# SECURE BOOT MANAGEMENT ON DELL EMC POWEREDGE SERVERS

This technical whitepaper provides information about Secure Boot concepts and how to configure it on PowerEdge Servers.

## ABSTRACT

As software-based attacks grow more sophisticated, system administrators must employ a wider variety of defenses. Many solutions include protection for the operating system (OS) environment, but neglect the firmware which executes before the OS boots.

The New Dell EMC PowerEdge servers provide customers an uncompromising security by adopting the industry's best practices in UEFI Secure Boot. At the time of this white paper's release, PowerEdge servers support Secure Boot on RACADM, WS-Man, Redfish, BIOS, Lifecycle Controller graphical user interface (GUI).

This technical white paper describes step-step process to manage Secure Boot and its certificates effectively to ensure your business-critical operations are safe and uninterrupted.

June, 2017

Version 1.0

**Authors** (Dell EMC Server Solutions)

Vinod P
Lokesh Kumar
Balakrishna Padhy
Shekar Babu S
Revathi U
Raghavendra Venkataramudu
Sheshadri PR Rao (InfoDev)

WHITE PAPER

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

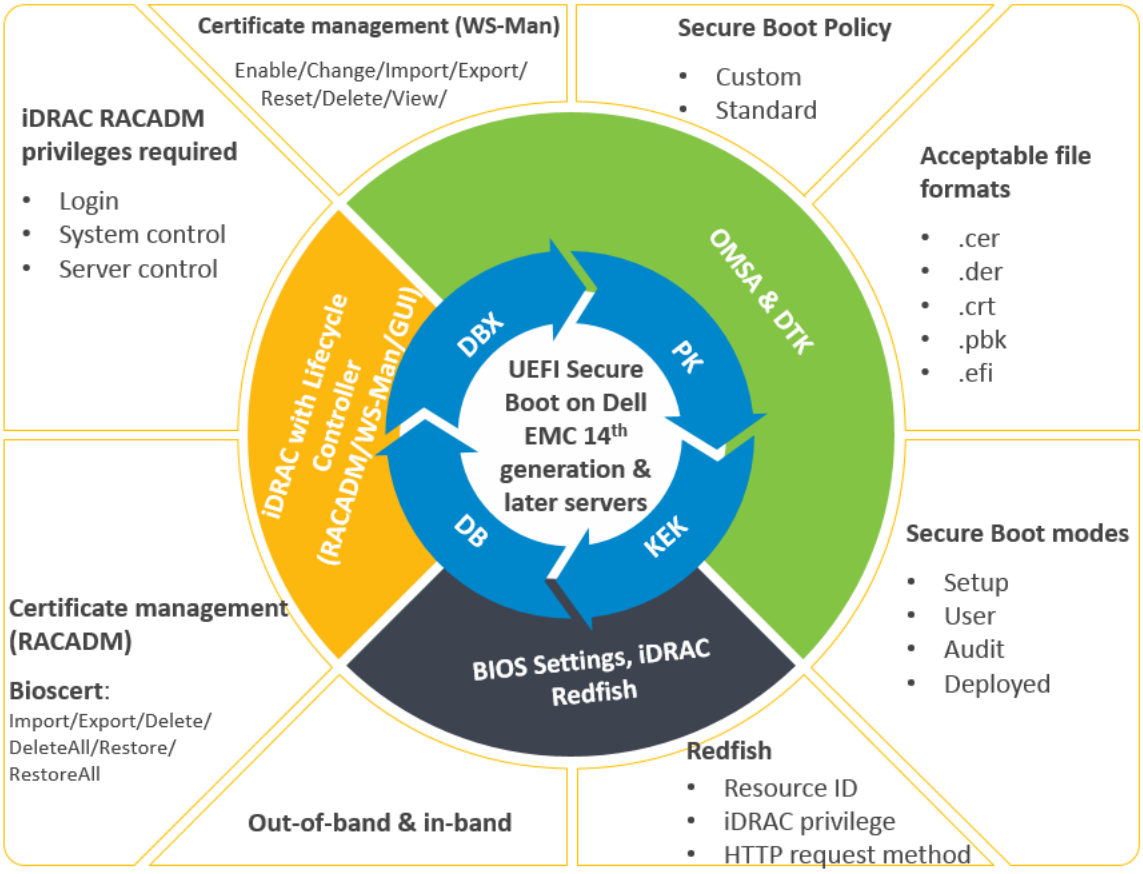
Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © June-2017 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA [date 06/17] [Technical White Paper]

Dell EMC believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

## CONTENTS

# EXECUTIVE SUMMARY


Certificate management (WS-Man)
Enable/Change/Import/Export/ Reset/Delete/View/
Secure Boot Policy
• Custom
• Standard
iDRAC RACADM privileges required
• Login
• System control
• Server control
Acceptable file formats
• .cer
• .der
• .crt
• .pbk
• .efi
OMSA & DTK
iDRAC with Lifecycle Controller (RACADM/WS-Man/GUI)
DBX
PK
UEFI Secure Boot on Dell EMC 14th generation & later servers
DB
KEK
Secure Boot modes
• Setup
• User
• Audit
• Deployed
Certificate management (RACADM)
Bioscert:
Import/Export/Delete/ DeleteAll/Restore/ RestoreAll
BIOS Settings, iDRAC Redfish
Redfish
• Resource ID
• iDRAC privilege
• HTTP request method
Out-of-band & in-band

As software-security breaches are becoming more frequent and in-cognitive, system administrators must employ a wider variety of defenses. Admins mainly look for operating system (OS) environment protection, but ignore the firmware itself which executes before the OS environments comes into existence.

Attackers find this pre-boot environment lucrative. Pre-boot malware avoids OS privilege levels, escapes detection by OS anti-malware tools, and even survives re-installation of the OS. If an attacker injects malware into the pre-boot environment, administrators may find it difficult to remove, if they detect it at all.

This technical whitepaper discusses Secure Boot, a Basic Input/output System (BIOS) feature that protects the pre-boot environment. Readers will learn what is Secure Boot, how Secure Boot works, and how to configure a Secure boot from various user interfaces on PowerEdge servers.

## Audience

This technical white paper is intended for server administrators, architects, and other stake holders in decision making capacities. The reader is expected to have basic knowledge about server management applications and troubleshooting techniques on PowerEdge servers.

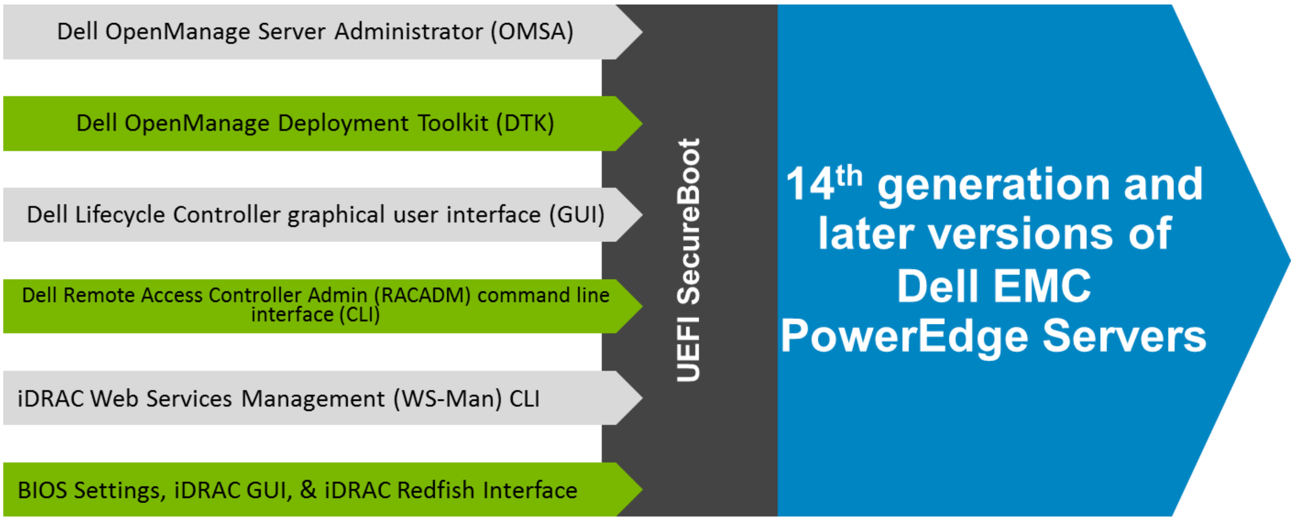
# Introduction

UEFI Secure Boot is a technology that eliminates a major security void that may occur during a handoff between the UEFI firmware and UEFI operating system (OS). In UEFI Secure Boot, each component in the chain is validated and authorized against a specific certificate before it is allowed to load or run. Secure Boot removes the legacy threat and provides software identity checking at every step of the boot—Platform firmware, Option-ROMs, and OS BootLoader.

The Unified Extensible Firmware Interface (UEFI) Forum—an industry body that develops standards for pre-boot software—defines Secure Boot in the UEFI specification. Computer system vendors, expansion card vendors, and operating system providers collaborate on this specification to promote interoperability. As a portion of the UEFI specification, Secure Boot represents an industry-wide standard for security in the pre-boot environment.

When enabled, UEFI Secure Boot prevents the unsigned UEFI device drivers from being loaded, displays an error message, and does not allow the device to function. You must disable Secure Boot to load the unsigned device drivers.

On the Dell 14th generation and later versions of PowerEdge servers, you can enable or disable the Secure Boot feature by using different interfaces.


Dell OpenManage Server Administrator (OMSA)
Dell OpenManage Deployment Toolkit (DTK)
Dell Lifecycle Controller graphical user interface (GUI)
Dell Remote Access Controller Admin (RACADM) command line interface (CLI)
iDRAC Web Services Management (WS-Man) CLI
BIOS Settings, iDRAC GUI, & iDRAC Redfish Interface
UEFI SecureBoot
14th generation and later versions of Dell EMC PowerEdge Servers

## *Secure Boot importance*

Technology solutions without Secure Boot may be vulnerable to firmware rootkits and bootkits. Attackers use firmware rootkits to hide malicious code in device firmware or system firmware. Bootkits infect the software that boots the OS. In addition to compromise the security of the system, firmware rootkits and bootkits can escape detection techniques in OS tools and survive reinstallation of the OS. Secure Boot guards against these attacks because it prevents execution of unauthorized pre-boot code.

## *UEFI Secure Boot working principle*

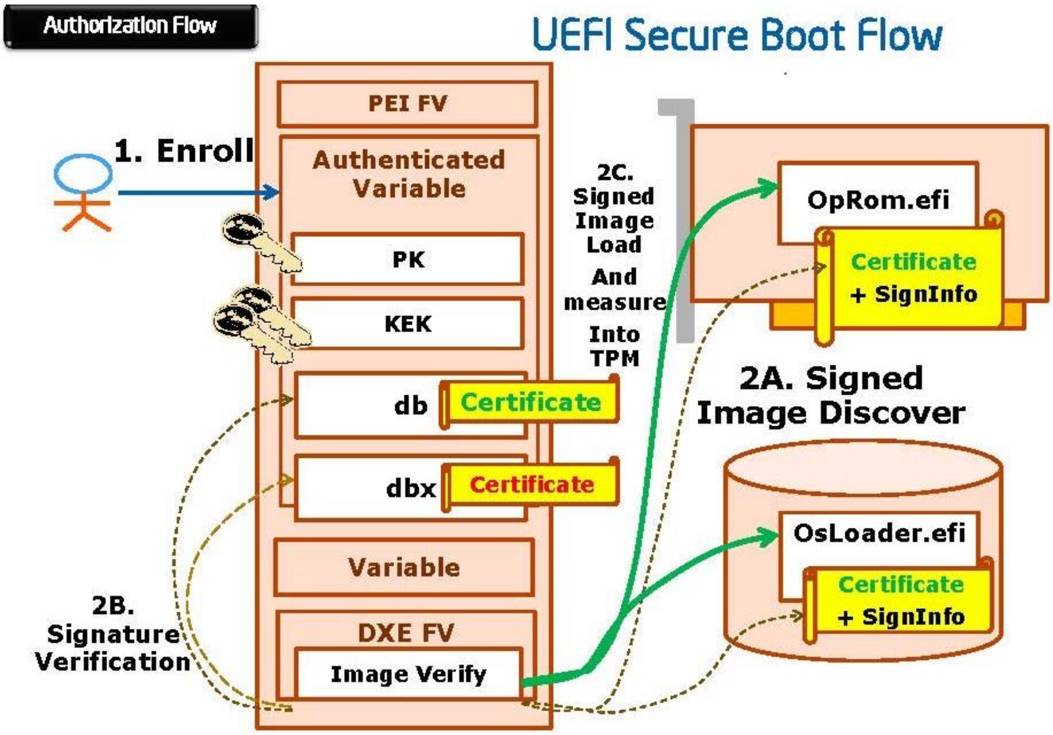
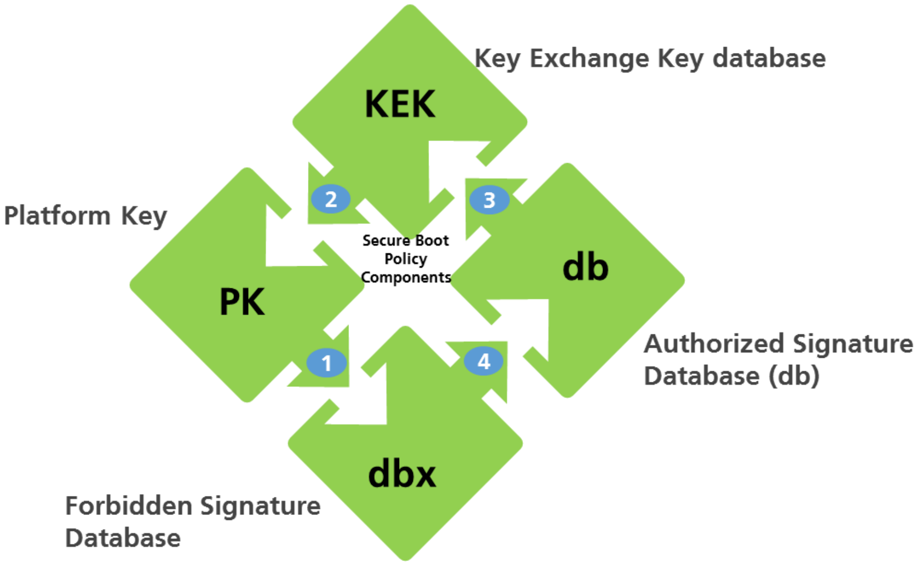


Figure 1. UEFI Secure Boot working principle

## Secure Boot policy components


Key Exchange Key database
KEK
Platform Key
PK
2
3
Secure Boot Policy Components
db
1
4
Authorized Signature Database (db)
dbx
Forbidden Signature Database

The system BIOS uses the first two components (PK and KEK) to verify changes to the Secure Boot policy itself. The last two components (db and dbx) help the system BIOS determine whether or not to execute a pre-boot image file.

Secure Boot policies contain public keys only. Private keys do not reside anywhere on the system. The system BIOS uses public keys to verify signatures, while module providers use private keys to sign modules. Owners of private keys use specialized hardware and techniques for protecting the keys such as Hardware Security Modules (HSMs), secure Smart Cards, or a Key Management System (KMS). Neither the system BIOS nor Secure Boot require private keys during the boot process.

First, consider the authorization of pre-boot image files:

- The Authorized Signature Database (db) contains public keys, certificates, and image digests for image files authorized to execute.
- If a pre-boot image file includes a digital signature, the BIOS verifies the signature by using the keys and certificates in the database (DB).
- If a pre-boot image file does not contain a digital signature, the system BIOS determines the digest (also known as a hash value) of the image and compares it against the image digests in db.
- The BIOS executes the image file only if it verifies the digital signature by using a key in db, or finds the digest in db.

*The Forbidden Signature Database (dbx) specifies image files that must not execute even if they are allowed by the db. Similar to db, dbx may contain public keys, certificates, or hash values. The BIOS will not execute an image if it verifies the image's digital signature with a key in dbx, or finds the image's hash value in dbx.*

Meaning, db acts as a "whitelist" and dbx acts as a "blacklist". To execute an image file, Secure Boot must verify that the image file is on the "whitelist" and not on the "blacklist". If Secure Boot does not find the image file in either list, the system BIOS will not execute the image file. Similarly, if Secure Boot finds the image file in both lists, the system BIOS will not execute the image file.

The Secure Boot policy applies to all pre-boot code image files, including device firmware and OS boot loaders. When installing expansion cards or operating systems, make sure that db includes information to authorize the images (and dbx does not forbid them). Otherwise, the images will not execute.

## PK and KEK control policy updates

Second, consider changes to the Secure Boot policy itself. Periodically, administrators might add or remove entries in the policy, and attackers might attempt malicious updates to the policy. Anyone wanting to modify db or dbx must sign their modifications with the private PK or KEK. In this way, the BIOS can use the public keys in PK and KEK to verify updates to db and dbx. Therefore, if an attacker attempts to modify db or dbx, the signature verification with PK and KEK fails (because the attacker does not possess the private PK or KEK), and the system BIOS does not permit the modifications.

Also, any agent wanting to modify PK or KEK must possess the private half of PK. PK acts as a master key—anyone with access to the private half of PK can modify any portion of the Secure Boot policy. Figure illustrates the relationship between PK, KEK, db, and dbx.

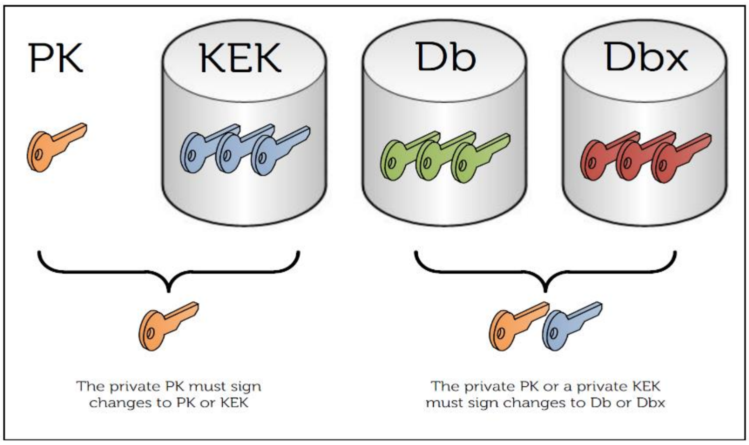

PK
KEK
Db
Dbx
The private PK must sign changes to PK or KEK
The private PK or a private KEK must sign changes to Db or Dbx

Figure 2. Relationship between PK, KEK, DB, and DBX

The Secure Boot policy contains only one key in PK, but multiple keys may reside in KEK. Ideally, either the platform manufacturer or platform owner maintains the private key corresponding to the public PK. Third parties (such as OS providers and device providers) maintain the private keys corresponding to the public keys in KEK. In this way, platform owners or third parties may add or remove entries in db or dbx.

