

---

# Deploying YubiHSM 2 for Microsoft Host Guardian Service

**Yubico**

**May 11, 2022**



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	The Host Guardian Service – Guarded Fabric Concept . . . . .	1
1.2	HGS Key Protection Service . . . . .	2
1.3	Scope of this Document . . . . .	3
<b>2</b>	<b>Prerequisites and Preparations</b>	<b>5</b>
<b>3</b>	<b>Basic Setup of YubiHSM 2 and Host Guardian Service</b>	<b>7</b>
3.1	Install and Configuring YubiHSM 2 . . . . .	7
3.2	Basic Deployment of HGS . . . . .	8
<b>4</b>	<b>Create Signing and Encryption Keys for HGS</b>	<b>9</b>
4.1	Generate Signing and Encryption Keys and Certificates . . . . .	9
4.2	Initialize HGS with Signing and Encryption Keys and Certificates . . . . .	11
<b>5</b>	<b>Back Up Key Material</b>	<b>13</b>
<b>6</b>	<b>Getting Help and Further Reading</b>	<b>15</b>
<b>7</b>	<b>Terminology</b>	<b>17</b>
<b>8</b>	<b>Copyright</b>	<b>19</b>



## INTRODUCTION

In a Microsoft Host Guardian Service (HGS) environment, the signing key and the encryption key must be protected in hardware. The YubiHSM 2 protects these keys in hardware and thereby guards the HGS.

This guide is intended to help systems administrators deploy YubiHSM 2 for use with HGS in a Windows server environment. The expected outcome is that the signing key and the encryption key are generated and stored securely on a YubiHSM 2 and that a hardware-based backup copy of key materials has been produced.

These guidelines for deployment cover basic topics, so the instructions should be modified as required for your particular environment. It is assumed that you are familiar with the concepts and processes for working with HGS. It is also assumed that the installation is performed on a single HSG, but the concept can be extended to multiple servers.

---

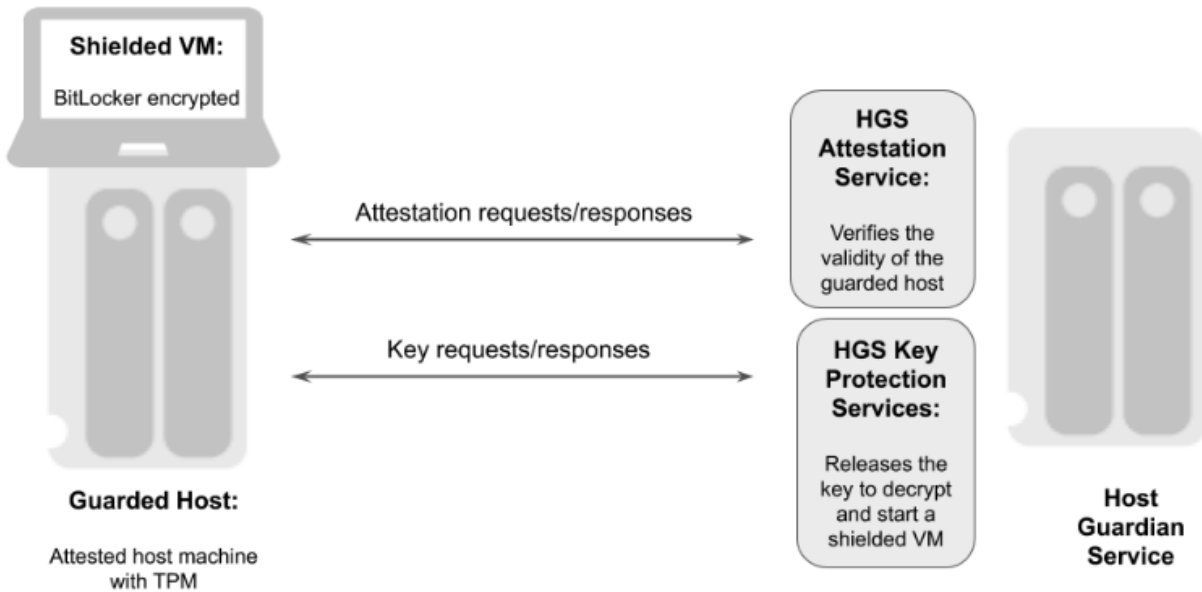
**Important:** We recommend that you install and test the HGS installation and setup of the YubiHSM 2 in a test or lab environment before deploying to production. For guidance on enabling the HGS in a production environment, see [Microsoft's documentation on how to deploy a guarded fabric and shielded virtual machines \(VMs\)](#).

