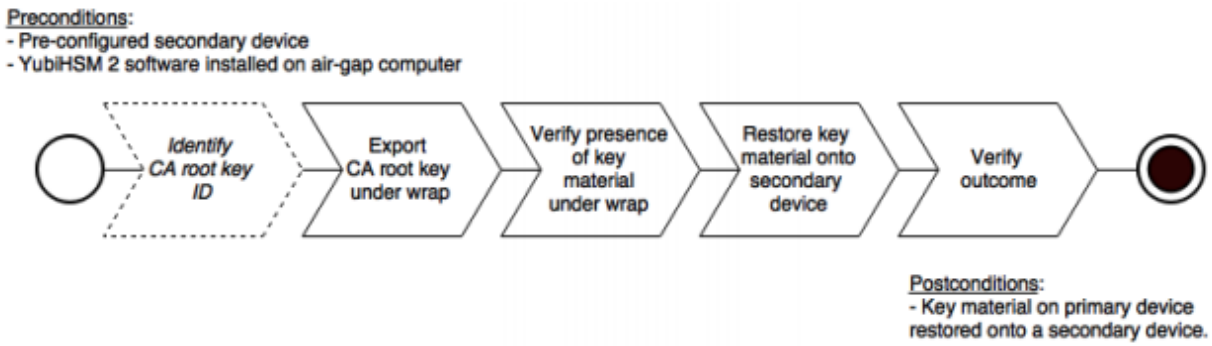

Figure: Pre and Post Conditions

Since the CA root key was created on the device when setting up the CA, it currently only exists on the device. In order to back it up using the YubiHSM Setup program, it must first be exported from the device using the wrap key that also sits on the device alongside the application authentication key and the audit key. To export the CA root key under wrap using the wrap key on the device:

Step 1 In your command line application, run YubiHSM Shell program. To do this, if you haven't already:

a. Launch your command line application and navigate to the directory containing the YubiHSM Shell program.

b. Then run the following command and press **Enter**.

```
$ yubihsm-shell
```

Step 2 To connect to the YubiHSM, at the `yubihsm` prompt, type `connect` and press **Enter**. A message verifying that you have a successful connection is displayed.

Step 3 To open a session with the YubiHSM 2, type `session open 3` and press **Enter**.

Step 4 Type in the password for the application authentication key.

You will receive a confirmation message that the session has been set up successfully.

Step 5 If you already know the object ID of the root CA, you can skip this step. If you need to identify the root CA, you can list the objects available.

a. To list the objects, type `list objects 0` (where `0` is the session number) and press **Enter**.

b. You will receive a list of the objects on the device that application authentication key with ID `3` has access to, which will include the CA root key. Identify its ID.

Step 6 To export the CA root key under wrap from the primary device to the local file system, in the YubiHSM Shell program, run

```
yubihsm> get wrapped 0 2 asymmetric {rootkeyID} rootkey.yhw
```

Step 7 Verify that all the keys that were exported under wrap to file reside in the same directory as the YubiHSM Setup program. The tool looks for files with the `.yhw` file extension in the current working directory and attempts to read and import them into the device. The wrap key will be imported as a result of providing the wrap key shares to the tool. Given the example object IDs in this guide, the following files should be present:

- `0x0003.yhw` (Application authentication key under wrap)
- `0x0004.yhw` (Audit key under wrap)
- `rootkey.yhw` (CA root key under wrap)

Step 8 To begin the process of restoring the data onto the secondary YubiHSM 2, if the primary YubiHSM 2 device is inserted into your computer, remove it and insert the secondary device. Restoring a device must be performed in an air-gapped environment in order to guarantee integrity.

Step 9 In your command line application (where $ is the prompt), run YubiHSM Setup with the argument `restore`.

a. To do this, launch your command line application, navigate to the directory containing the YubiHSM Setup program,

b. Run the following command, and press **Enter**.

```
$ yubihsm-setup restore
```

Step 10 To start the YubiHSM Setup process, type the default authentication key password: `password` and press **Enter**.

A confirmation message is displayed that the default authentication key was used and that you successfully have authenticated to the device: `Using authentication key 0x0001`.

You will now start the restore procedure, which involves providing the number of wrap keyshares required by the privacy threshold defined when setting up the primary device.

Step 11 When prompted, type the number of shares required by the privacy threshold and press **Enter**.

In this guide, we have specified that 2 shares are required to be rejoined. These must be present in order to proceed.