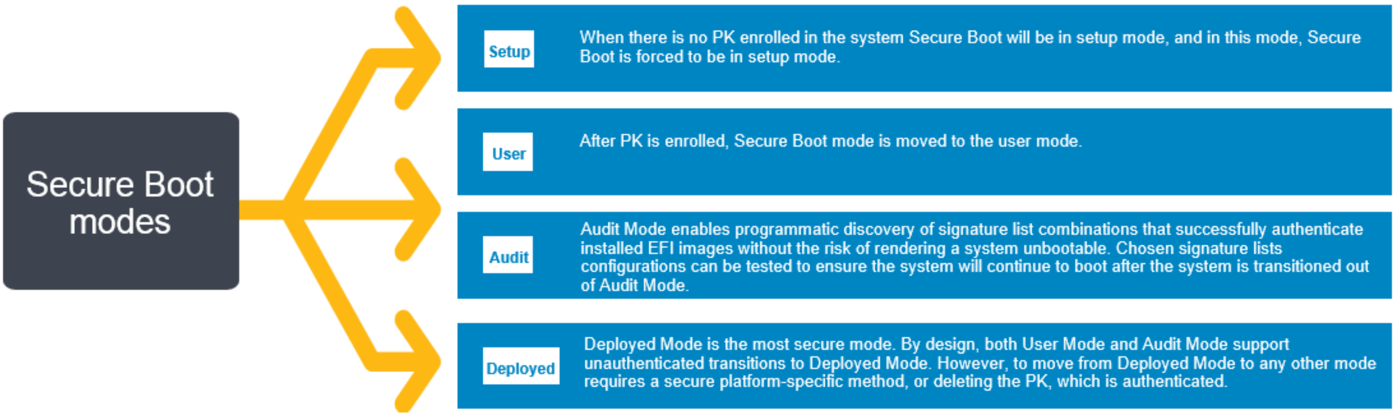
Observe that the owner of a private KEK possesses similar authority as the owner of a private PK. Similar to the private PK owner, owners of private KEKs can authorize or prevent module execution by updating db and dbx. The private PK owner possesses slightly more authority—they can modify the contents of KEK or PK.

In summary, the Secure Boot policy uses db and dbx to authorize pre-boot image file execution. For an image file to get executed, it must associate with a key or hash value in db, and not associate with a key or hash value in dbx. Any attempts to update the contents of db or dbx must be signed by a private PK or KEK. Any attempts to update the contents of PK or KEK must be signed by a private PK.

## *Acceptable file formats*

| Policy Component | Acceptable File Formats | Acceptable File Extensions | Max records allowed |
|---|---|---|---|
| **PK** | X.509 Certificate (binary DER format only) | `.cer`<br>`.der`<br>`.crt` | One |
| **KEK** | X.509 Certificate (binary DER format only)<br>Public Key Store | `.cer`<br>`.der`<br>`.crt`<br>`.pbk` | More than one |
| **DB and DBX** | X.509 Certificate (binary DER format only)<br>EFI image (system BIOS will calculate and import image digest) | `.cer`<br>`.der`<br>`.crt`<br>`.efi` | More than one |

## *Secure Boot modes*



When there is no PK enrolled in the system Secure Boot will be in setup mode, and in this mode, Secure Boot is forced to be in setup mode.

After PK is enrolled, Secure Boot mode is moved to the user mode.

Audit Mode enables programmatic discovery of signature list combinations that successfully authenticate installed EFI images without the risk of rendering a system unbootable. Chosen signature lists configurations can be tested to ensure the system will continue to boot after the system is transitioned out of Audit Mode.

Deployed Mode is the most secure mode. By design, both User Mode and Audit Mode support unauthenticated transitions to Deployed Mode. However, to move from Deployed Mode to any other mode requires a secure platform-specific method, or deleting the PK, which is authenticated.

The flow diagram describes the transition between different Secure Boot modes.

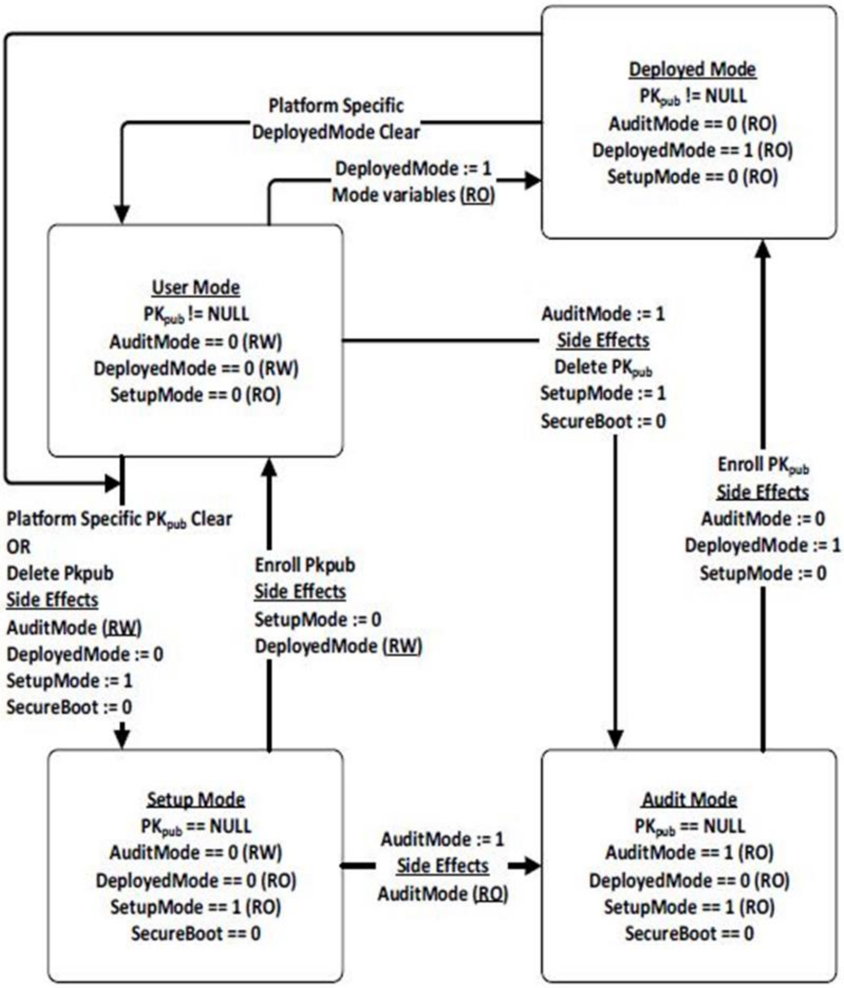
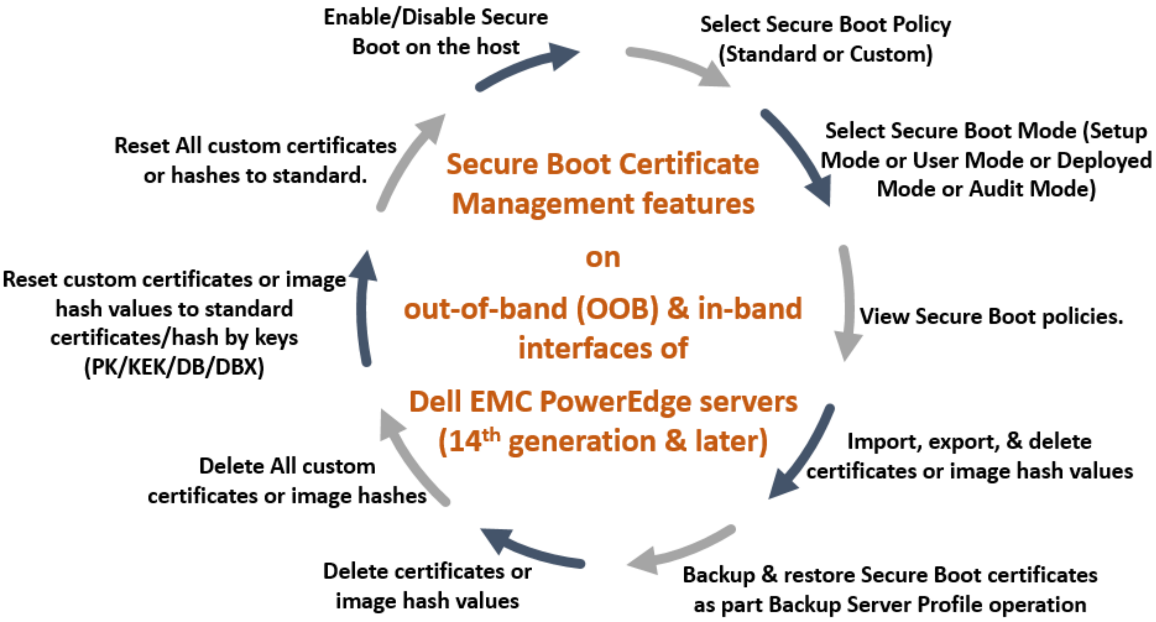


Figure 3. Transition between Secure Boot modes

## *Secure Boot Certificate management features*


Secure Boot Certificate Management features
on
out-of-band (OOB) & in-band interfaces of
Dell EMC PowerEdge servers (14th generation & later)
Enable/Disable Secure Boot on the host
Select Secure Boot Policy (Standard or Custom)
Select Secure Boot Mode (Setup Mode or User Mode or Deployed Mode or Audit Mode)
View Secure Boot policies.
Import, export, & delete certificates or image hash values
Backup & restore Secure Boot certificates as part Backup Server Profile operation
Delete certificates or image hash values
Delete All custom certificates or image hashes
Reset custom certificates or image hash values to standard certificates/hash by keys (PK/KEK/DB/DBX)
Reset All custom certificates or hashes to standard.

- By default, Secure Boot will be in disabled state and Secure Boot Policy will be set to ‘Standard’. When the server is shipped out of the factory, every PowerEdge server is installed with standard set of certificates or image digests which support the Secure Boot feature. If the customers want different set of policies, the customers must change the Secure Boot Policy to ‘Custom’, and then use one or more of the features listed here to configure Secure Boot Policies of their choice.

- The ‘Import Certificate’ feature gives you the flexibility to configure a new certificate of your choice. This feature enables you to configure a new certificate to authenticate a new or existing driver or firmware during the secure boot.

- ‘Export Certificate’ can be used to export an existing certificate on the system. Export operation can be performed on any certificate or hash record irrespective of the policies (PK/KEK/DB/DBX).

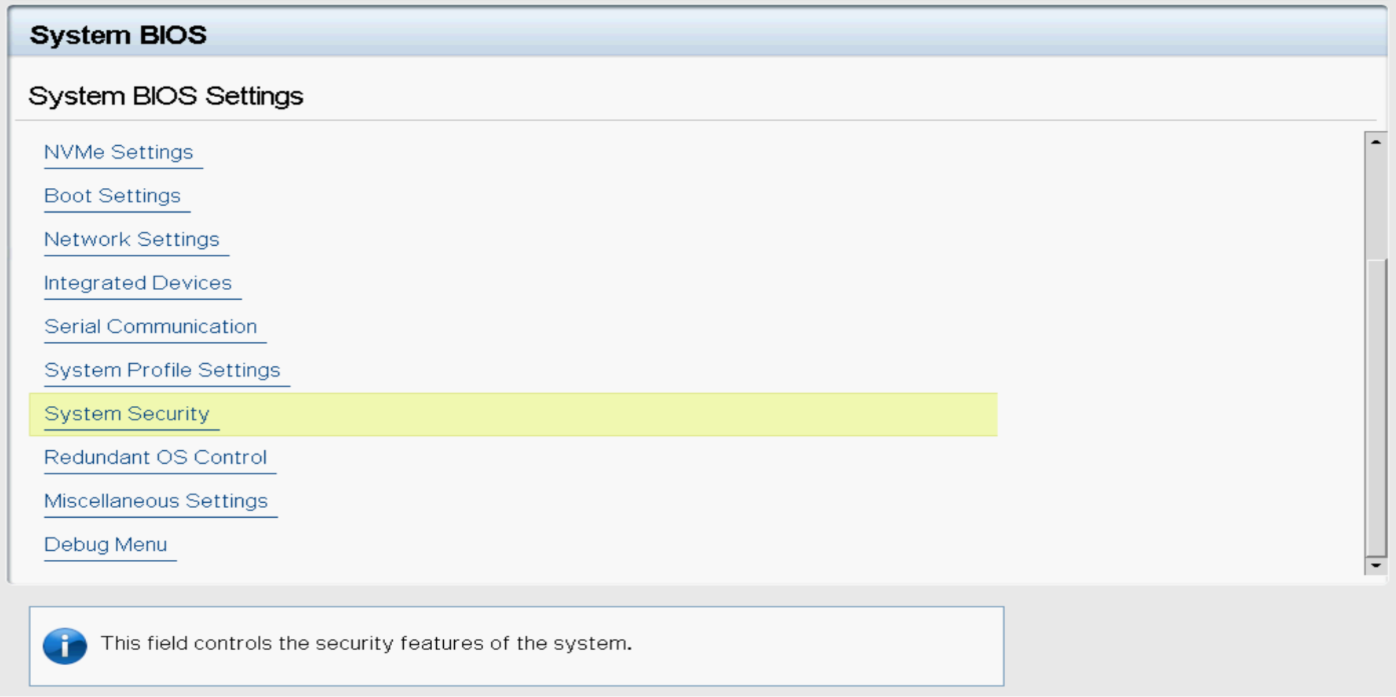
- ‘Delete’ operations can be performed if you want to wipe off the standard certificates and have your own custom certificates. The delete operation can be performed on specific record or complete certificate store. The Delete All operation will wipe out all the custom certificates or hash records present in the system.

- ‘Reset’ operations can be performed if one wants to restore the standard policies. Reset operations can be performed policy wise or on complete certificate store. Reset PK will restore only the PK from standard store whereas Reset All will restore all the policies from standard store.

- Performing any of these operations except View and Export will result in creating a pending task for the respective request which will be serviced during the next host system reboot. Running the pending task will sync the request operations on BIOS certificate store which is ultimately used for performing Secure Boot.

## *Secure Boot certificate management using BIOS settings*

The Secure Boot Settings feature can be accessed by clicking **System Security** under **System BIOS Settings**. To go to System BIOS, press F2 when the company logo is displayed while restarting the server.



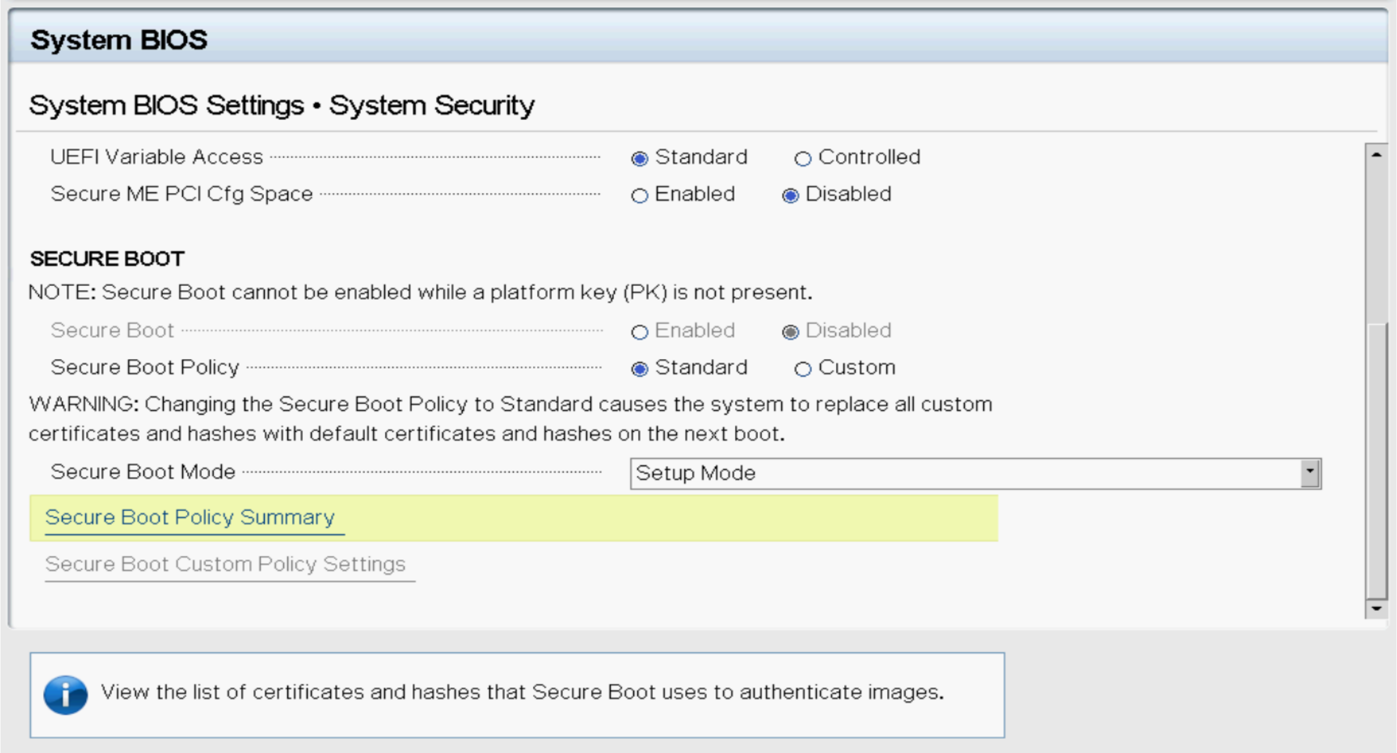
This field controls the security features of the system.

Figure 4. BIOS Settings

- By default, Secure Boot is in the Disabled mode and the Secure Boot policy will be set to Standard. If the Secure Boot needs to be made active then the Secure Boot must be configured as Enabled.
- Secure Boot Policy in Standard means that the system will have default certificates and image digests loaded from the factory which will cater to the security of standard firmware, drivers, option-roms, and boot loader loaded from the factory.
- But, if a new driver or firmware has to be supported on the server then the respective certificate must be enrolled into the DB of Secure Boot certificate store. Therefore, Secure Boot Policy must be configured to Custom.

When the Secure Boot Policy is configured as Custom, it inherits the standard certificates and image digests loaded in the system by default, on which, you can make any modifications as necessary. Secure Boot Policy configured as Custom allows you to perform operations such as View, Export, Import, Delete, Delete All, Reset, and Rest All, by using which, you can configure the Secure Boot Policies according to your requirements.

*Secure Boot Policy configured as Standard restricts the operations to be performed on the certificate store. Standard Secure Boot Policy restricts the user to only View the certificates. No other actions on certificate store are allowed.*



System BIOS

System BIOS Settings • System Security

UEFI Variable Access ● Standard ○ Controlled

Secure ME PCI Cfg Space ○ Enabled ● Disabled

SECURE BOOT

NOTE: Secure Boot cannot be enabled while a platform key (PK) is not present.

Secure Boot ○ Enabled ● Disabled

Secure Boot Policy ● Standard ○ Custom

WARNING: Changing the Secure Boot Policy to Standard causes the system to replace all custom certificates and hashes with default certificates and hashes on the next boot.