---

### 1.1 The Host Guardian Service – Guarded Fabric Concept

In order to raise the security level for virtualization, Microsoft Windows Server 2016 introduced the concept of [Guarded Fabric](#) to increase the security of Hyper-V Virtual Machines (VMs). A guarded fabric is used to protect hosts from a VM running malicious software and to protect VMs from a compromised host.

The graphic below gives an overview of a guarded fabric and the main components.



**Figure - Overview of a guarded fabric**

A guarded fabric is comprised of the following main components:

- **Host Guardian Service (HGS):** This is a Windows Server role that is typically installed on a cluster of physical servers. The HGS in turn is composed of the Attestation Service and the Key Protection Service. The Attestation Service verifies the Trusted Computing Group (TCG) logs of a guarded host, and issues a health certificate if the Guarded Host is attested by HGS. The HGS Key Protection Service is described in “HGS Key Protection Service” below.
- **Guarded Host:** This is an attested host machine, equipped with a Trusted Platform Module (TPM) that can run shielded Hyper-V VMs. The guarded Hyper-V host must be attested by the HGS Attestation Service in order to power on or migrate shielded VMs.
- **Shielded VM:** This is a Hyper-V VM equipped with a virtual TPM, that is encrypted using BitLocker and can run only on attested guarded hosts in a guarded fabric.

The guarded fabric components are described in [Microsoft’s overview of guarded fabric and shielded VMs](#).

## 1.2 HGS Key Protection Service

The HGS Key Protection Service (KPS) is configured with at least two certificates (and corresponding private keys), which are used for signing and encrypting the keys used to start up shielded VMs. The two mandatory certificates are:

- **Encryption certificate:** This certificate is used to encrypt and decrypt the key protector, which itself contains the symmetric key that encrypts the virtual TPM of a shielded VM at rest. When a shielded VM is booting up on an attested guarded host, the HGS KPS decrypts and releases its symmetric key, which is used by the guarded host to decrypt the virtual TPM and the hard drive of a shielded VM.
- **Signing certificate:** This certificate is used to digitally sign the key protector to ensure its authenticity.

In addition to these mandatory certificates, the HGS KPS can also be configured with four optional certificates:

- Communications certificate
- Attestation signer certificate

- HTTPS (SSL/TLS) certificate
- Dump encryption certificate.

If those certificates are not configured, the Encryption certificate and Signing certificate will provide the necessary operations.

The Encryption certificate and Signing certificate can either be self-signed or issued by a Certification Authority (CA).

The private keys corresponding to the certificates can be stored in an HSM or in software in PKCS #12 format. The recommended option is to protect the keys in hardware in an HSM.

For more information on these topics, see [Frequently Asked Questions About HGS Certificates](#) in the Microsoft Tech Community.

### 1.3 Scope of this Document

The scope of this document is to describe how to use the HGS KPS to generate the Encryption and Signing certificates/keys using the YubiHSM. In this document, the Encryption and Signing certificates will be self-signed and created with PowerShell scripts.

How to use CA to issue the certificates is out of scope for this document.

How to deploy and configure the HGS Attestation Service, guarded hosts, shielded VMs, and additional features of a guarded fabric are also out of scope for this document.