Step 12 When prompted, for share number 1, the wrap key custodian holding the first share inputs this information and presses **Enter**. A message is displayed that the share is received:

```
Received share 2-1␣
↪WWmTQj5PHGJQ4H9Y2ouURm8m75QkDOeYzFzOX1VyMpAOeF3YKYZyAVdM0W
    Y4GErclVuAC
```

Step 13 Continue to have each wrap key custodian enter the share information for each of the wrap key shares required to rejoin the key share. Once the sufficient number of wrap key shares have been inserted by the wrap key custodians, a final message is displayed:

```
Stored wrap key with ID 0x0002 on the device
```

Note: The ID of the wrap key on the secondary device is the same as that for the primary device.

After the wrap key has been stored on the secondary device, the YubiHSM Setup program reads the files containing the application authentication key, the CA root key, and, if applicable, the audit key that were saved to file under wrap during the configuration of the primary device.

```
reading ./0x0004.yhw
Successfully imported object Authkey, with ID 0x0004
reading ./0x0003.yhw
Successfully imported object Authkey, with ID 0x0003
reading ./rootkey.yhw
Successfully imported object Asymmetric, with ID {rootkeyID}
```

If there are files containing wrapped objects with the `.yhw` file extension in this directory that were exported with a different wrap key than the one reconstituted by the shares here, the setup tool attempts to also read those but will fail gracefully and only restores the files it can decrypt.

The restore process finishes and the setup tool lets you know that the default, factory-installed authentication key has been deleted.

```
Previous authentication key 0x0001 deleted
All done
```

Finally, the YubiHSM Setup application exits.

8.2 Confirming the Duplicated YubiHSM 2

You now have a duplicate of the device configured with the three key objects you created on the primary device earlier. These are identical to the primary device that was configured earlier.

8.2.1 To Confirm the Duplicated YubiHSM 2

Step 1 In your command line application, run YubiHSM Shell program. To do this, if you haven't already:

a. Launch your command line application and navigate to the directory containing the YubiHSM Shell program.

b. Then run the following command and press **Enter**.

```
$ yubihsm-shell
```

Step 2 To connect to the YubiHSM, at the `yubihsm` prompt, type `connect` and press **Enter**. A message verifying that you have a successful connection is displayed.

Step 3 To open a session with the YubiHSM 2, type `session open 3` (where `3` is the ID for your application authentication key) and press **Enter**.

Step 4 Type in the password for the application authentication key. You will receive a confirmation message that the session has been set up successfully.

Step 5 To list the objects, type `list objects 0` (or instead of `0` some other session number that was given to you in step 4) and press **Enter**. Verify that the secondary device now contains all of the key material that you intended to restore.

Depending on the order in which the keys under wrap were imported, the order of the enumerated keys on the secondary device may be different than on the primary device when using the list command. This has no practical implication and the object IDs are identical between the devices.

Step 6 If you have verified that the secondary device now contains all of the key material that you intended to restore, you should now remove the keys under wrap currently on file in the current working directory for the YubiHSM Setup program.

CHAPTER

NINE

GETTING HELP

Should you require assistance when deploying YubiHSM 2 with ADCS by using this guide, start by referencing the product documentation and currently known issues. If you need additional help, contact Yubico:

- Yubico Knowledge Base
- Product Solution Brief
- YubiHSM 2 Product Overview
- Submit a support request to report an issue

CHAPTER

TEN

TERMINOLOGY USED

Default authentication key Factory-installed AES key used when initializing the device. Possesses all capabilities.

Application authentication key AES key used to authenticate to device. Performs operations according to its defined capabilities.

Audit key AES authentication key with rights to access audit log.

Wrap key AES key used to protect key material when exporting to file from device and when importing from file to device. Key material exported under wrap will be encrypted and can only be decrypted using the wrap key.

Capability A description of what operations are allowed on or with an object such as a key.

Delegated capability A description of what operations are allowed on or with an object delegated by the authentication key or wrap key that was used to create it.

Domain A logical "container" for objects that can be used to control access to objects on the device.

Object ID OIDs are unique identifiers for any kind of object stored on YubiHSM 2. IDs can range from 1 to 65535; however, the device can hold no more than 256 unique objects.

M of n Scheme where Wrap key is split into n shares held by key custodians, where at least m shares are needed to use the key (sometimes this is also called 'quorum').

Key custodian Holder of a wrap key share.

CHAPTER

ELEVEN

COPYRIGHT

Trademarks

Yubico and YubiKey are registered trademarks of Yubico AB. All other trademarks are the property of their respective owners.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Yubico shall have no liability for any error or damages of any kind resulting from the use of this document.

The Yubico Software referenced in this document is licensed to you under the terms and conditions accompanying the software or as otherwise agreed between you or the company that you are representing.

Contact Information

Yubico Inc.
530 Lytton Street
Suite 301
Palo Alto, CA 94301
USA

Click the links to:

- Submit a support request
- Send a Contact Me request
- See additional contact options for getting touch with us

Document Updated

2022-05-12 22:57:40 UTC