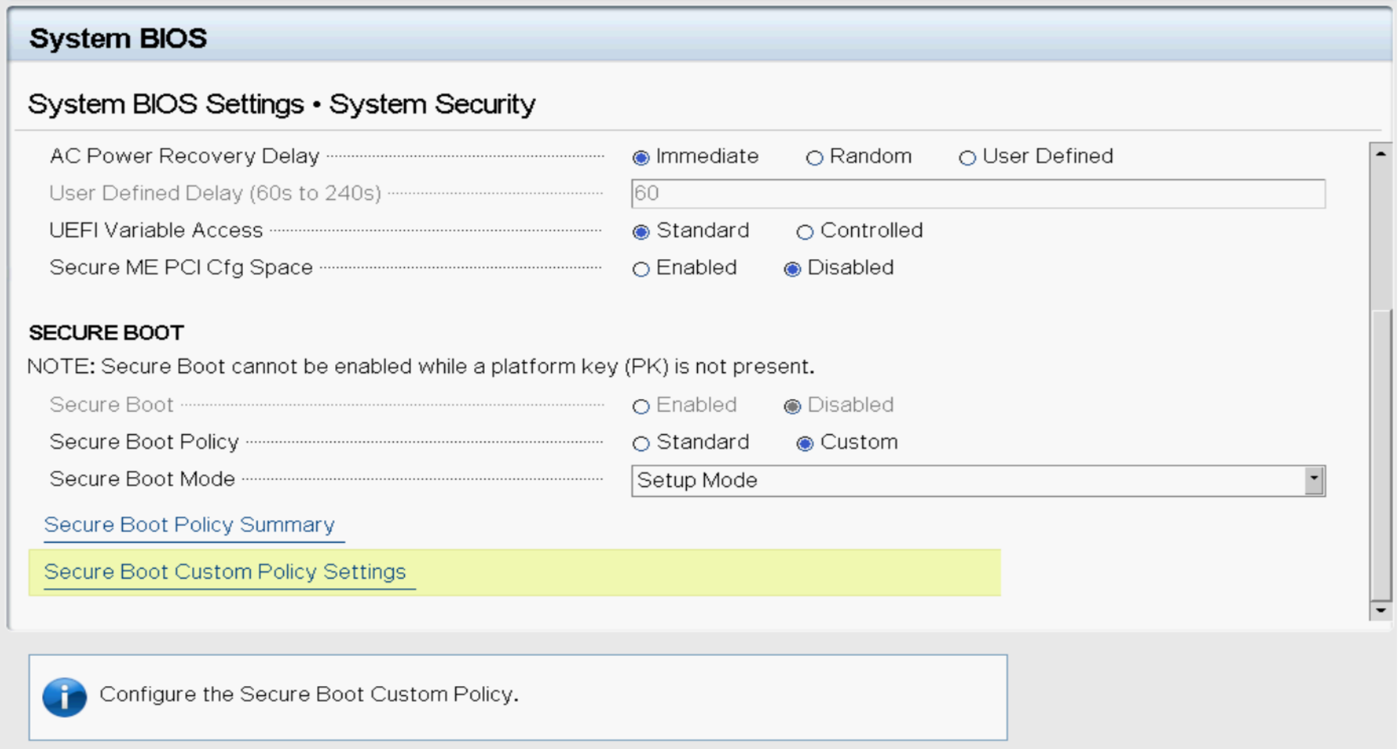
Secure Boot Mode: Setup Mode

Secure Boot Policy Summary

Secure Boot Custom Policy Settings

View the list of certificates and hashes that Secure Boot uses to authenticate images.

**Figure 5. Secure Boot Policy Summary**



System BIOS

System BIOS Settings • System Security

AC Power Recovery Delay ● Immediate ○ Random ○ User Defined

User Defined Delay (60s to 240s): 60

UEFI Variable Access ● Standard ○ Controlled

Secure ME PCI Cfg Space ○ Enabled ● Disabled

SECURE BOOT

NOTE: Secure Boot cannot be enabled while a platform key (PK) is not present.

Secure Boot ○ Enabled ● Disabled

Secure Boot Policy ○ Standard ● Custom

Secure Boot Mode: Setup Mode

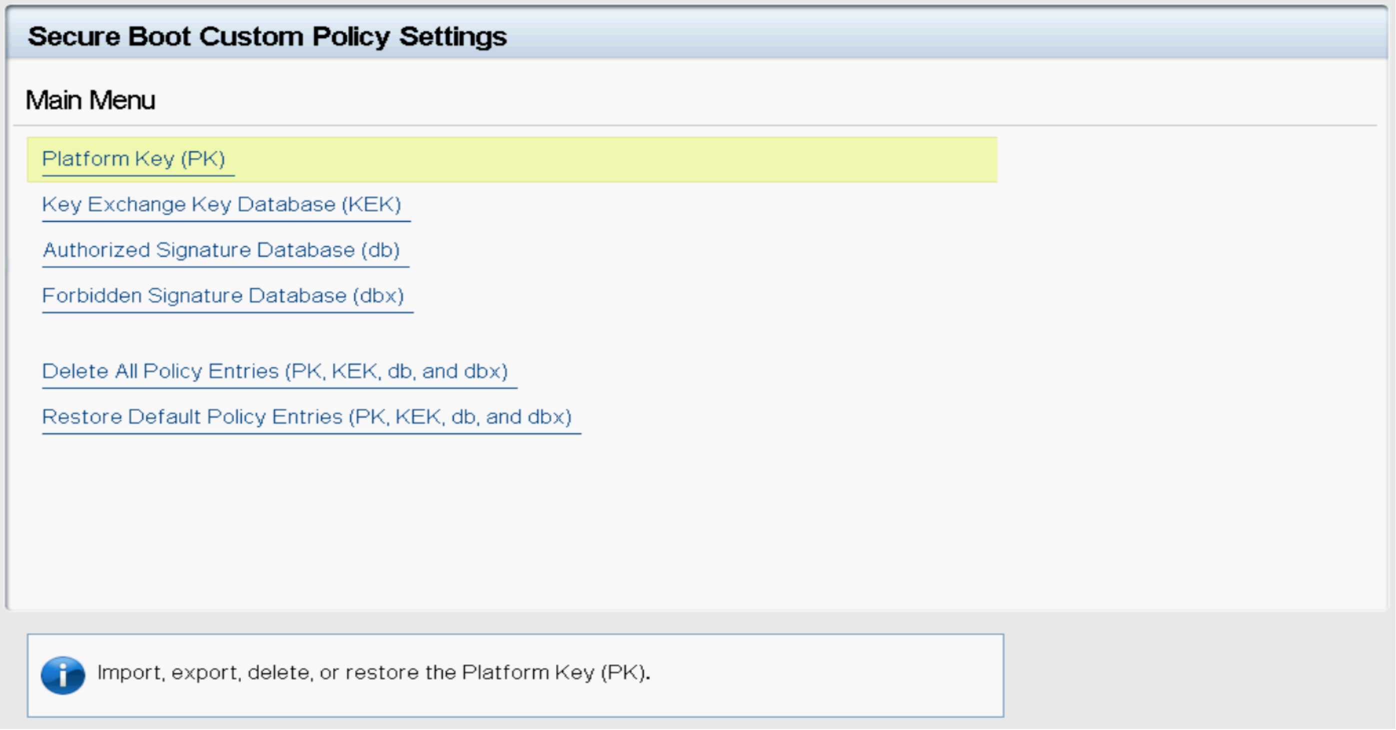
Secure Boot Policy Summary

Secure Boot Custom Policy Settings

Configure the Secure Boot Custom Policy.

**Figure 6. Secure Boot Custom Policy Settings**

Configuring the Secure Boot Policy to `Custom` enables the options to manage the certificate store by using various actions such as Export, Import, Delete, Delete All, Reset, and Rest All on PK, KEK, DB, and DBX. You can select the policy (PK / KEK / DB / DBX) on which you want to make the change and perform appropriate actions.


Secure Boot Custom Policy Settings

Main Menu

Platform Key (PK)

Key Exchange Key Database (KEK)

Authorized Signature Database (db)

Forbidden Signature Database (dbx)

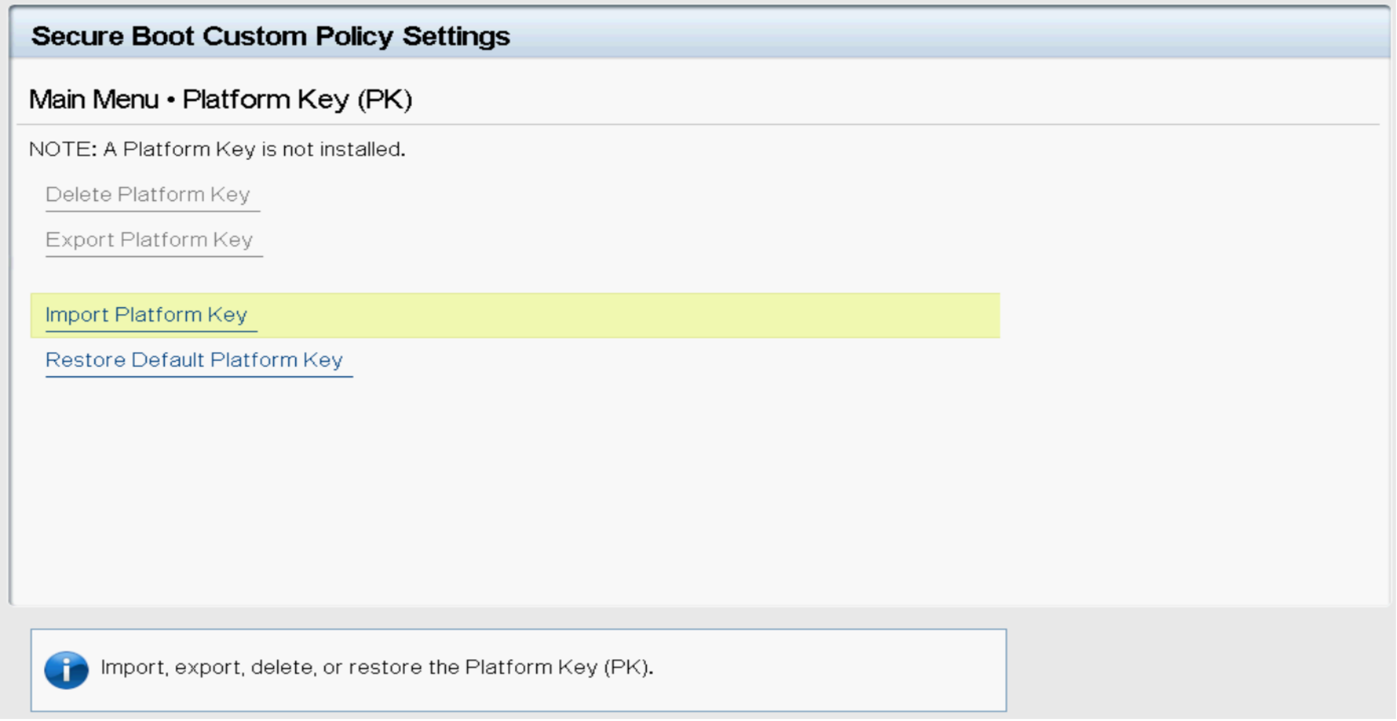
Delete All Policy Entries (PK, KEK, db, and dbx)

Restore Default Policy Entries (PK, KEK, db, and dbx)

Import, export, delete, or restore the Platform Key (PK).

**Figure 7. Platform Key (PK)**

Each section will have links to perform the Import, Export, Delete, and Reset operations. Links are enabled based on what is applicable, which depends on the configuration at the point of time. For example, in the screen shot here, Delete and Export operations are disabled because there is no PK configured yet.


Secure Boot Custom Policy Settings

Main Menu • Platform Key (PK)

NOTE: A Platform Key is not installed.

Delete Platform Key

Export Platform Key

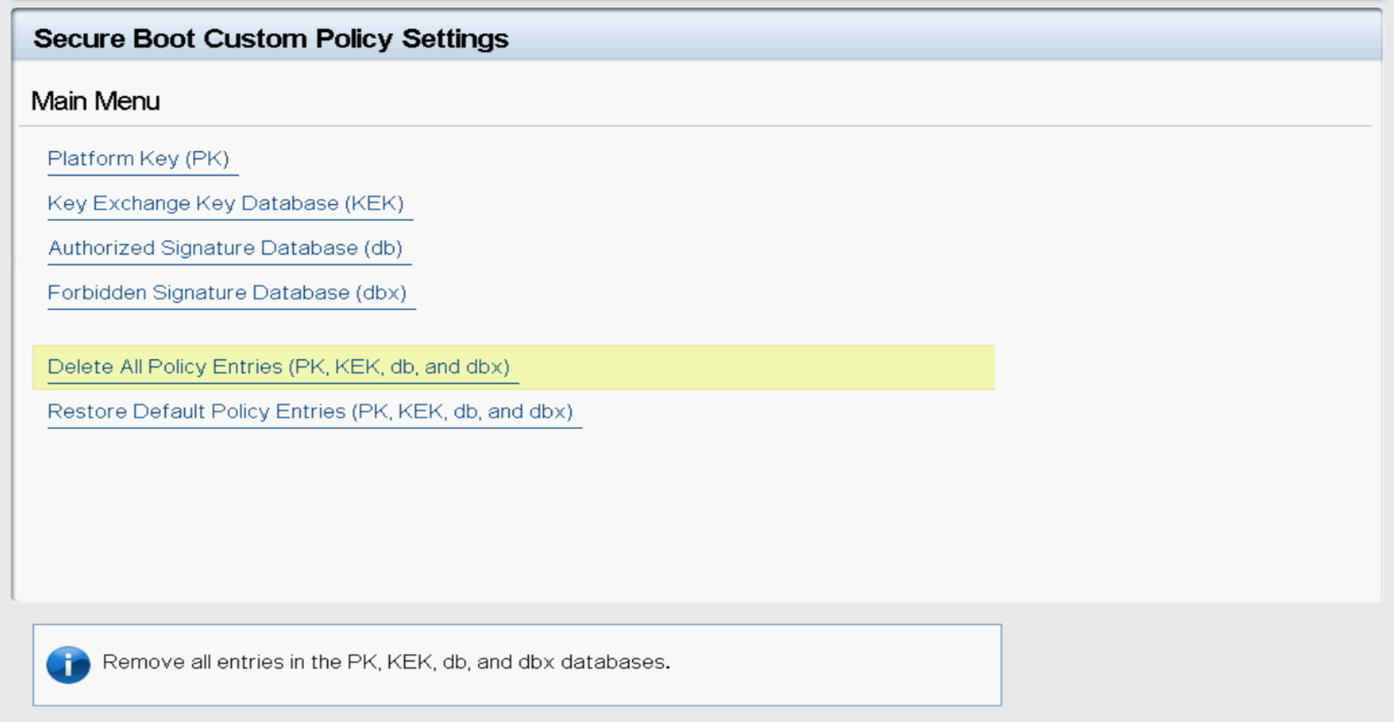
Import Platform Key

Restore Default Platform Key

Import, export, delete, or restore the Platform Key (PK).

**Figure 8. Import Platform Key**

The *Delete All* and *Reset All* are the operations that have impact on all the policies. *Delete All* deletes all the certificates and image digests in the Custom policy, and *Rest All* restores all the certificates and image digests from *Standard* or *Default* certificate store.



Secure Boot Custom Policy Settings

Main Menu

Platform Key (PK)

Key Exchange Key Database (KEK)

Authorized Signature Database (db)

Forbidden Signature Database (dbx)

Delete All Policy Entries (PK, KEK, db, and dbx)

Restore Default Policy Entries (PK, KEK, db, and dbx)

Remove all entries in the PK, KEK, db, and dbx databases.

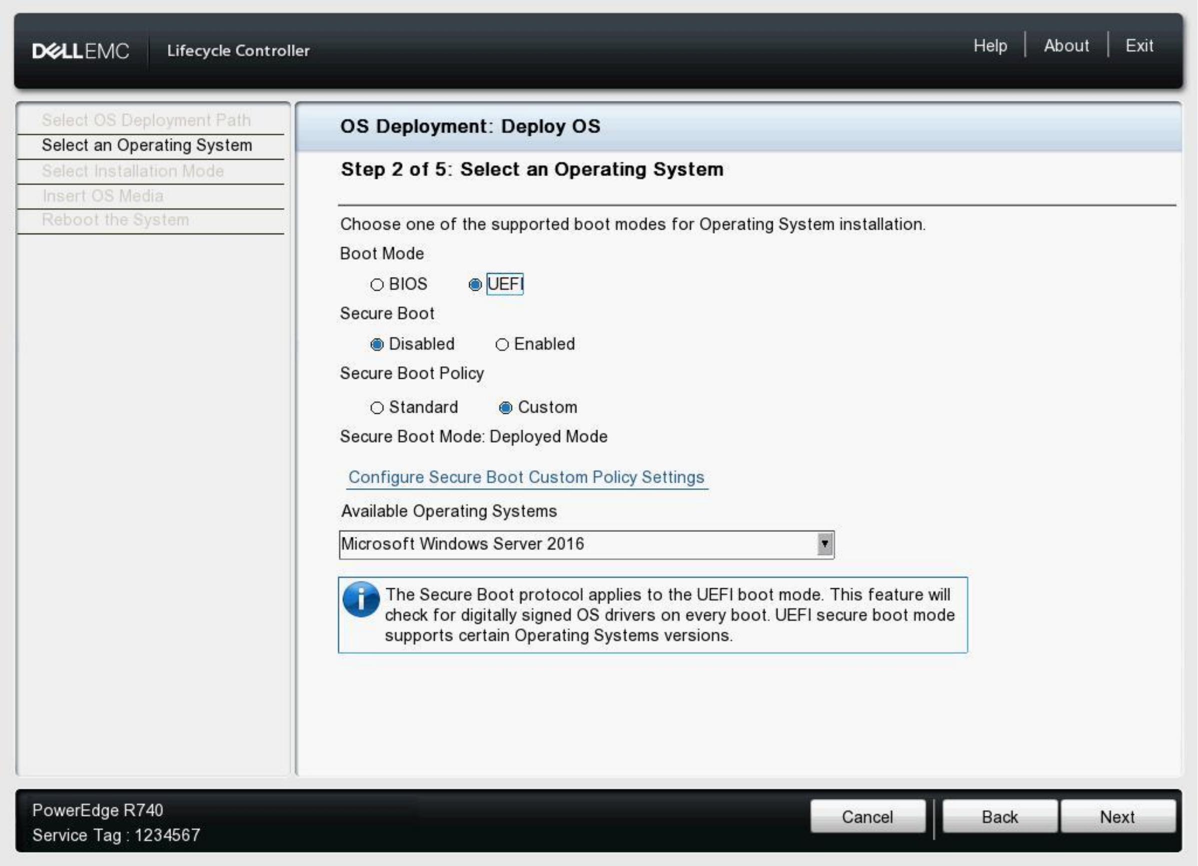
Figure 9. Delete All Policy Entries (PK, KEK, db, and dbx)

## *Secure Boot certificate management using Lifecycle Controller GUI*

- Lifecycle Controller GUI supports enabling and disabling Secure Boot and allows you to set the secure Boot policy to Standard or Custom. Secure Boot Mode is a read-only option in Lifecycle Controller GUI.

- Lifecycle Controller GUI provides the Configure Secure Boot Custom Policy Settings hyperlink which directs you to **Secure Boot Custom Policy Setting** page of the **BIOS Settings** page.

- You can perform all the Secure Boot certificate management actions such as View, Export, Import, Delete, Delete All, Reset, and Reset All as though it is performed from the BIOS Settings by pressing F2 when the company logo is displayed while starting the server.



OS Deployment: Deploy OS

Step 2 of 5: Select an Operating System

Choose one of the supported boot modes for Operating System installation.

Boot Mode

○ BIOS ● UEFI

Secure Boot

● Disabled ○ Enabled

Secure Boot Policy

○ Standard ● Custom

Secure Boot Mode: Deployed Mode

Configure Secure Boot Custom Policy Settings

Available Operating Systems

Microsoft Windows Server 2016

The Secure Boot protocol applies to the UEFI boot mode. This feature will check for digitally signed OS drivers on every boot. UEFI secure boot mode supports certain Operating Systems versions.