For information on how to install and configure a complete guarded fabric, see Microsoft's documentation on guarded fabric deployment.





## PREREQUISITES AND PREPARATIONS

The audience of this document is an experienced systems administrator with a good understanding of Microsoft Hyper-V virtualization management. In addition, it is helpful to be familiar with the terminology, software and tools specific to YubiHSM 2. As a primer for these, refer to the *Terminology* section in this guide.

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

- Microsoft Windows Server 2016 or higher. The operating system should be installed in a secure computer network. The system administrator must also have elevated system privileges.
- YubiHSM 2 software and tools for Windows downloaded from the [Yubico YubiHSM 2 Release page](#) and available on the system to be used.
- Two (2) YubiHSM 2 devices, one for deployment and one for backup in hardware.
- Your organization's policies may require key custodians to be available for the YubiHSM 2 deployment. For more information about key custodians and the associated M of N key shares, see “[Key Splitting and Key Custodians](#)” in the YubiHSM 2 Windows Deployment Guide.

Configuration for this Integration For the integration described in this guide, the following hardware and software configuration was used:

- Microsoft Windows Server 2016.
- Yubico YubiHSM v 2.1.2.
- Yubico YubiHSM v 2.1.2 software tools.



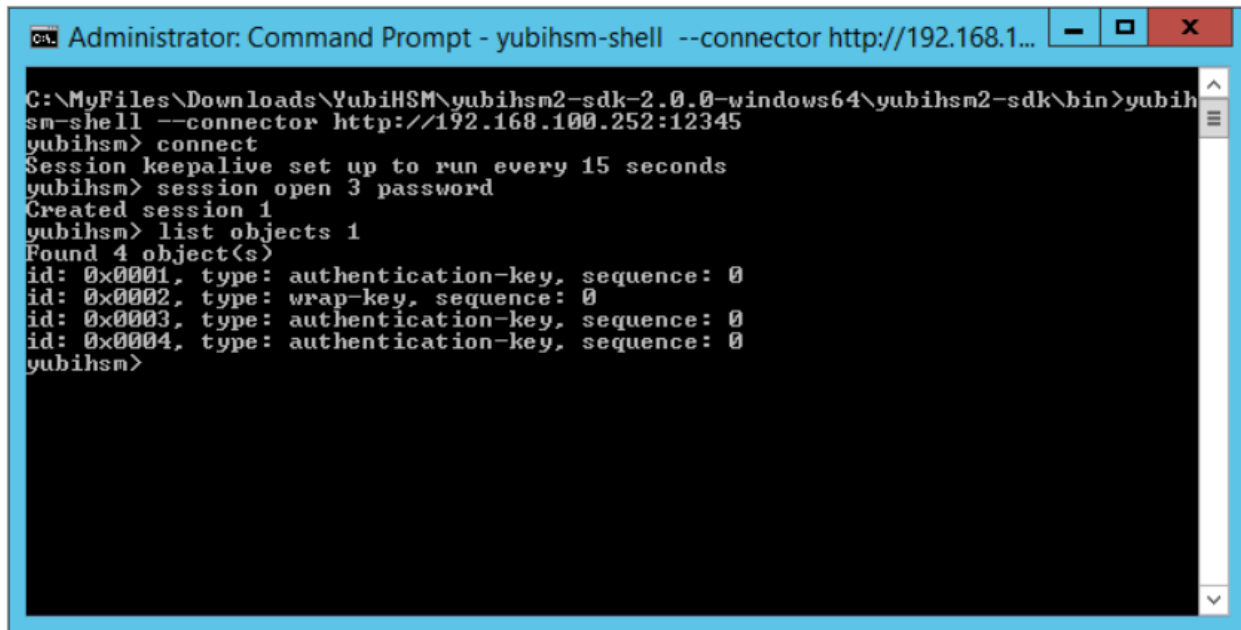
## BASIC SETUP OF YUBIHSM 2 AND HOST GUARDIAN SERVICE

### 3.1 Install and Configuring YubiHSM 2

Install and configure the YubiHSM 2 and software using the instructions in the following sections in the [YubiHSM 2 Windows Deployment Guide](#):

1. Install YubiHSM 2 Tools and Software
2. Configure the Primary YubiHSM 2 Device
3. Configure the YubiHSM 2 Software

Once these instructions have been followed, the YubiHSM 2 should be configured with the example we are using, one domain with a wrap key (id 0x0002), an application authentication key (id 0x0003), and an audit key (id 0x0004). The configuration of the YubiHSM 2 can be inspected by using the YubiHSM-Shell in a command prompt as shown in the screenshot below.



```
Administrator: Command Prompt - yubihsm-shell --connector http://192.168.1...
C:\MyFiles\Downloads\YubiHSM\yubihsm2-sdk-2.0.0-windows64\yubihsm2-sdk\bin>yubih
sm-shell --connector http://192.168.100.252:12345
yubihsm> connect
Session keepalive set up to run every 15 seconds
yubihsm> session open 3 password
Created session 1
yubihsm> list objects 1
Found 4 object(s)
id: 0x0001, type: authentication-key, sequence: 0
id: 0x0002, type: wrap-key, sequence: 0
id: 0x0003, type: authentication-key, sequence: 0
id: 0x0004, type: authentication-key, sequence: 0
yubihsm>
```

Figure - Example of the YubiHSM 2 basic configuration

### 3.2 Basic Deployment of HGS

To test the encryption and signing certificate/key generation for HGS Key Protection Services, configure a basic HGS environment on a single server. For more information on how to install and configure a complete guarded fabric, see [Microsoft's documentation on guarded fabric deployment](#).

To use shielded VMs, begin by adding the HGS role and configuring the HGS domain. In the following, we are showing the PowerShell prompt as PS C:\users\your-username\.

#### Step 1 Add HGS Role.

To add the HGS role to a Windows Server, open a PowerShell console and enter the following command:

```
PS C:\users\your-username\ Install-WindowsFeature  
-Name HostGuardianServiceRole -IncludeManagementTools -Restart
```

For more information on this PowerShell command, see [Microsoft's documentation on how to Install HGS](#).

#### Step 2 Install Host Guardian Server on Bastion Host.

To configure the Active Directory (AD) forest for HGS, configure the HGS service, and lock down the Windows Server to a bastion host, open a PowerShell console and enter the following command:

```
PS C:\users\your-username\ $adminPassword = ConvertTo-SecureString  
-AsPlainText '<password>' -Force  
  
PS C:\users\your-username\ Install-HgsServer -HgsDomainName  
'bastion.local' -SafeModeAdministratorPassword $adminPassword  
-Restart
```

For more information on this PowerShell command, see [Microsoft's documentation on how to Install HGS](#).

## CREATE SIGNING AND ENCRYPTION KEYS FOR HGS

### 4.1 Generate Signing and Encryption Keys and Certificates

Generate the signing and encryption keys and certificates for HGS by using the PowerShell cmdlet `New-SelfSignedCertificate`. In this guide, self-signed certificates will be used for HGS.

The HGS signing and encryption certificates must adhere to the following specifications:

- Crypto provider: YubiHSM Key Storage Provider.
- Key algorithm: RSA
- Minimum key size: 2048 bits
- Signature algorithm: SHA256
- Key usage: Digital signature and data encipherment
- Enhanced key usage: Server authentication
- Subject name: Recommended: your company's name or web address

Do the following to create the self-signed HGS certificates:

**Step 1** Create the Self-signed HGS Signing Certificate and Key.

Start a command prompt with administrator rights and type the command `PowerShell`. In the PowerShell command prompt, run the following cmdlet:

```
PS New-SelfSignedCertificate -Provider "YubiHSM Key Storage  
Provider" -Subject "CN=HGS Signing Certificate" -KeyExportPolicy  
NonExportable -KeyUsage DigitalSignature,DataEncipherment  
-TextExtension @"2.5.29.37={text}1.3.6.1.5.5.7.3.1"  
-KeyAlgorithm RSA -KeyLength 2048 -CertStoreLocation  
"Cert:\LocalMachine\My" -Verbose
```

**Step 2** Create the Self-signed HGS Encryption Certificate and Key.

In the PowerShell command prompt, run the following cmdlet:

```
PS C:\users\your-username\ New-SelfSignedCertificate -Provider  
"YubiHSM Key Storage Provider" -Subject "CN=HGS Encryption  
Certificate" -KeyExportPolicy NonExportable -KeyUsage  
DigitalSignature,DataEncipherment -TextExtension  
@"2.5.29.37={text}1.3.6.1.5.5.7.3.1" -KeyAlgorithm RSA  
-KeyLength 2048 "Cert:\LocalMachine\My" -Verbose
```

```

Administrator: Command Prompt - powershell
PS C:\Users\Administrator> New-SelfSignedCertificate -Provider "YubiHSM Key Storage Provider" -Subject "CN=HGS Signing Certificate" -KeyExportPolicy NonExportable -KeyUsage DigitalSignature,DataEncipherment -TextExtension @(("2.5.29.37-{text}1.3.6.1.5.5.7.3.1") -KeyAlgorithm RSA -KeyLength 2048 -CertStoreLocation "Cert:\LocalMachine\My" -Verbose
VERBOSE: Performing the operation "Create a new self-signed certificate" on target "Cert:\LocalMachine\My".

PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\My

Thumbprint                Subject
-----
A576F936B6F044586123FDE8CB3C7BDDA1431DA8  CN=HGS Signing Certificate

PS C:\Users\Administrator> New-SelfSignedCertificate -Provider "YubiHSM Key Storage Provider" -Subject "CN=HGS Encryption Certificate" -KeyExportPolicy NonExportable -KeyUsage DigitalSignature,DataEncipherment -TextExtension @(("2.5.29.37-{text}1.3.6.1.5.5.7.3.1") -KeyAlgorithm RSA -KeyLength 2048 -CertStoreLocation "Cert:\LocalMachine\My" -Verbose
VERBOSE: Performing the operation "Create a new self-signed certificate" on target "Cert:\LocalMachine\My".

PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\My

Thumbprint                Subject
-----
5701A22B99C029FCFB578B9191AEFA8AF7454188  CN=HGS Encryption Certificate

PS C:\Users\Administrator>
    
```

**Figure – Example of PowerShell cmdlet to create self-signed certificates**

Make a note of the thumbprints of the self-signed certificates. In this example, the signing certificate thumbprint is A576F936B6F044586123FDE8CB3C7BDDA1431DA8 and the encryption certificate thumbprint is 5701A22B99C029FCFB578B9191AEFA8AF7454188.