PowerEdge R740
Service Tag : 1234567

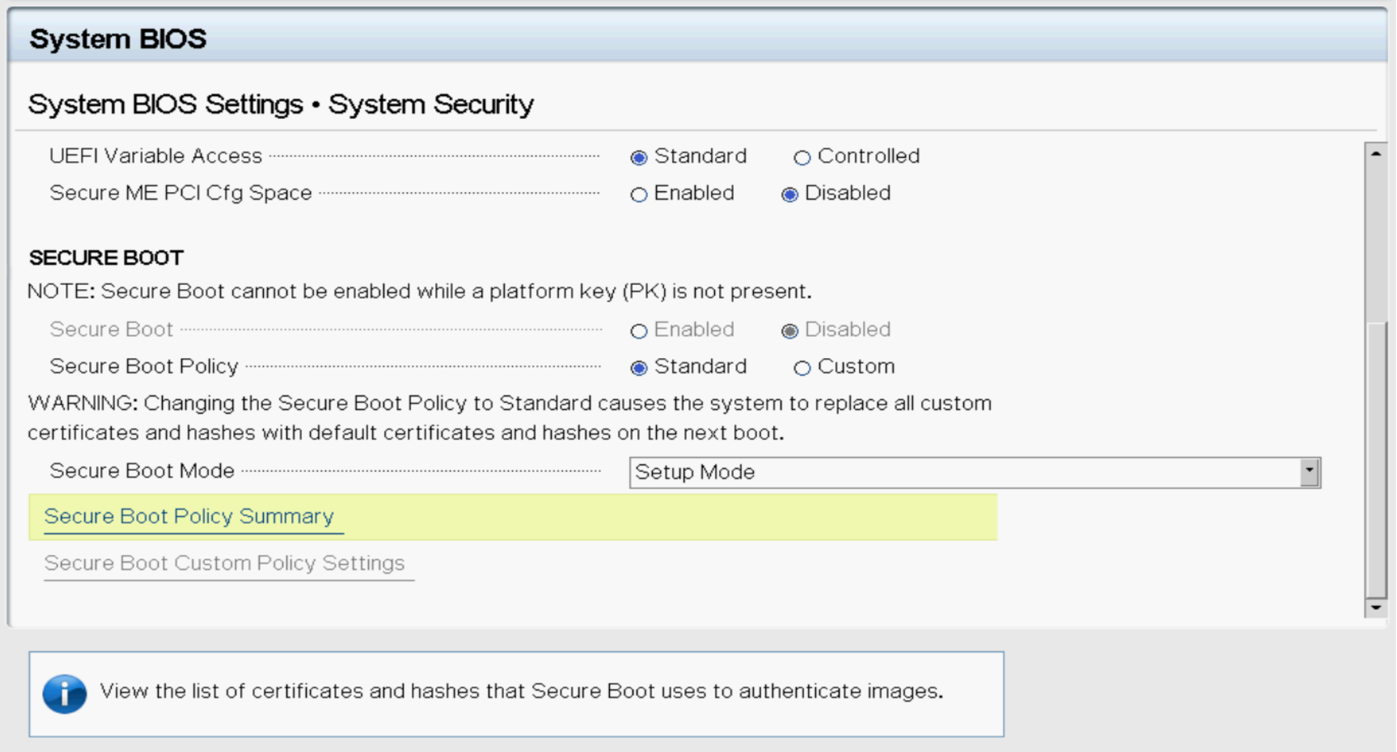
Figure 10. Enabling and disabling Secure Boot by using Lifecycle Controller GUI

By default, Secure Boot will be in the *Disabled* mode and the Secure Boot Policy will be set to Standard. If the Secure Boot needs to be made active then the Secure Boot should be configured as *Enabled*. Secure Boot Policy in Standard means that the system will have default certificates and image digests loaded from the factory which will cater to the security of standard firmware, drivers, option-roms and boot loader loaded from the factory.

But in case a new driver or firmware to be supported on the machine then the respective certificate need to be enrolled into the DB of Secure Boot certificate store. In order to do that Secure Boot Policy need to be configured to Custom. When the Secure Boot Policy is configured as Custom, it inherits the standard certificates and image digests loaded in the system by default on which user can make any modifications, if required.

*Secure Boot Policy configured as Custom allows you to perform operations such as View, Export, Import, Delete, Delete All, Reset, and Rest All by* *using which you can configure the Secure Boot policies according to your requirements.*

Secure Boot Policy configured as `Standard` restricts the operations to be performed on the certificate store. Standard Secure Boot Policy restricts the user to view only the certificates—no other actions are allowed on the Certificate Store.



System BIOS

System BIOS Settings • System Security

UEFI Variable Access ● Standard ○ Controlled

Secure ME PCI Cfg Space ○ Enabled ● Disabled

SECURE BOOT

NOTE: Secure Boot cannot be enabled while a platform key (PK) is not present.

Secure Boot ○ Enabled ● Disabled

Secure Boot Policy ● Standard ○ Custom

WARNING: Changing the Secure Boot Policy to Standard causes the system to replace all custom certificates and hashes with default certificates and hashes on the next boot.

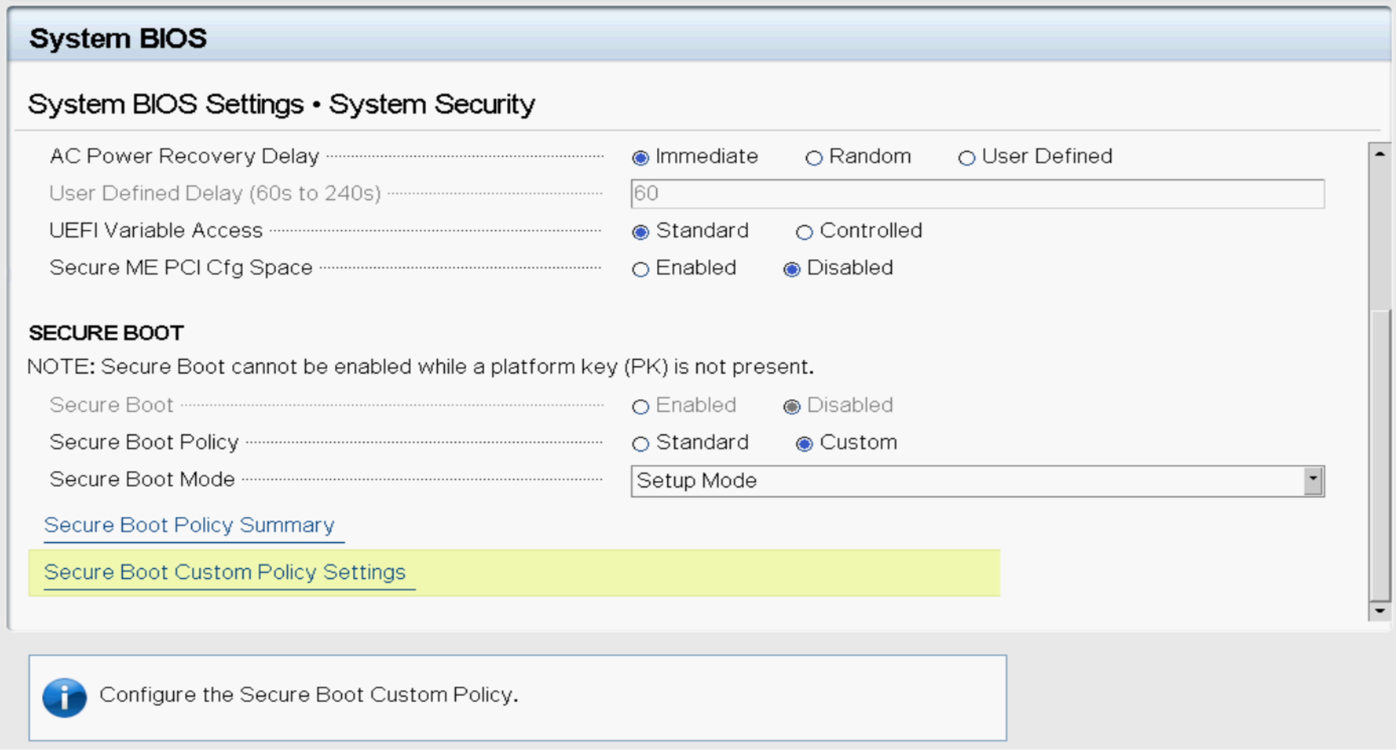
Secure Boot Mode: Setup Mode

Secure Boot Policy Summary

Secure Boot Custom Policy Settings

View the list of certificates and hashes that Secure Boot uses to authenticate images.

Figure 11. Standard Secure Boot policy



System BIOS

System BIOS Settings • System Security

AC Power Recovery Delay ● Immediate ○ Random ○ User Defined

User Defined Delay (60s to 240s) 60

UEFI Variable Access ● Standard ○ Controlled

Secure ME PCI Cfg Space ○ Enabled ● Disabled

SECURE BOOT

NOTE: Secure Boot cannot be enabled while a platform key (PK) is not present.

Secure Boot ○ Enabled ● Disabled

Secure Boot Policy ○ Standard ● Custom

Secure Boot Mode Setup Mode

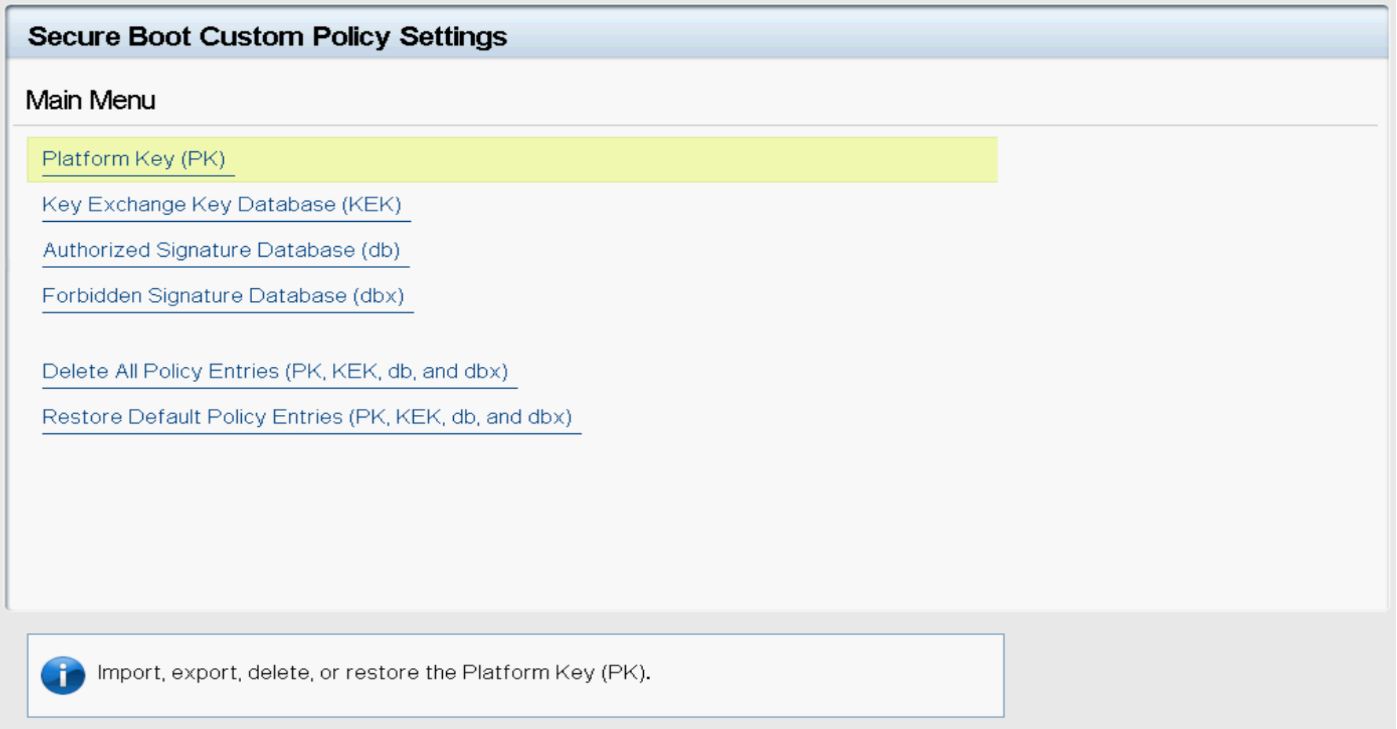
Secure Boot Policy Summary

Secure Boot Custom Policy Settings

Configure the Secure Boot Custom Policy.

Figure 12. Custom Secure Boot policy

Configuring the Secure Boot Policy to Custom enables the options to manage the certificate store through various actions such as View, Export, Import, Delete, Delete All, Reset, and Rest All on PK, KEK, DB and DBX. User can select the policy (PK / KEK / DB / DBX) on which you want to make the change and perform appropriate actions.



Secure Boot Custom Policy Settings

Main Menu

Platform Key (PK)

Key Exchange Key Database (KEK)

Authorized Signature Database (db)

Forbidden Signature Database (dbx)

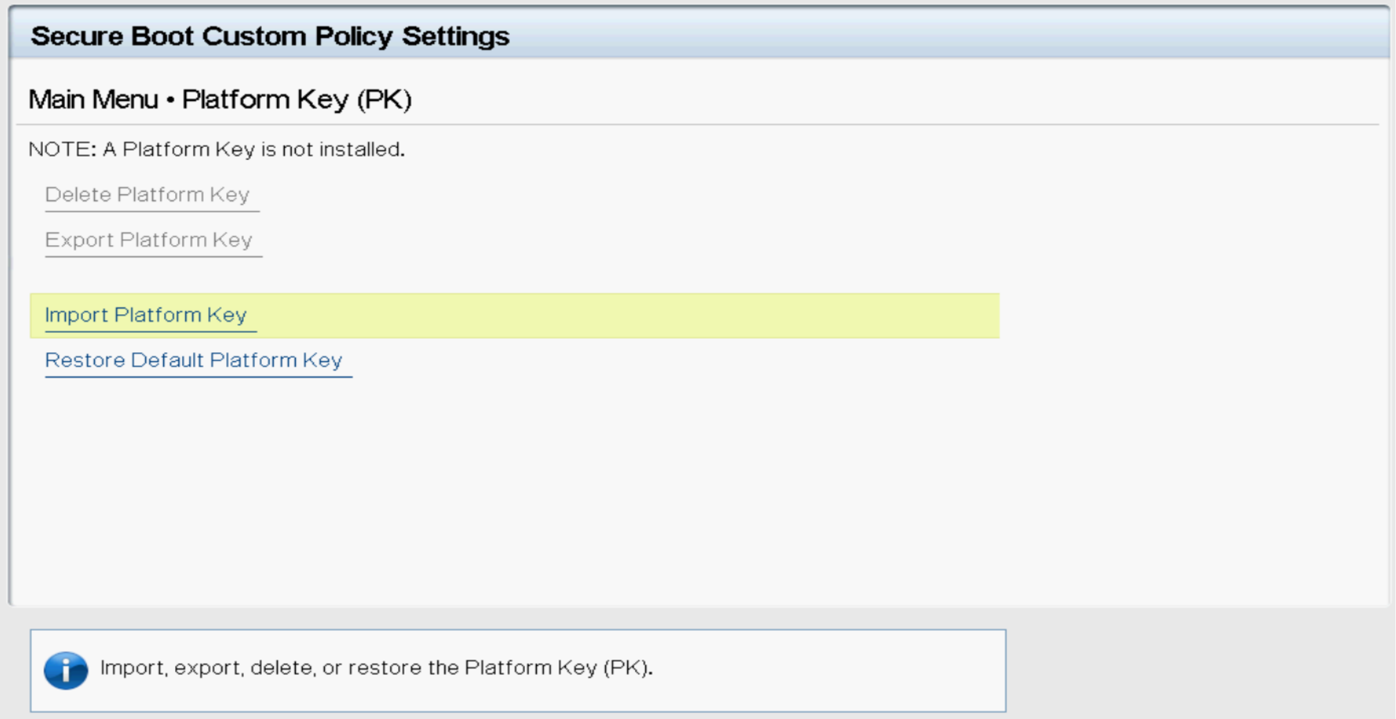
Delete All Policy Entries (PK, KEK, db, and dbx)

Restore Default Policy Entries (PK, KEK, db, and dbx)

Import, export, delete, or restore the Platform Key (PK).

Figure 13. Secure Boot Custom Policy Settings-Main Menu

Each section will have links to perform Import, Export, Delete, and Reset operations. Links are enabled based on what is applicable based on the configuration at the point of time. For example, in the screen shot, Delete and Export operations are disabled because there is no PK configured yet.



Secure Boot Custom Policy Settings

Main Menu • Platform Key (PK)

NOTE: A Platform Key is not installed.

Delete Platform Key

Export Platform Key

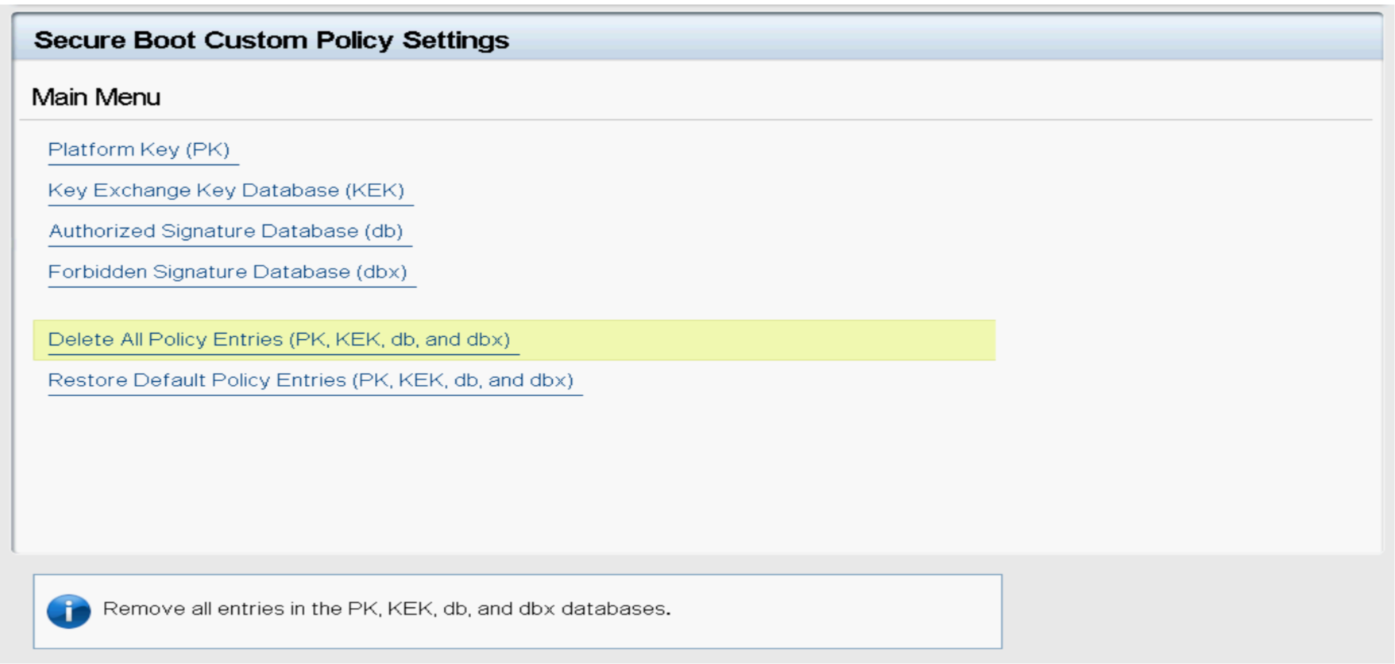
Import Platform Key

Restore Default Platform Key

Import, export, delete, or restore the Platform Key (PK).

Figure 14. Export and Delete Platform Key is disabled

*Delete All* and *Reset All* are the operations that have impact on all the policies. *Delete All* deletes all the certificates and image digests in the Custom policy, and *Reset All* restores all the certificates and image digests from Standard / *Default* certificate store.


Secure Boot Custom Policy Settings
Main Menu
Platform Key (PK)
Key Exchange Key Database (KEK)
Authorized Signature Database (db)
Forbidden Signature Database (dbx)
Delete All Policy Entries (PK, KEK, db, and dbx)
Restore Default Policy Entries (PK, KEK, db, and dbx)

Remove all entries in the PK, KEK, db, and dbx databases.

**Figure 15. Delete All Policy Entries**

## *Secure Boot certificate management using RACADM*

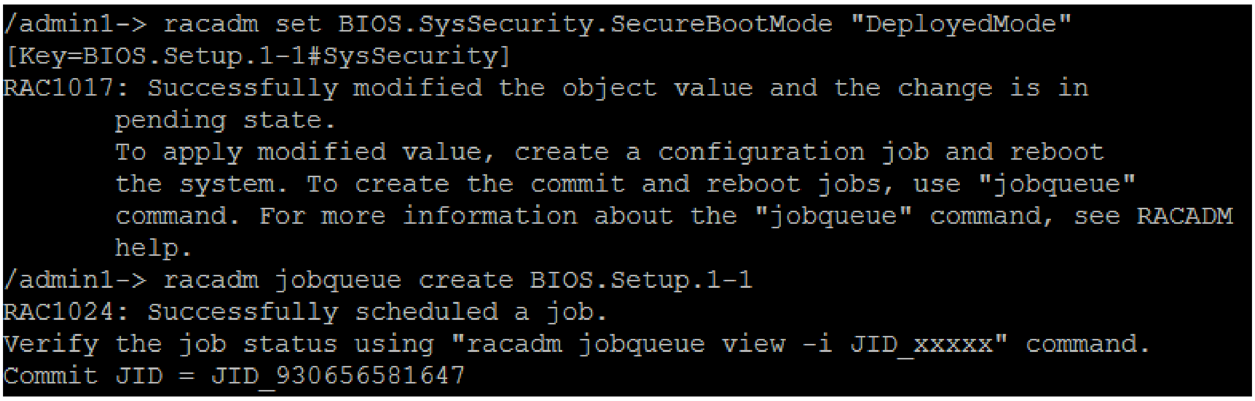
RACADM supports Secure Boot certificate management by using a new command **`bioscert`**. Following is the list of operations supported by using `bioscret`.

| Secure boot certificate management operations | Role/Privilege required for<br>iDRAC Users | Value/Setting required for<br>“SecureBootPolicy” attribute |
|---|---|---|
| **View** | Login | `Standard/Custom` |
| **Export** | Login | `Custom` |
| **Import** | Login & System/Server Control | `Custom` |
| **Restore** | Login & System/Server Control | `Custom` |
| **Delete** | Login & System/Server Control | `Custom` |

RACADM supports the following Secure Boot attribute configurations:

| Attributes | Legal Values | Default Value | RACADM command to read value |
|---|---|---|---|
| **`SecureBoot`** | `Enabled,`<br>`Disabled` | `Disabled` | `racadm get BIOS.SysSecurity.SecureBoot` |
| **`SecureBootPolicy`** | `Standard,`<br>`Custom` | `Standard` | `racadm get BIOS.SysSecurity.SecureBootPolicy` |
| **`SecureBootMode`** | `UserMode,`<br>`SetupMode, AuditMode,`<br>`DeployedMode` | `UserMode` | `racadm get BIOS.SysSecurity.SecureBootMode` |

To modify settings of the above attributes, `racadm set` can be used as shown below. After modification, change goes to pending state. Therefore, to apply the modified value, a configuration job must be created with help of `racadm jobqueue create BIOS.Setup.1-1` and then an iDRAC restart.



```
/admin1-> racadm set BIOS.SysSecurity.SecureBootMode "DeployedMode"
[Key=BIOS.Setup.1-1#SysSecurity]
RAC1017: Successfully modified the object value and the change is in
       pending state.
       To apply modified value, create a configuration job and reboot
       the system. To create the commit and reboot jobs, use "jobqueue"
       command. For more information about the "jobqueue" command, see RACADM
       help.
/admin1-> racadm jobqueue create BIOS.Setup.1-1
RAC1024: Successfully scheduled a job.
Verify the job status using "racadm jobqueue view -i JID_xxxxx" command.
Commit JID = JID_930656581647
```

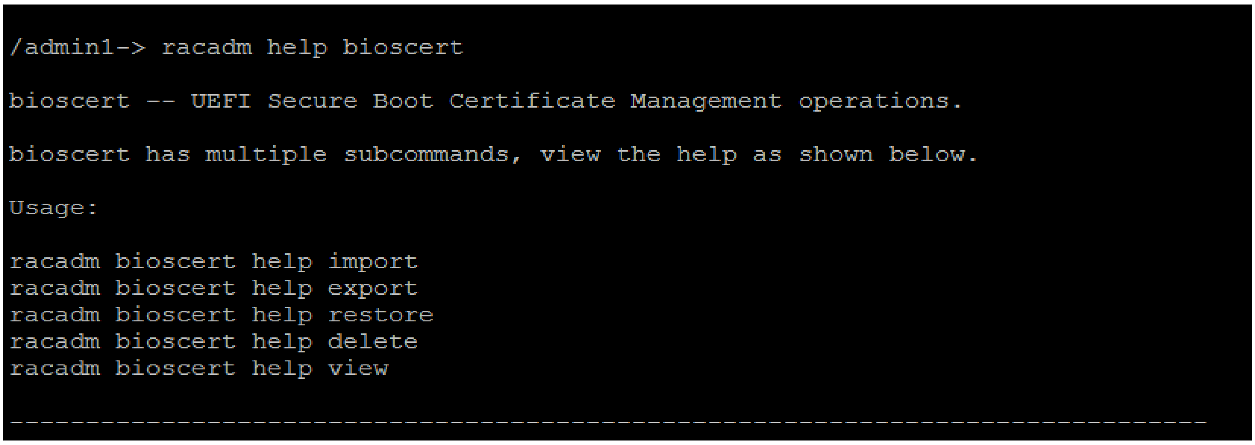
*Firmware and remote racadm allow certificate management operations irrespective of the SecureBootMode settings. However, Local RACADM (Inband tool) does not allow Secure Boot certificate management operations when “SecureBootMode” is set to DeployedMode.*

Currently, the Secure Boot Certificate management by using RACADM is not a licensed feature. Therefore, RACADM allows certificate management operations irrespective of the license installed on your iDRAC.

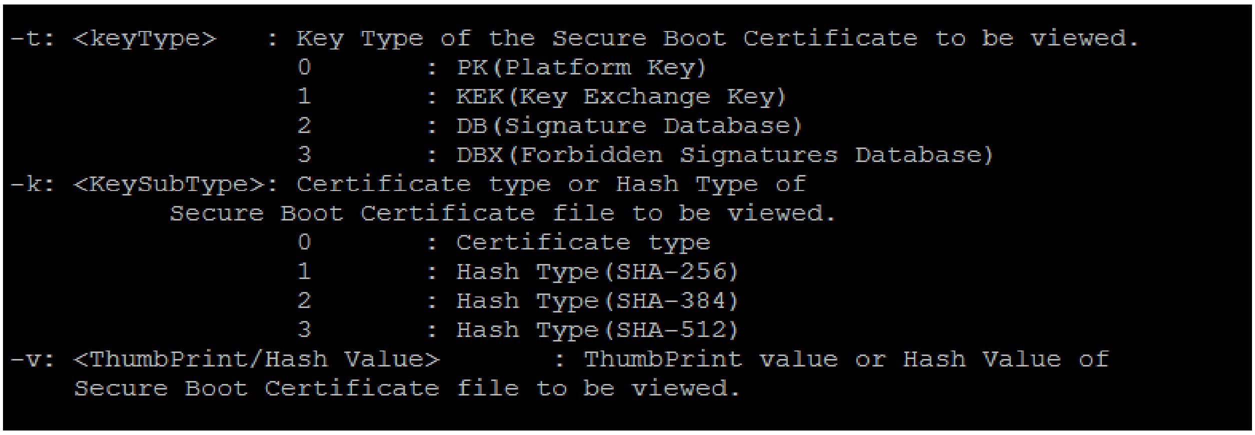
## Certificate management operations

Allowed operations and their syntaxes can be retrieved by running `racadm help bioscert` as shown in the screen shot. Generic syntax of `bioscert` operations which act on individual certificate or hash is:

```
racadm bioscert <sub-command/operation> -t <KeyType> -k <KeySubType> -v <Hashvalue or
Thumbprintvalue>
```



```
/admin1-> racadm help bioscert

bioscert -- UEFI Secure Boot Certificate Management operations.

bioscert has multiple subcommands, view the help as shown below.

Usage:

racadm bioscert help import
racadm bioscert help export
racadm bioscert help restore
racadm bioscert help delete
racadm bioscert help view

-------------------------------------------------------------------------------
```



```
-t: <keyType>    : Key Type of the Secure Boot Certificate to be viewed.
                   0        : PK(Platform Key)
                   1        : KEK(Key Exchange Key)
                   2        : DB(Signature Database)
                   3        : DBX(Forbidden Signatures Database)
-k: <KeySubType>: Certificate type or Hash Type of
          Secure Boot Certificate file to be viewed.
                   0        : Certificate type
                   1        : Hash Type(SHA-256)
                   2        : Hash Type(SHA-384)
                   3        : Hash Type(SHA-512)
-v: <ThumbPrint/Hash Value>        : ThumbPrint value or Hash Value of
    Secure Boot Certificate file to be viewed.
```

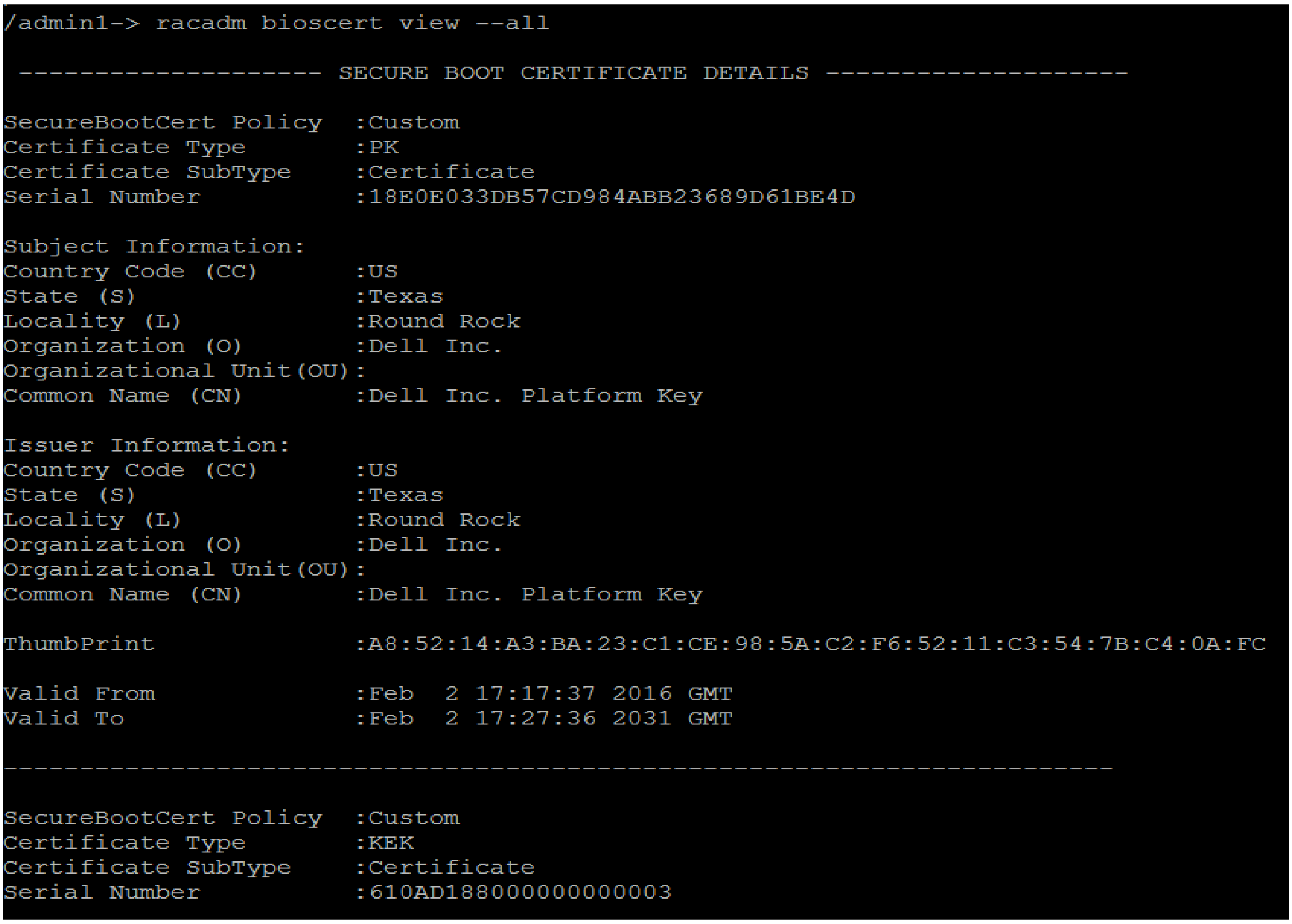
## Bioscert View Operation

Based on the “SecureBootPolicy” settings, it retrieves data from respective certificate store and displays. If the request is to view a certificate record, the output will list details of the certificate attributes such as subject information, issuer details, valid from, valid to, and thumb print. If the request is to view a hash record then the output lists the details of the record along with the hash value of the record.

**`racadm bioscert view --all`**

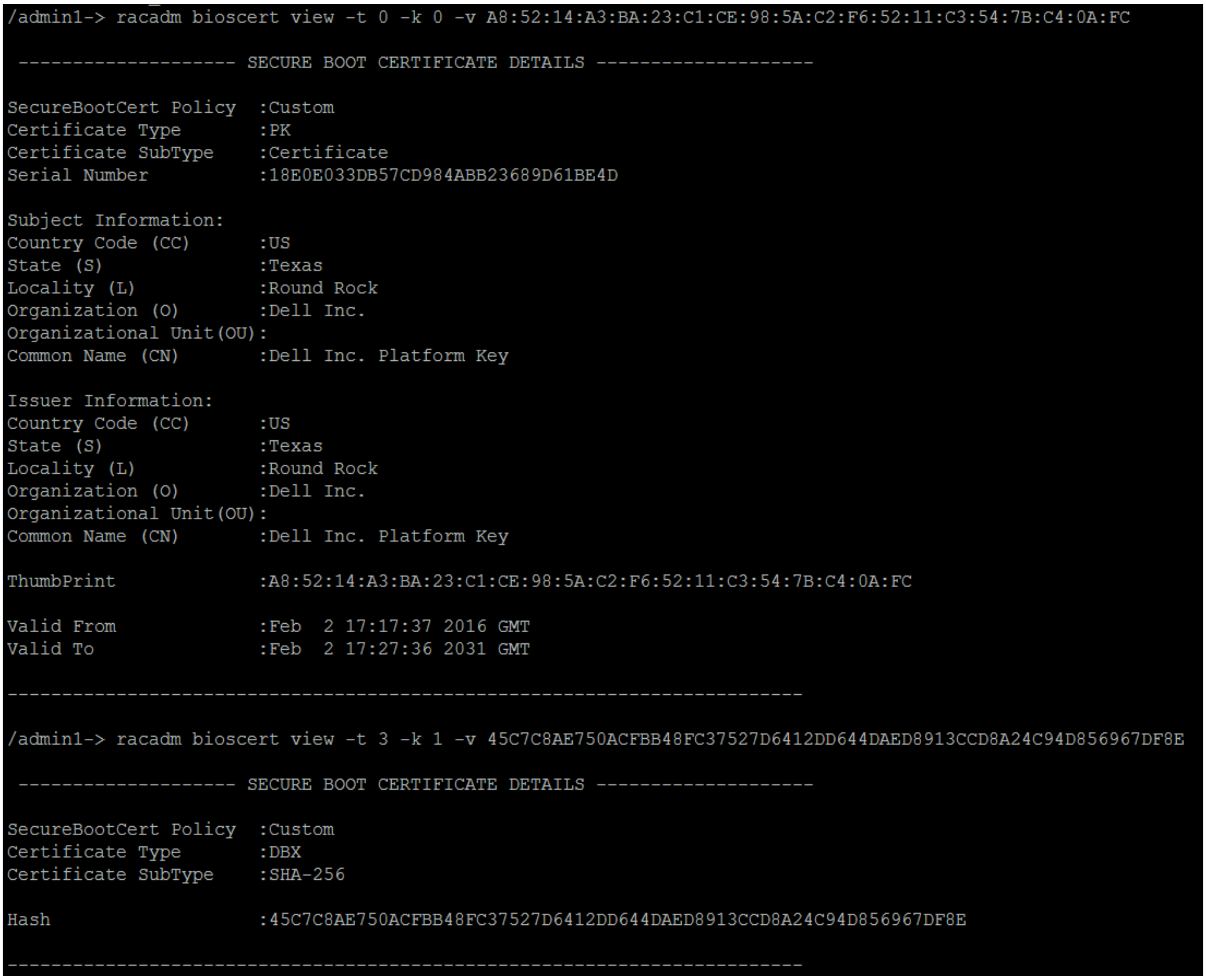
**`racadm bioscert view -t <keyType> -k <KeySubType> -v <HashValue or ThumbPrintValue>`**

Examples are shown in the screen shot here:



```
/admin1-> racadm bioscert view --all

 -------------------- SECURE BOOT CERTIFICATE DETAILS --------------------

SecureBootCert Policy  :Custom
Certificate Type       :PK
Certificate SubType    :Certificate
Serial Number          :18E0E033DB57CD984ABB23689D61BE4D

Subject Information:
Country Code (CC)      :US
State (S)              :Texas
Locality (L)           :Round Rock
Organization (O)       :Dell Inc.
Organizational Unit(OU):
Common Name (CN)       :Dell Inc. Platform Key

Issuer Information:
Country Code (CC)      :US
State (S)              :Texas
Locality (L)           :Round Rock
Organization (O)       :Dell Inc.
Organizational Unit(OU):
Common Name (CN)       :Dell Inc. Platform Key

ThumbPrint             :A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC

Valid From             :Feb  2 17:17:37 2016 GMT
Valid To               :Feb  2 17:27:36 2031 GMT

-------------------------------------------------------------------------

SecureBootCert Policy  :Custom
Certificate Type       :KEK
Certificate SubType    :Certificate
Serial Number          :610AD188000000000003
```

The `View --all` command lists all the certificates and image digests present on the system at that point in time. If you want to view information about a specific record (certificate or image digest) then the command must specify the certificate type, subtype, and identifier of the record as shown in the screen shot here:



```
/admin1-> racadm bioscert view -t 0 -k 0 -v A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC

 -------------------- SECURE BOOT CERTIFICATE DETAILS --------------------

SecureBootCert Policy  :Custom
Certificate Type       :PK
Certificate SubType    :Certificate
Serial Number          :18E0E033DB57CD984ABB23689D61BE4D

Subject Information:
Country Code (CC)      :US
State (S)              :Texas
Locality (L)           :Round Rock
Organization (O)       :Dell Inc.
Organizational Unit(OU):
Common Name (CN)       :Dell Inc. Platform Key

Issuer Information:
Country Code (CC)      :US
State (S)              :Texas
Locality (L)           :Round Rock
Organization (O)       :Dell Inc.
Organizational Unit(OU):
Common Name (CN)       :Dell Inc. Platform Key

ThumbPrint             :A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC

Valid From             :Feb  2 17:17:37 2016 GMT
Valid To               :Feb  2 17:27:36 2031 GMT

-------------------------------------------------------------------------

/admin1-> racadm bioscert view -t 3 -k 1 -v 45C7C8AE750ACFBB48FC37527D6412DD644DAED8913CCD8A24C94D856967DF8E

 -------------------- SECURE BOOT CERTIFICATE DETAILS --------------------

SecureBootCert Policy  :Custom
Certificate Type       :DBX
Certificate SubType    :SHA-256

Hash                   :45C7C8AE750ACFBB48FC37527D6412DD644DAED8913CCD8A24C94D856967DF8E

-------------------------------------------------------------------------
```