### Step 3 Verify Generation and Storage of HGS Key-pairs in YubiHSM 2.

Verify that the HGS key-pairs have been properly generated and stored in YubiHSM 2 by starting a command prompt and using YubiHSM-Shell to list the objects, as shown in the figure below.

```

Administrator: Command Prompt - yubihsm-shell --connector http://192.168.100.252:12345
C:\MyFiles\Downloads\YubiHSM\yubihsm2-sdk-2019-03-win64-amd64\yubihsm2-sdk\bin>yubihsm-shell --connector http://192.168.100.252:12345
yubihsm> connect
Session keepalive set up to run every 15 seconds
yubihsm> session open 1 password
Created session 1
yubihsm> list objects 1
Found 6 object(s)
id: 0x0001, type: authentication-key, sequence: 0
id: 0x0002, type: wrap-key, sequence: 0
id: 0x0003, type: authentication-key, sequence: 0
id: 0x0004, type: authentication-key, sequence: 0
id: 0xc68d, type: asymmetric-key, sequence: 0
id: 0xd664, type: asymmetric-key, sequence: 0
yubihsm> get objectinfo 1 0xc68d asymmetric-key
id: 0xc68d, type: asymmetric-key, algorithm: rsa2048, label: "te-bb34e59b-59d5-49d3-85a6-5e2bace8908b", length: 896, domains: 1, sequence: 0, origin: generated, capabilities: decrypt-oaep:decrypt-pkcs:exportable-under-wrap:sign-pkcs:sign-ps
yubihsm> get objectinfo 1 0xd664 asymmetric-key
id: 0xd664, type: asymmetric-key, algorithm: rsa2048, label: "te-3a0aaefc-08fd-4efb-a465-78eab3a0a935", length: 896, domains: 1, sequence: 0, origin: generated, capabilities: decrypt-oaep:decrypt-pkcs:exportable-under-wrap:sign-pkcs:sign-ps
yubihsm>
    
```

**Figure – Example of HGS keys in YubiHSM-Shell**

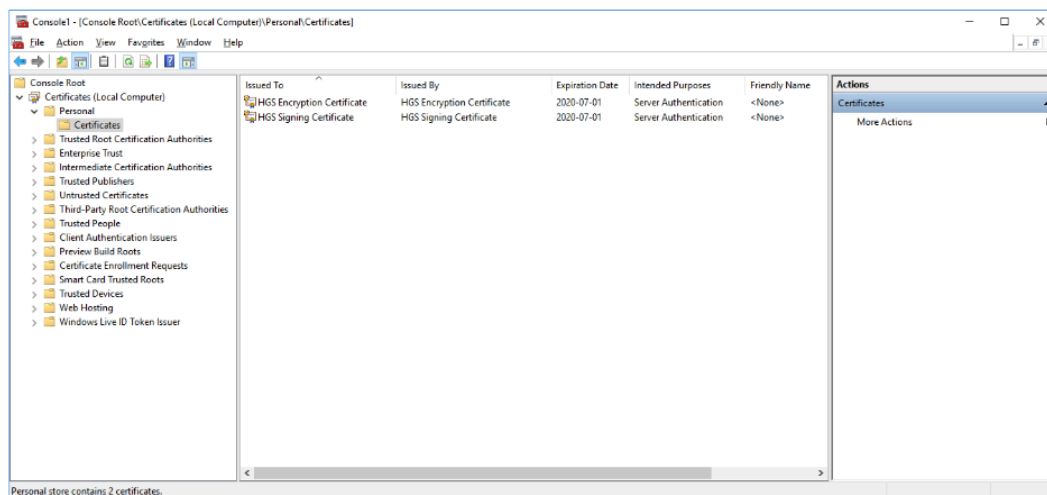
### Step 4 Verify Storage of HGS Certificates in Microsoft Certificate Store

Verify that corresponding HGS certificates have been stored in Microsoft certificate store. Launch the Microsoft Management Console (MMC) by going to the command line and typing MMC.exe.

- a. In MMC, select **File > Add/remove Snap-in**.
- b. In the Add or Remove Snap-ins window, select the option **Certificates > Computer Account >**

**Local Computer.**

- c. In the Certificates (Local Computer) console, expand the folders **Personal > Certificates**, and verify that the self-signed HGS signing and encryption certificates appear.



**Figure – Example of HGS certificates in Microsoft certificate store**

For more information on how to generate HGS signing and encryption keys and certificates, see [Microsoft's documentation on HGS certificate management](#).

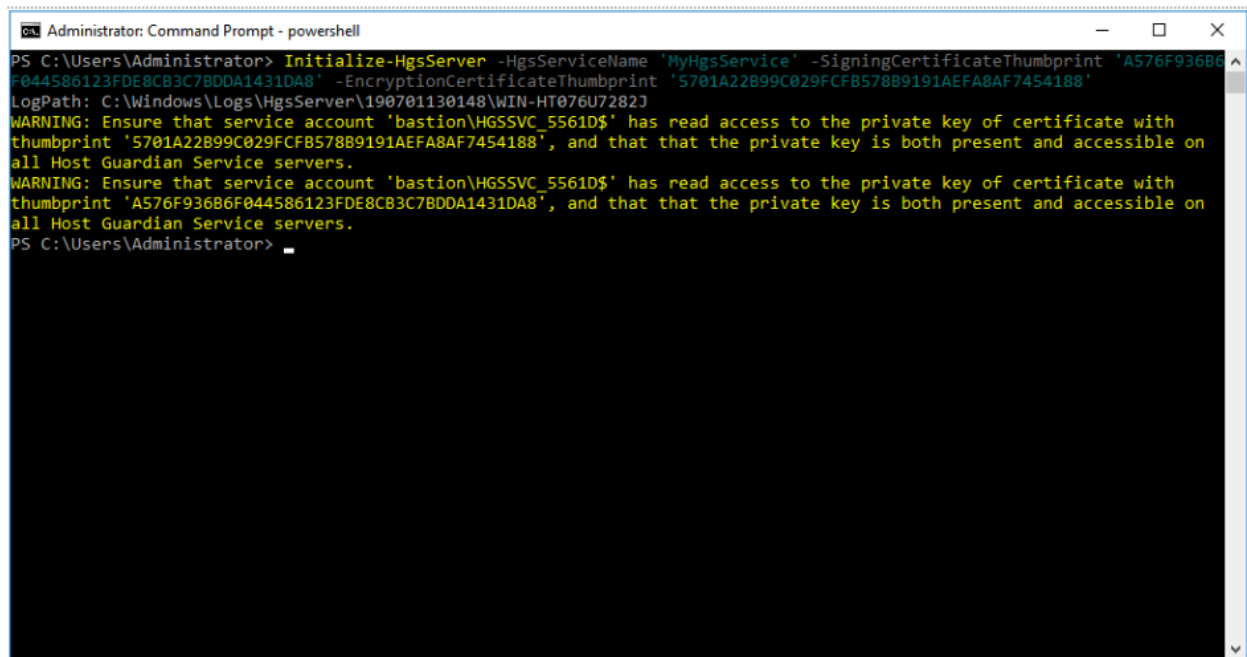
## 4.2 Initialize HGS with Signing and Encryption Keys and Certificates

Once the HGS signing and encryption keys and certificates have been generated, use them to initialize HGS.

Create the self-signed HGS certificates by starting a command prompt with administrator rights and typing the command PowerShell. In the PowerShell command prompt, run the following cmdlet to initialize HGS with the signing and encryption certificates.

**Note:** The parameters `SigningCertificateThumbprint` and `EncryptionCertificateThumbprint` should be set to the output values from the PowerShell cmdlet `New-SelfSignedCertificate` as described in the previous section.

```
PS C:\users\your-username\ Initialize-HgsServer -HgsServiceName
'MyHgsService' -SigningCertificateThumbprint
'<SigningCertificateThumbprint>' -EncryptionCertificateThumbprint
'<EncryptionCertificateThumbprint>'
```



```
Administrator: Command Prompt - powershell
PS C:\Users\Administrator> Initialize-HgsServer -HgsServiceName 'MyHgsService' -SigningCertificateThumbprint 'A576F936B6F044586123FDE8CB3C7BDDA1431DA8' -EncryptionCertificateThumbprint '5701A22B99C029FCFB578B9191AEFA8AF7454188'
LogPath: C:\Windows\Logs\HgsServer\190701130148\WIN-HT076U7282J
WARNING: Ensure that service account 'bastion\HGSSVC_5561D$' has read access to the private key of certificate with thumbprint '5701A22B99C029FCFB578B9191AEFA8AF7454188', and that that the private key is both present and accessible on all Host Guardian Service servers.
WARNING: Ensure that service account 'bastion\HGSSVC_5561D$' has read access to the private key of certificate with thumbprint 'A576F936B6F044586123FDE8CB3C7BDDA1431DA8', and that that the private key is both present and accessible on all Host Guardian Service servers.
PS C:\Users\Administrator>
```

**Figure – Example of PowerShell cmdlet to initialize HGS with the certificates**

For more information on how to initialize HGS with the signing and encryption certificates, see [Microsoft's documentation on HGS initialization](#).