## Bioscert Export Operation

Exports the Secure Boot Certificate to a remote share (CIFS, NFS, HTTP, and HTTPS) or local share:

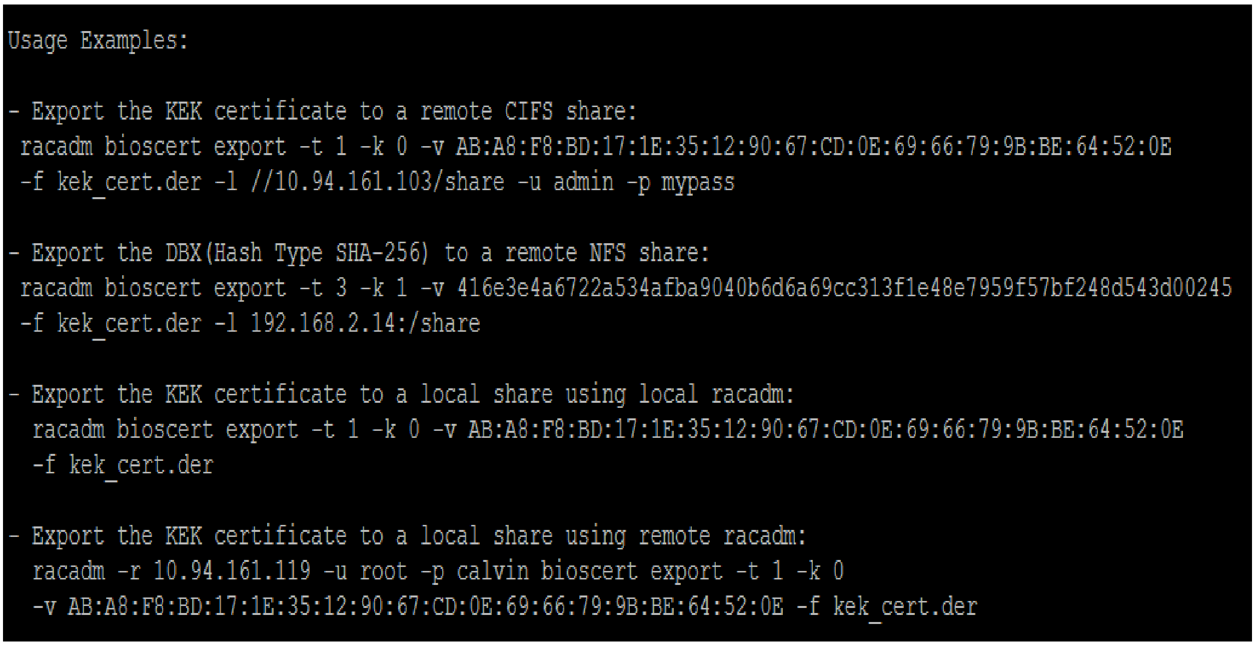
```
racadm bioscert export -t <keyType> -k <KeySubType> -v <HashValue or ThumbPrintValue> -f
<filename> [-l <CIFS/NFS/HTTP/HTTPS share path>] [-u <username>] [-p <password>]
```

## Example

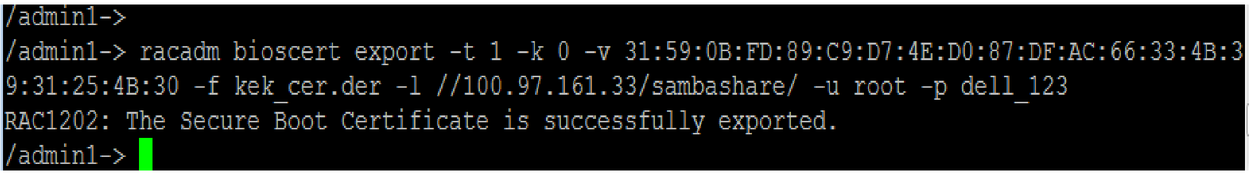
### *Export DB key to CIFS share by using the local or firmware RACADM*

```
$ racadm bioscert export -t 1 -k 0 -v
31:59:0B:FD:89:C9:D7:4E:D0:87:DF:AC:66:33:4B:39:31:25:4B:30 -f kek_cer.der -l
//100.97.161.33/sambashare/ -u root -p dell_123
```

*The Event and Error Message RAC1202: The Secure Boot Certificate is successfully exported.*



```
Usage Examples:

- Export the KEK certificate to a remote CIFS share:
 racadm bioscert export -t 1 -k 0 -v AB:A8:F8:BD:17:1E:35:12:90:67:CD:0E:69:66:79:9B:BE:64:52:0E
 -f kek_cert.der -l //10.94.161.103/share -u admin -p mypass

- Export the DBX(Hash Type SHA-256) to a remote NFS share:
 racadm bioscert export -t 3 -k 1 -v 416e3e4a6722a534afba9040b6d6a69cc313f1e48e7959f57bf248d543d00245
 -f kek_cert.der -l 192.168.2.14:/share

- Export the KEK certificate to a local share using local racadm:
  racadm bioscert export -t 1 -k 0 -v AB:A8:F8:BD:17:1E:35:12:90:67:CD:0E:69:66:79:9B:BE:64:52:0E
  -f kek_cert.der

- Export the KEK certificate to a local share using remote racadm:
  racadm -r 10.94.161.119 -u root -p calvin bioscert export -t 1 -k 0
  -v AB:A8:F8:BD:17:1E:35:12:90:67:CD:0E:69:66:79:9B:BE:64:52:0E -f kek_cert.der
```



```
/admin1->
/admin1-> racadm bioscert export -t 1 -k 0 -v 31:59:0B:FD:89:C9:D7:4E:D0:87:DF:AC:66:33:4B:3
9:31:25:4B:30 -f kek_cer.der -l //100.97.161.33/sambashare/ -u root -p dell_123
RAC1202: The Secure Boot Certificate is successfully exported.
/admin1->
```

**Note**:

- In case "SecureBootPolicy" is set to `Standard`, RACADM will not allow export operation, but the following Event and Error Message is displayed: `RAC1212: Unable to complete the operation because the Secure Boot policy is set to Standard.`
- If the remote share does not have enough space during export, the following Event and Error Message is displayed: `RAC1219: Unable to export the Secure Boot certificate data to remote share because of insufficient storage space.`
- Local file share support is allowed only from Remote and Local RACADM.
- To know more about export command, enter the `racadm bioscert help export` command at the RACADM CLI.

## Bioscert Import Operation

This feature enables you to import a Secure Boot Certificate to iDRAC from remote share (CIFS, NFS, HTTP, and HTTPS) or local share. If you want to enroll a certificate to authenticate a firmware or driver or Option ROM, which is likely to get executed during the POST, the Import feature enables you to add the certificate to the iDRAC Secure Boot certificate store. Subcommand to be used for import operation is “import”. The command must provide details of certificate type, subtype, path to the file to import, and share details.

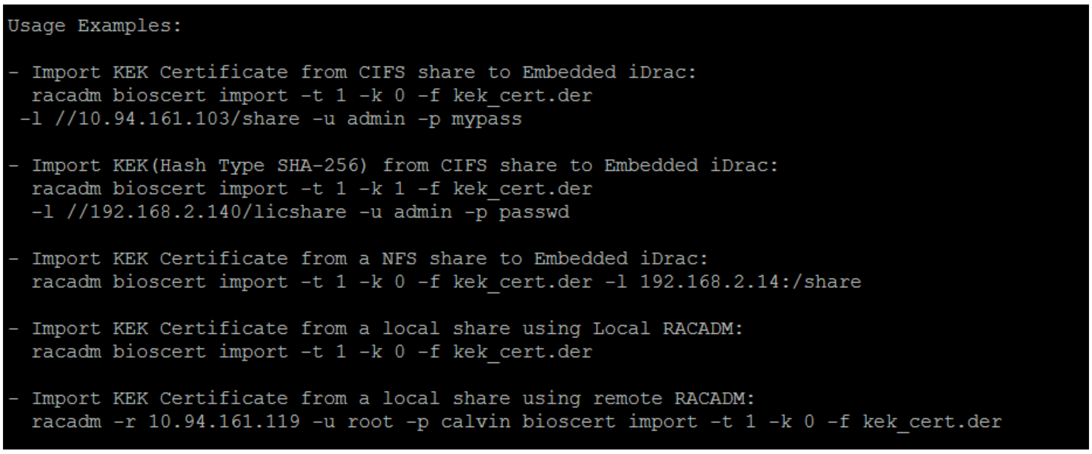
```
racadm bioscert import -t <keyType> -k <KeySubType> -f <filename> [-l <CIFS/NFS/HTTP/HTTPS
share path>] [-u <username>] [-p <password>]
```

### Example

*Import KEK Key from CIFS share using local RACADM*

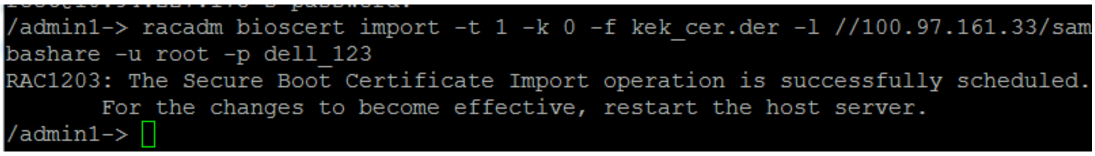
```
$ racadm bioscert import -t 1 -k 0 -f kek_cer.der -l //100.97.161.33/sambashare -u root -p
dell_123
```

> *The Event and Error Message RAC1203: The Secure Boot Certificate Import operation is successfully scheduled. For the changes to become effective, restart the host server.*



```
Usage Examples:

- Import KEK Certificate from CIFS share to Embedded iDrac:
  racadm bioscert import -t 1 -k 0 -f kek_cert.der
 -l //10.94.161.103/share -u admin -p mypass

- Import KEK(Hash Type SHA-256) from CIFS share to Embedded iDrac:
  racadm bioscert import -t 1 -k 1 -f kek_cert.der
  -l //192.168.2.140/licshare -u admin -p passwd

- Import KEK Certificate from a NFS share to Embedded iDrac:
  racadm bioscert import -t 1 -k 0 -f kek_cert.der -l 192.168.2.14:/share

- Import KEK Certificate from a local share using Local RACADM:
  racadm bioscert import -t 1 -k 0 -f kek_cert.der

- Import KEK Certificate from a local share using remote RACADM:
  racadm -r 10.94.161.119 -u root -p calvin bioscert import -t 1 -k 0 -f kek_cert.der
```

After the import request is successfully serviced, a pending task is added to the pending list in iDRAC which gets serviced during the next restart of the host server. Therefore, to apply the changes of requested Secure Boot certificate management operations, the host server must be restarted.



```
/admin1-> racadm bioscert import -t 1 -k 0 -f kek_cer.der -l //100.97.161.33/sam
bashare -u root -p dell_123
RAC1203: The Secure Boot Certificate Import operation is successfully scheduled.
        For the changes to become effective, restart the host server.
/admin1->
```

**Note**

- If “SecureBootPolicy” is set to `Standard`, RACADM will not allow export operation, but the following Event and Error Message is displayed: `RAC1212: Unable to complete the operation because the Secure Boot policy is set to Standard.`
- If “SystemLockDownMode” is enabled, RACADM does not allow import operation and the following Event and Error Message is displayed: `RAC1201: Unable to complete the operation because the server is in the lockdown mode.` “SystemLockDownMode” must be disabled before trying the import operation.
- After importing a new PK, if a PK already exists, RACADM displays the following Event and Error Message: `RAC1213: Unable to import the Public Key (PK) because a PK already exists.` Therefore, before importing the new PK, the existing PK must be deleted.
- Local file share support is allowed only from Remote and Local RACADM.
- To know more about export command, enter the `racadm bioscert help import` command at the RACADM CLI.

## Bioscert Delete Operation

The *Delete* operation deletes the installed Secure Boot certificate or an image digest in the iDRAC Secure Boot certificate store. *Delete* operation is applicable when you want to delete one or more of the standard certificates or image digests, and enroll your own customized certificates or image digests. *Delete* command must provide the record type, subtype, and identifier (Thumb print or hash) of the record.

```
racadm bioscert delete --all

racadm bioscert delete -t <keyType> -k <KeySubType> -v <HashValue or ThumbPrintValue>
```

### Example

*To delete an installed DBX Secure Boot Certificate of HASH type SHA-256*

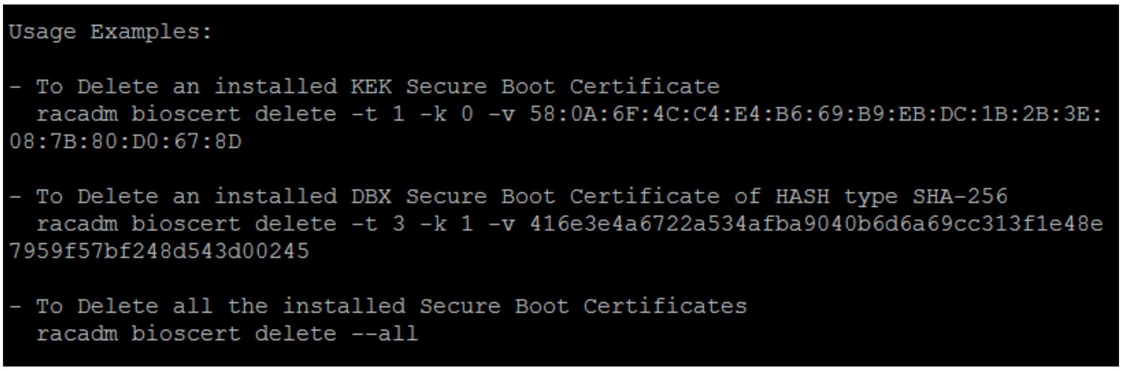
```
$ racadm bioscert delete -t 3 -k 1 -v
416e3e4a6722a534afba9040b6d6a69cc313f1e48e7959f57bf248d543d00245
```

> *The Event and Error Message RAC1204: The Secure Boot Certificate Delete operation is successfully scheduled. For the changes to become effective, restart the host server.*

## DeleteAll certificates

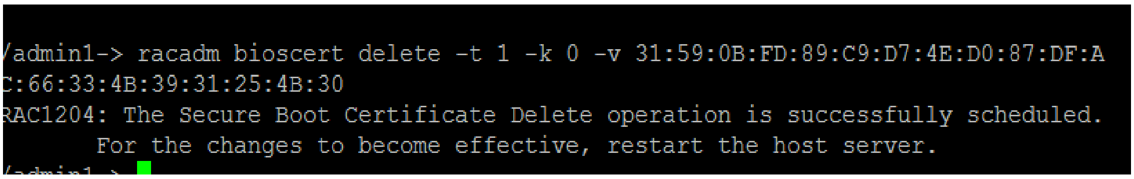
```
racadm bioscert Delete --all
```

> *The Event and Error Message RAC1206: The Secure Boot Certificate DeleteAll operation is successfully scheduled. For the changes to become effective, restart the host server.*



```
Usage Examples:

- To Delete an installed KEK Secure Boot Certificate
  racadm bioscert delete -t 1 -k 0 -v 58:0A:6F:4C:C4:E4:B6:69:B9:EB:DC:1B:2B:3E:
08:7B:80:D0:67:8D

- To Delete an installed DBX Secure Boot Certificate of HASH type SHA-256
  racadm bioscert delete -t 3 -k 1 -v 416e3e4a6722a534afba9040b6d6a69cc313f1e48e
7959f57bf248d543d00245

- To Delete all the installed Secure Boot Certificates
  racadm bioscert delete --all
```

After the *Delete* request is successfully serviced, a pending task is added to the pending list in iDRAC which gets serviced during the next restart of the host server. Therefore, to apply the changes of requested Secure Boot certificate management operations, the host server must be restarted.



```
/admin1-> racadm bioscert delete -t 1 -k 0 -v 31:59:0B:FD:89:C9:D7:4E:D0:87:DF:A
C:66:33:4B:39:31:25:4B:30
RAC1204: The Secure Boot Certificate Delete operation is successfully scheduled.
       For the changes to become effective, restart the host server.
/admin1->
```

**Note**:

- If “SecureBootPolicy” is set to `Standard`, RACADM will not allow export operation, but the following Event and Error Message is displayed: `RAC1212: Unable to complete the operation because the Secure Boot policy is set to Standard.`
- If “SystemLockDownMode” is enabled, RACADM does not allow import operation and the following Event and Error Message is displayed: `RAC1201: Unable to complete the operation because the server is in the lockdown mode.` “SystemLockDownMode” must be disabled before trying the import operation.
- To know more about export command, enter the `racadm bioscert help delete` command at the RACADM CLI.

## Bioscert Restore Operation

The *Restore* operation resets the installed custom certificates to default standard certificates. Restore operation helps you to undo the changes you made on certificate store by replacing the customized certificates with standard default certificates. Restore operations can be performed on certificate store based on section (PK/KEK/DB/DBX) or as a whole.

> *RACADM does not support individual certificate Hash restore.*

```
racadm bioscert restore -all

racadm bioscert restore -t <keyType>
```

### Example

*Restore DB section*

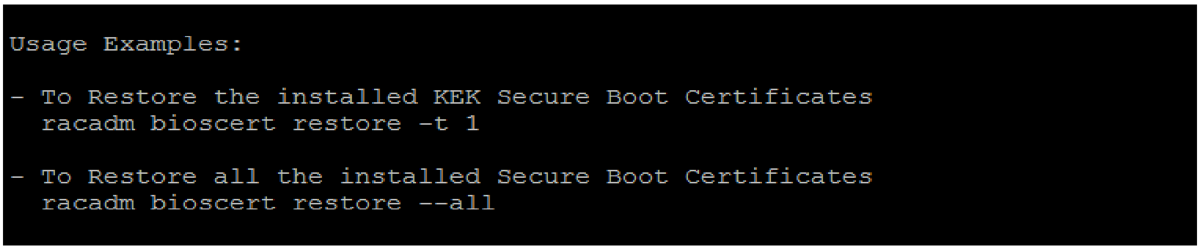
racadm bioscert restore –t 2

> *The Event and Error Message RAC1205: The Secure Boot Certificate Restore operation is successfully scheduled. For the changes to become effective, restart the host server.*

## RestoreAll certificates

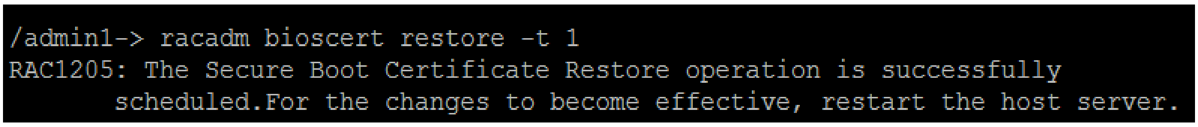
```
racadm bioscert restore -all
```

*The Event and Error Message RAC1207: The Secure Boot Certificate RestoreAll operation is successfully scheduled. For the changes to become effective, restart the host server.*



```
Usage Examples:

- To Restore the installed KEK Secure Boot Certificates
  racadm bioscert restore -t 1

- To Restore all the installed Secure Boot Certificates
  racadm bioscert restore --all
```

After the restore request is successfully serviced, a pending task is added to the pending list in iDRAC which gets serviced during the next restart of the host server. Therefore, to apply the changes of requested Secure Boot certificate management operations, the host machine should be rebooted.



```
/admin1-> racadm bioscert restore -t 1
RAC1205: The Secure Boot Certificate Restore operation is successfully
       scheduled.For the changes to become effective, restart the host server.
```

**Note**:

- If “SecureBootPolicy” is set to `Standard`, RACADM will not allow export operation, but the following Event and Error Message is displayed: `RAC1212: Unable to complete the operation because the Secure Boot policy is set to Standard.`
- If “SystemLockDownMode” is enabled, RACADM does not allow import operation and the following Event and Error Message is displayed: `RAC1201: Unable to complete the operation because the server is in the lockdown mode.` “SystemLockDownMode” must be disabled before trying the import operation.

To know more about export command, enter the `racadm bioscert help restore` command at the RACADM CLI.

# Secure Boot certificate management using WS-Man

WS-Man (Web Services-Management) is a DMTF open standard that defines a SOAP-XML based protocol for the management of servers, devices, applications used by Systems Management consoles or management applications.

## *Enabling or disabling the Secure Boot feature?*

To enable or disable the Secure Boot, management applications can use the `DCIM_BIOSService.SetAttribute()` method from the Dell_BIOSandBOOT management profile to set the **`BIOS.Setup.1-1:SecureBoot`** attribute and create a job by using `DCIM_BIOSService.CreateTargetedConfigJob()` method to execute this change. This enumeration attribute can be set to Enabled or Disabled.

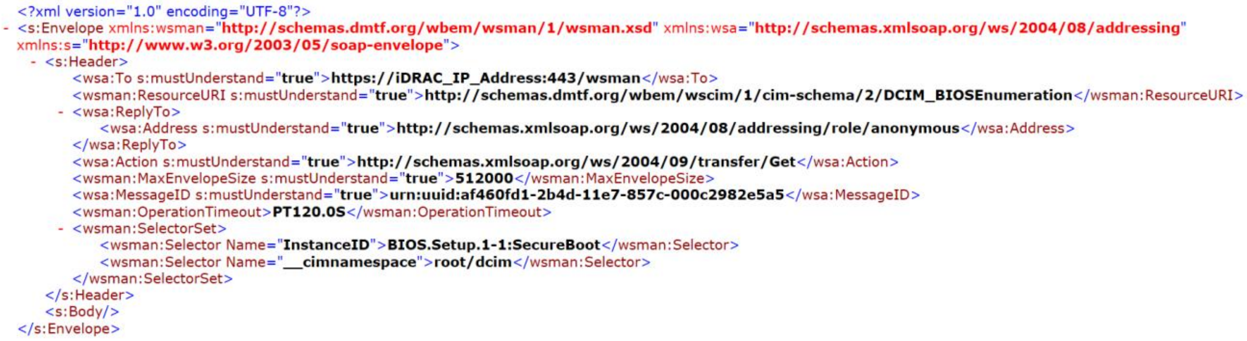
For more information, see the Dell_BIOSandBOOTManagementProfile_4.0.0 Session 8.1 and 8.4 in the WS-Man Profile Document available at http://en.community.dell.com/techcenter/systems-management/w/wiki/1906.dcim-library-profile.

By default, Secure Boot is configured as Disabled. Therefore, it must be changed to Enabled to activate the Secure Boot feature.

Here is an example WS-Man workflow by using SOAP to set the `BIOS.Setup.1-1:SecureBoot` attribute. In this workflow, the current value of `BIOS.Setup.1-1:SecureBoot` will be checked, a job created to set a new value, job status queried to verify success, and finally verified that the value of `BIOS.Setup.1-1:SecureBoot` has changed.

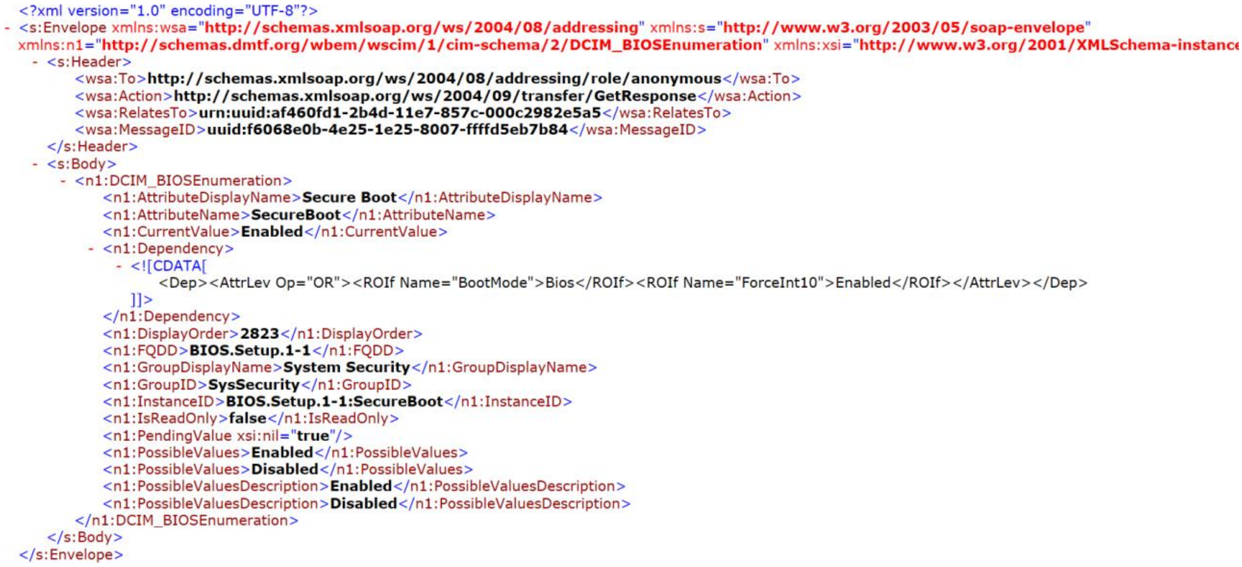
1. Management application sends the `get` request for the following form to check the current value of the `BIOS.Setup.1-1:SecureBoot`.

SOAP Request for GET BIOS.Setup.1-1:SecureBoot attribute



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSEnumeration</wsman:ResourceURI>
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:af460fd1-2b4d-11e7-857c-000c2982e5a5</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="InstanceID">BIOS.Setup.1-1:SecureBoot</wsman:Selector>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
    <s:Body/>
  </s:Envelope>
```

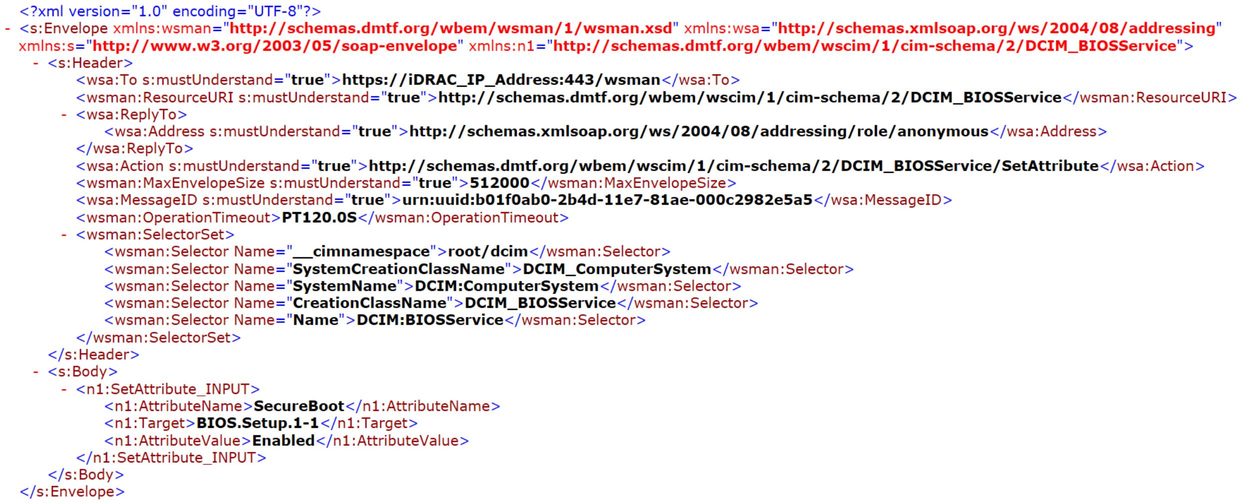
SOAP Response for GET BIOS.Setup.1-1:SecureBoot attribute



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSEnumeration" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:af460fd1-2b4d-11e7-857c-000c2982e5a5</wsa:RelatesTo>
      <wsa:MessageID>uuid:f6068e0b-4e25-1e25-8007-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:DCIM_BIOSEnumeration>
        <n1:AttributeDisplayName>Secure Boot</n1:AttributeDisplayName>
        <n1:AttributeName>SecureBoot</n1:AttributeName>
        <n1:CurrentValue>Enabled</n1:CurrentValue>
      - <n1:Dependency>
        - <![CDATA[
              <Dep><AttrLev Op="OR"><ROIf Name="BootMode">Bios</ROIf><ROIf Name="ForceInt10">Enabled</ROIf></AttrLev></Dep>
          ]]>
        </n1:Dependency>
        <n1:DisplayOrder>2823</n1:DisplayOrder>
        <n1:FQDD>BIOS.Setup.1-1</n1:FQDD>
        <n1:GroupDisplayName>System Security</n1:GroupDisplayName>
        <n1:GroupID>SysSecurity</n1:GroupID>
        <n1:InstanceID>BIOS.Setup.1-1:SecureBoot</n1:InstanceID>
        <n1:IsReadOnly>false</n1:IsReadOnly>
        <n1:PendingValue xsi:nil="true"/>
        <n1:PossibleValues>Enabled</n1:PossibleValues>
        <n1:PossibleValues>Disabled</n1:PossibleValues>
        <n1:PossibleValuesDescription>Enabled</n1:PossibleValuesDescription>
        <n1:PossibleValuesDescription>Disabled</n1:PossibleValuesDescription>
      </n1:DCIM_BIOSEnumeration>
    </s:Body>
  </s:Envelope>
```

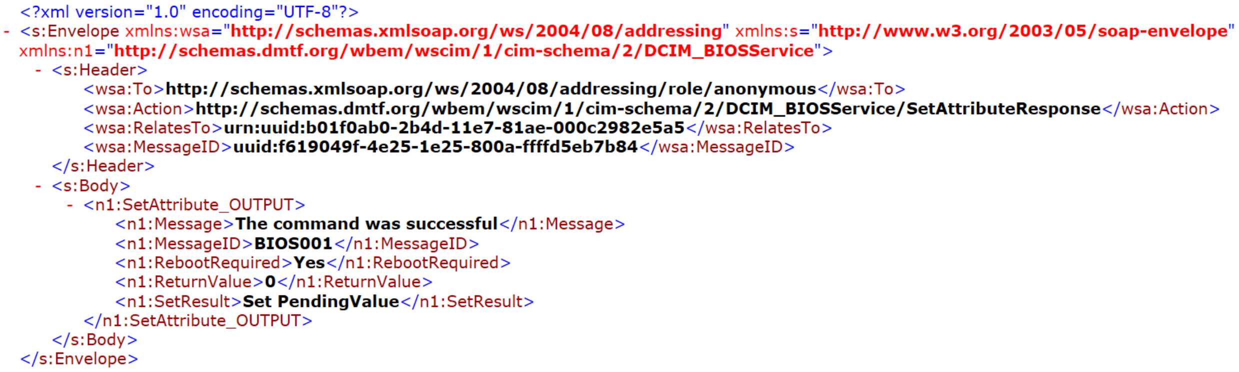
2. Management application sends the request for setting the `BIOS.Setup.1-1:SecureBoot` attribute to pending value.

SOAP Request for DCIM_BIOSService.SetAttribute() to set BIOS.Setup.1-1:SecureBoot attribute



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService</wsman:ResourceURI>
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService/SetAttribute</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:b01f0ab0-2b4d-11e7-81ae-000c2982e5a5</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        <wsman:Selector Name="SystemCreationClassName">DCIM_ComputerSystem</wsman:Selector>
        <wsman:Selector Name="SystemName">DCIM:ComputerSystem</wsman:Selector>
        <wsman:Selector Name="CreationClassName">DCIM_BIOSService</wsman:Selector>
        <wsman:Selector Name="Name">DCIM:BIOSService</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
    - <n1:SetAttribute_INPUT>
        <n1:AttributeName>SecureBoot</n1:AttributeName>
        <n1:Target>BIOS.Setup.1-1</n1:Target>
        <n1:AttributeValue>Enabled</n1:AttributeValue>
      </n1:SetAttribute_INPUT>
    </s:Body>
  </s:Envelope>
```

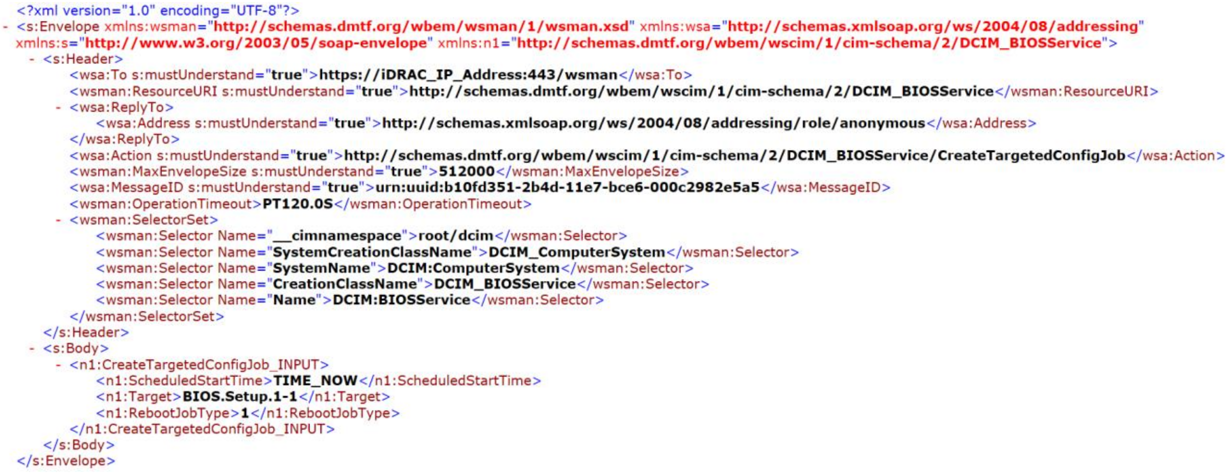
SOAP Response for DCIM_BIOSService.SetAttribute() to set BIOS.Setup.1-1:SecureBoot attribute



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService">
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService/SetAttributeResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:b01f0ab0-2b4d-11e7-81ae-000c2982e5a5</wsa:RelatesTo>
      <wsa:MessageID>uuid:f619049f-4e25-1e25-800a-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:SetAttribute_OUTPUT>
        <n1:Message>The command was successful</n1:Message>
        <n1:MessageID>BIOS001</n1:MessageID>
        <n1:RebootRequired>Yes</n1:RebootRequired>
        <n1:ReturnValue>0</n1:ReturnValue>
        <n1:SetResult>Set PendingValue</n1:SetResult>
      </n1:SetAttribute_OUTPUT>
    </s:Body>
  </s:Envelope>
```

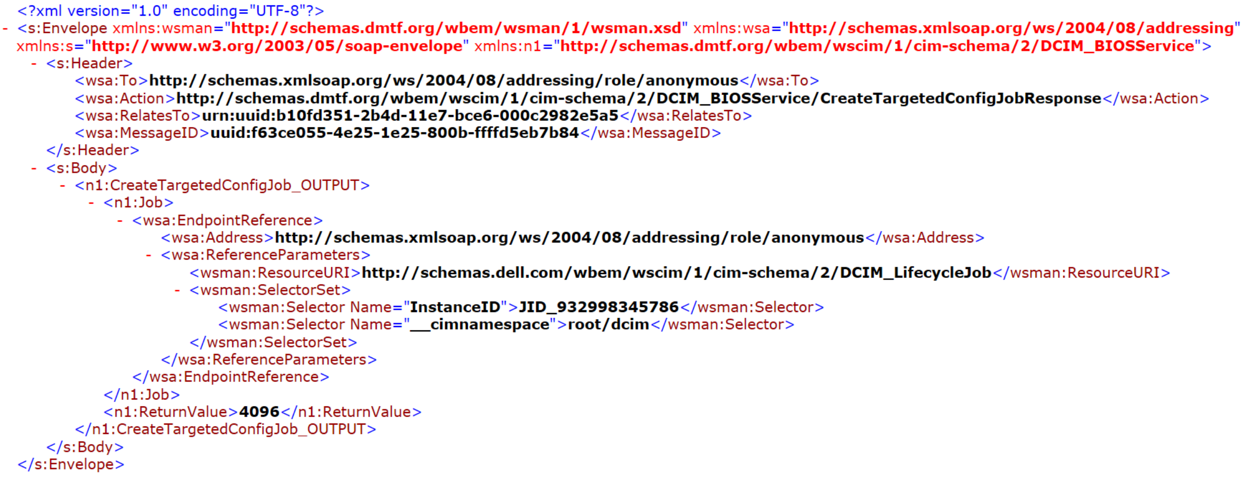
3. Management application requests to execute the changes.

SOAP Request for DCIM_BIOSService.CreateTargetedConfigJob()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService</wsman:ResourceURI>
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService/CreateTargetedConfigJob</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:b10fd351-2b4d-11e7-bce6-000c2982e5a5</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        <wsman:Selector Name="SystemCreationClassName">DCIM_ComputerSystem</wsman:Selector>
        <wsman:Selector Name="SystemName">DCIM:ComputerSystem</wsman:Selector>
        <wsman:Selector Name="CreationClassName">DCIM_BIOSService</wsman:Selector>
        <wsman:Selector Name="Name">DCIM:BIOSService</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
    - <n1:CreateTargetedConfigJob_INPUT>
        <n1:ScheduledStartTime>TIME_NOW</n1:ScheduledStartTime>
        <n1:Target>BIOS.Setup.1-1</n1:Target>
        <n1:RebootJobType>1</n1:RebootJobType>
      </n1:CreateTargetedConfigJob_INPUT>
    </s:Body>
  </s:Envelope>
```

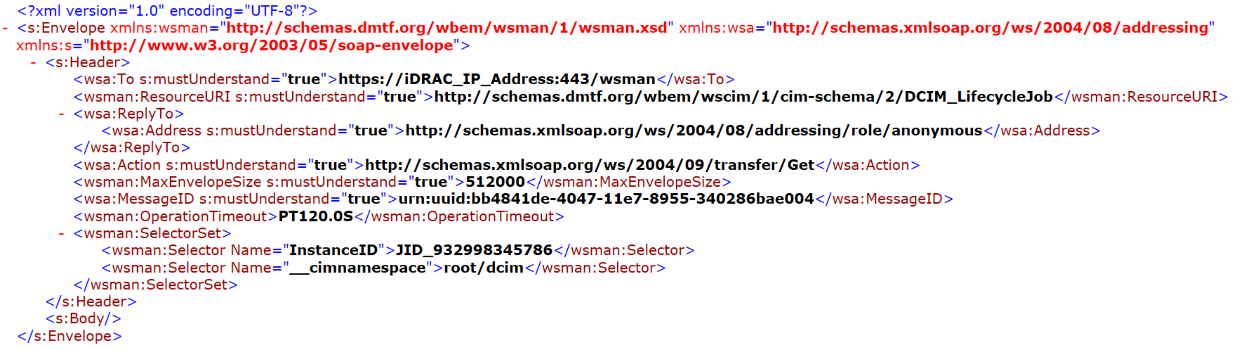
### SOAP Response for DCIM_BIOSService.CreateTargetedConfigJob()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService">
  - <s:Header>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService/CreateTargetedConfigJobResponse</wsa:Action>
        <wsa:RelatesTo>urn:uuid:b10fd351-2b4d-11e7-bce6-000c2982e5a5</wsa:RelatesTo>
        <wsa:MessageID>uuid:f63ce055-4e25-1e25-800b-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:CreateTargetedConfigJob_OUTPUT>
      - <n1:Job>
        - <wsa:EndpointReference>
              <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
          - <wsa:ReferenceParameters>
                <wsman:ResourceURI>http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LifecycleJob</wsman:ResourceURI>
            - <wsman:SelectorSet>
                  <wsman:Selector Name="InstanceID">JID_932998345786</wsman:Selector>
                  <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
              </wsman:SelectorSet>
            </wsa:ReferenceParameters>
          </wsa:EndpointReference>
        </n1:Job>
        <n1:ReturnValue>4096</n1:ReturnValue>
      </n1:CreateTargetedConfigJob_OUTPUT>
    </s:Body>
  </s:Envelope>
```