## BACK UP KEY MATERIAL

Yubico strongly recommends making a backup copy of all production objects residing on your production devices, particularly once the HGS signing and encryption keys have been generated on the YubiHSM 2. If there is a hardware failure of the production device, having a backup ensures that you can resume operations quickly. The backup process will result in two identical YubiHSM 2 devices with the same number of objects, keys, labels, etc.

Backup the YubiHSM 2 according to the instructions in [Back Up and Restore Key Material](#) in the YubiHSM 2 Windows Deployment Guide.



## GETTING HELP AND FURTHER READING

Should you require assistance when using this guide to deploy YubiHSM 2 on Windows, start by referencing the product documentation and currently known issues:

- [Yubico Developers website](#)
- [Yubico Support](#)
- [YubiHSM 2 Product Overview](#)
- [YubiHSM 2: Known Issues and Limitations](#)

If you need additional help, contact Yubico directly by filling in a ticket on the [Yubico Support](#) site.

In addition to the Yubico web sites listed above, Microsoft has published the following articles on Host Guardian Services:

- [Install HGS in a new forest](#)
- [Obtain certificates for HGS](#)
- [HGS key management](#)



## TERMINOLOGY

The following terminology as it relates to YubiHSM 2 is used throughout this guide.

**Application:** AES key used to authenticate to the device.

**authentication key** Performs operations according to its defined capabilities.

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

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

**Cryptographic API Next Generation** A CNG is Microsoft's cryptographic architecture, which allows developers to implement applications with features for encryption, electronic signatures, certificate management, etc.

**Default authentication key** Factory-installed Advanced Encryption Standards (AES) key used when initializing the device. Possesses all capabilities.

**Delegated capability** An operation that an object is allowed to perform by virtue of receiving those permissions from 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.

**Guarded Host** This is an attested Hyper-V host machine with a Trusted Platform Module (TPM) that can run shielded Hyper-V VMs.

**Host Guardian Services (HGS)** This is a Windows Server role that is composed of the Attestation Service and Key Protection Services.

**Hyper-V Virtual Machine (VM)** Microsoft Hyper-V is a native hypervisor that can create VMs on x86-64 systems running Windows.

**Key custodian** Holder of a wrap key share.

**Key Storage Provider (KSP)** This is a Dynamic Link Library (DLL) that is loaded by Microsoft CNG. KSPs can be used to create, delete, export, import, open and store keys.

**m of n** Scheme in which wrap key is split into a total number of shares (n) held by key custodians, where a minimum number of shares (m) (sometimes called a *quorum*, and sometimes a *privacy* threshold) is needed to regenerate and use the key.

**Object ID** These are unique identifiers for any kind of object stored on YubiHSM2. An ID can range from 1 to 65535; however, the device can hold a maximum of 256 unique objects.

**Shielded VM** This is a Hyper-V VM with a virtual TPM; it is encrypted using BitLocker, and can run only on attested guarded hosts in a guarded fabric.

**Trusted Computing Group (TCG)** This is a group formed by AMD, Hewlett-Packard, IBM, Intel and Microsoft to implement Trusted Computing concepts across personal computers.

**Trusted Platform Module (TPM)** This is a cryptographic chip on a device that stores RSA encryption keys specific to the host system for hardware authentication.

**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.

## COPYRIGHT

© 2022 Yubico AB. All rights reserved.

### 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

2022-05-11 23:53:26 UTC