4. Management application requests to query the job status.

### SOAP Request for GET Job Status



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  - <s:Header>
        <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
        <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_LifecycleJob</wsman:ResourceURI>
      - <wsa:ReplyTo>
            <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:Action s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</wsa:Action>
        <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
        <wsa:MessageID s:mustUnderstand="true">urn:uuid:bb4841de-4047-11e7-8955-340286bae004</wsa:MessageID>
        <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
      - <wsman:SelectorSet>
            <wsman:Selector Name="InstanceID">JID_932998345786</wsman:Selector>
            <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        </wsman:SelectorSet>
    </s:Header>
    <s:Body/>
  </s:Envelope>
```

## Changing the Secure Boot policy

To change the secure boot policy to Custom or Standard, management applications can use the `DCIM_BIOSService.SetAttribute()` method from Dell_BIOSandBOOT management profile to set the `BIOS.Setup.1-1:SecureBootPolicy` attribute and create a job using `DCIM_BIOSService.CreateTargetedConfigJob()`method to execute this change. This enumeration attribute can be set to Custom or Standard.

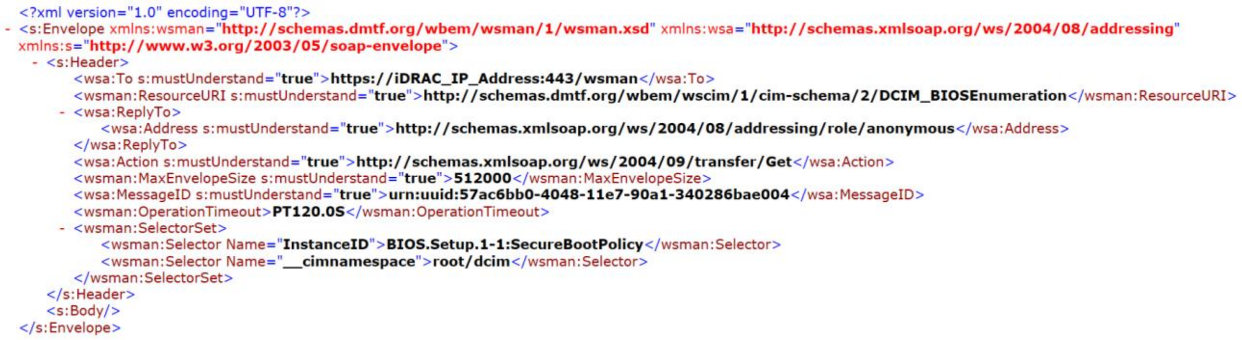
For more information, see the *Dell_BIOSandBOOTManagementProfile_4.0.0* Session 8.1 and 8.4 in the Profile Document available at http://en.community.dell.com/techcenter/systems-management/w/wiki/1906.dcim-library-profile.

By default, `BIOS.Setup.1-1:SecureBootPolicy` is configured as Standard. If the management application wants to Import, Export, Reset, or Delete the secure boot certificate, it must be changed to Custom.

Here is an example WS-Man workflow by using SOAP to set the `BIOS.Setup.1-1:SecureBootPolicy` attribute. In this workflow, the current value of BIOS.Setup.1-1:SecureBootPolicy will be checked, a job created to set a new value, job status queried to verify success, and finally verified that the value of `BIOS.Setup.1-1:SecureBootPolicy` has changed.

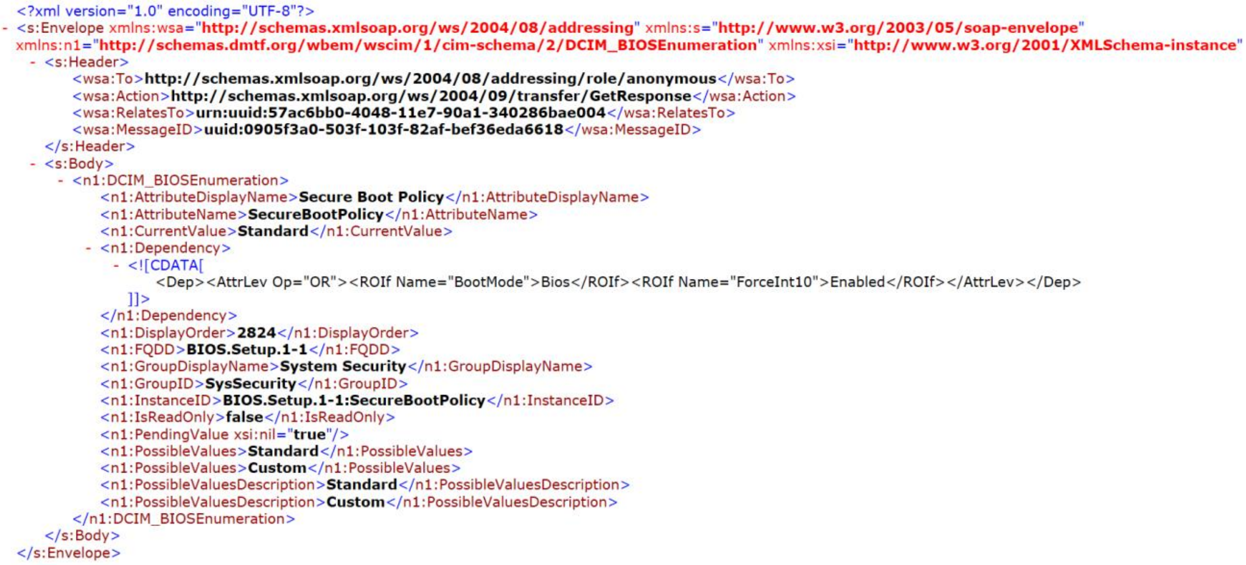
1. Management application requests to check the current value of the `BIOS.Setup.1-1:SecureBootPolicy`.

SOAP Request for GET BIOS.Setup.1-1:SecureBootPolicy attribute



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSEnumeration</wsman:ResourceURI>
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:57ac6bb0-4048-11e7-90a1-340286bae004</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="InstanceID">BIOS.Setup.1-1:SecureBootPolicy</wsman:Selector>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
    <s:Body/>
  </s:Envelope>
```

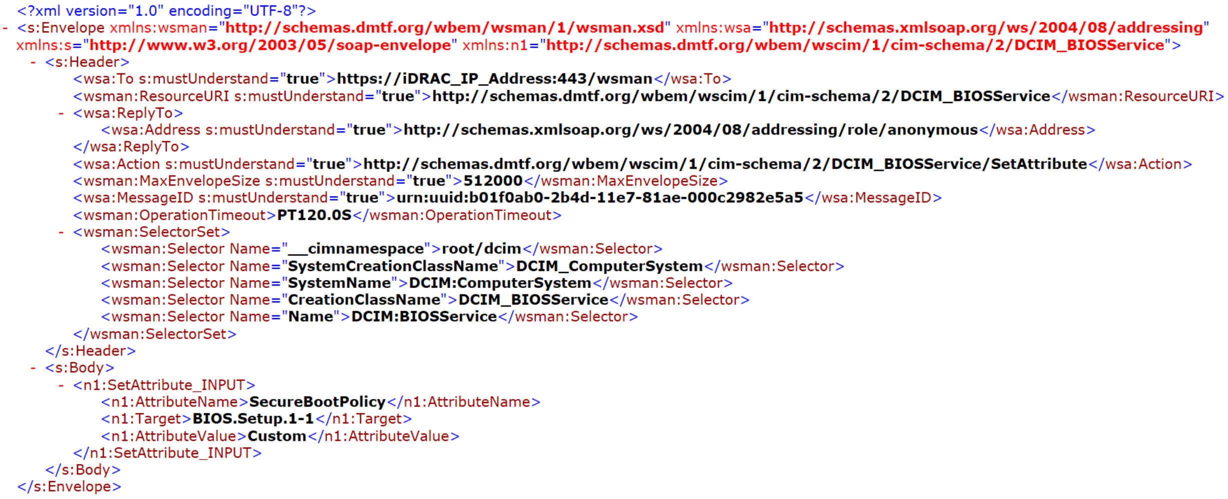
SOAP Response for GET BIOS.Setup.1-1:SecureBootPolicy attribute



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSEnumeration" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:57ac6bb0-4048-11e7-90a1-340286bae004</wsa:RelatesTo>
      <wsa:MessageID>uuid:0905f3a0-503f-103f-82af-bef36eda6618</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:DCIM_BIOSEnumeration>
        <n1:AttributeDisplayName>Secure Boot Policy</n1:AttributeDisplayName>
        <n1:AttributeName>SecureBootPolicy</n1:AttributeName>
        <n1:CurrentValue>Standard</n1:CurrentValue>
      - <n1:Dependency>
        - <![CDATA[
            <Dep><AttrLev Op="OR"><ROIf Name="BootMode">Bios</ROIf><ROIf Name="ForceInt10">Enabled</ROIf></AttrLev></Dep>
          ]]>
        </n1:Dependency>
        <n1:DisplayOrder>2824</n1:DisplayOrder>
        <n1:FQDD>BIOS.Setup.1-1</n1:FQDD>
        <n1:GroupDisplayName>System Security</n1:GroupDisplayName>
        <n1:GroupID>SysSecurity</n1:GroupID>
        <n1:InstanceID>BIOS.Setup.1-1:SecureBootPolicy</n1:InstanceID>
        <n1:IsReadOnly>false</n1:IsReadOnly>
        <n1:PendingValue xsi:nil="true"/>
        <n1:PossibleValues>Standard</n1:PossibleValues>
        <n1:PossibleValues>Custom</n1:PossibleValues>
        <n1:PossibleValuesDescription>Standard</n1:PossibleValuesDescription>
        <n1:PossibleValuesDescription>Custom</n1:PossibleValuesDescription>
      </n1:DCIM_BIOSEnumeration>
    </s:Body>
  </s:Envelope>
```

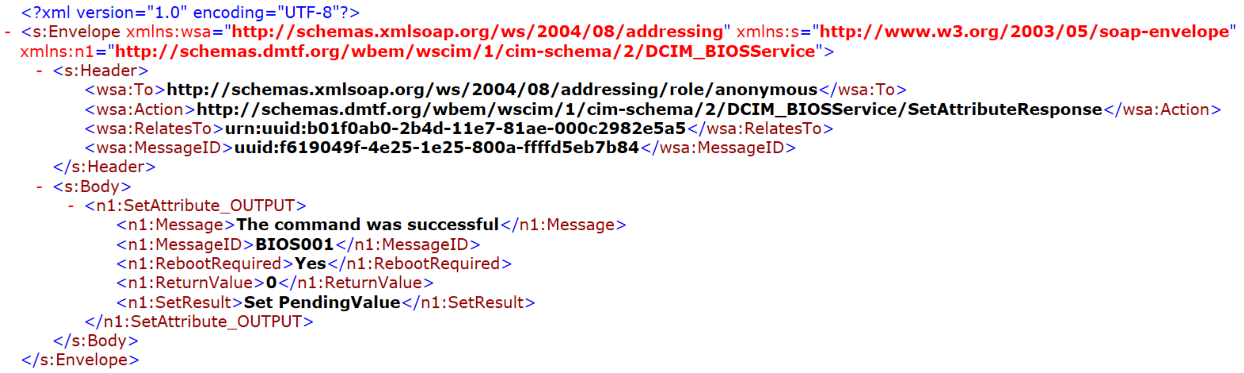
2. Management application requests to set the `BIOS.Setup.1-1:SecureBootPolicy` attribute to pending value.

SOAP Request for DCIM_BIOSService.SetAttribute() to set BIOS.Setup.1-1:SecureBootPolicy attribute



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService</wsman:ResourceURI>
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService/SetAttribute</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:b01f0ab0-2b4d-11e7-81ae-000c2982e5a5</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        <wsman:Selector Name="SystemCreationClassName">DCIM_ComputerSystem</wsman:Selector>
        <wsman:Selector Name="SystemName">DCIM:ComputerSystem</wsman:Selector>
        <wsman:Selector Name="CreationClassName">DCIM_BIOSService</wsman:Selector>
        <wsman:Selector Name="Name">DCIM:BIOSService</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
    - <n1:SetAttribute_INPUT>
        <n1:AttributeName>SecureBootPolicy</n1:AttributeName>
        <n1:Target>BIOS.Setup.1-1</n1:Target>
        <n1:AttributeValue>Custom</n1:AttributeValue>
      </n1:SetAttribute_INPUT>
    </s:Body>
  </s:Envelope>
```

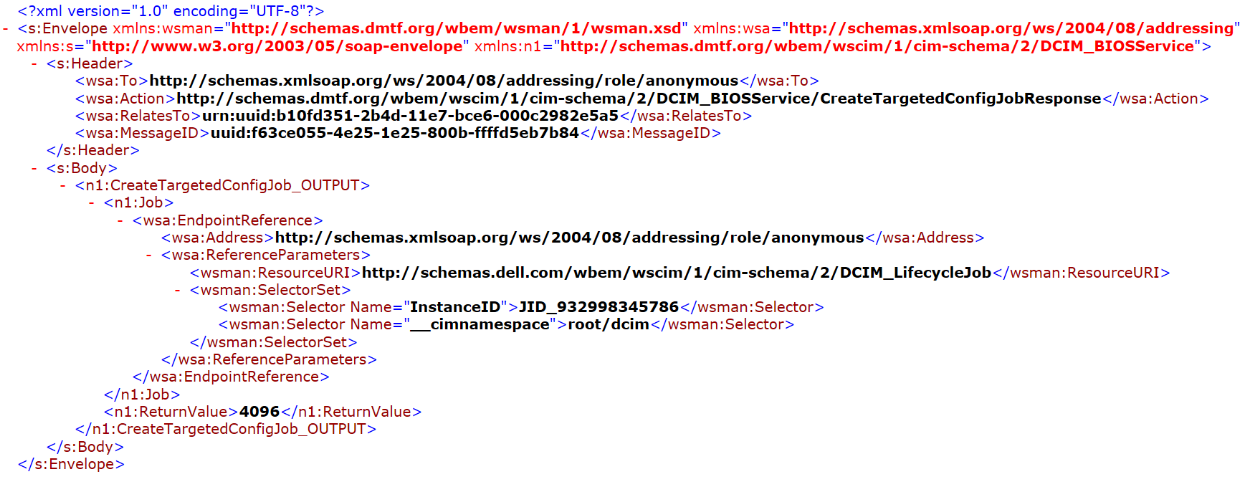
SOAP Response for DCIM_BIOSService.SetAttribute() to set BIOS.Setup.1-1:SecureBootPolicy attribute



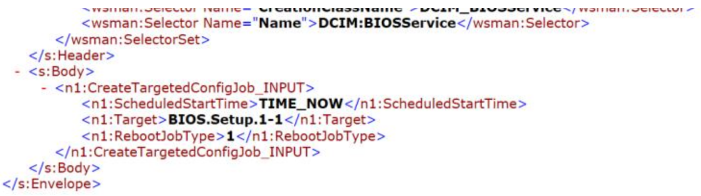
```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService">
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService/SetAttributeResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:b01f0ab0-2b4d-11e7-81ae-000c2982e5a5</wsa:RelatesTo>
      <wsa:MessageID>uuid:f619049f-4e25-1e25-800a-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:SetAttribute_OUTPUT>
        <n1:Message>The command was successful</n1:Message>
        <n1:MessageID>BIOS001</n1:MessageID>
        <n1:RebootRequired>Yes</n1:RebootRequired>
        <n1:ReturnValue>0</n1:ReturnValue>
        <n1:SetResult>Set PendingValue</n1:SetResult>
      </n1:SetAttribute_OUTPUT>
    </s:Body>
  </s:Envelope>
```

3. Management application requests to execute the changes SOAP Request for DCIM_BIOSService.CreateTargetedConfigJob()

SOAP Response for DCIM_BIOSService.CreateTargetedConfigJob()

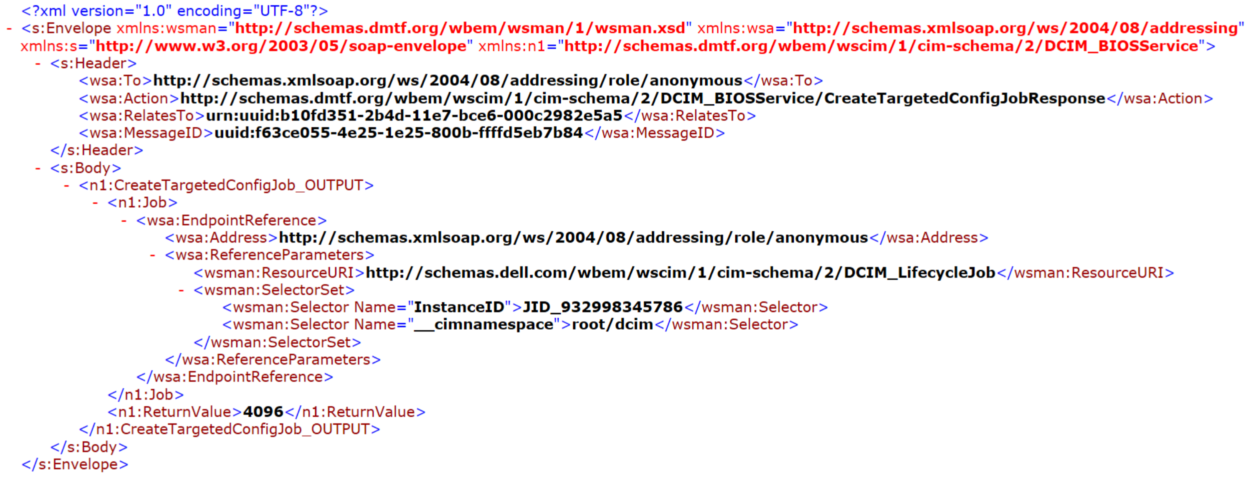


```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService">
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService/CreateTargetedConfigJobResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:b10fd351-2b4d-11e7-bce6-000c2982e5a5</wsa:RelatesTo>
      <wsa:MessageID>uuid:f63ce055-4e25-1e25-800b-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:CreateTargetedConfigJob_OUTPUT>
      - <n1:Job>
        - <wsa:EndpointReference>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
          - <wsa:ReferenceParameters>
              <wsman:ResourceURI>http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LifecycleJob</wsman:ResourceURI>
            - <wsman:SelectorSet>
                <wsman:Selector Name="InstanceID">JID_932998345786</wsman:Selector>
                <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
              </wsman:SelectorSet>
            </wsa:ReferenceParameters>
          </wsa:EndpointReference>
        </n1:Job>
        <n1:ReturnValue>4096</n1:ReturnValue>
      </n1:CreateTargetedConfigJob_OUTPUT>
    </s:Body>
  </s:Envelope>
```



```
        <wsman:Selector Name="CreationClassName">DCIM_BIOSService</wsman:Selector>
        <wsman:Selector Name="Name">DCIM:BIOSService</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
    - <n1:CreateTargetedConfigJob_INPUT>
        <n1:ScheduledStartTime>TIME_NOW</n1:ScheduledStartTime>
        <n1:Target>BIOS.Setup.1-1</n1:Target>
        <n1:RebootJobType>1</n1:RebootJobType>
      </n1:CreateTargetedConfigJob_INPUT>
    </s:Body>
  </s:Envelope>
```

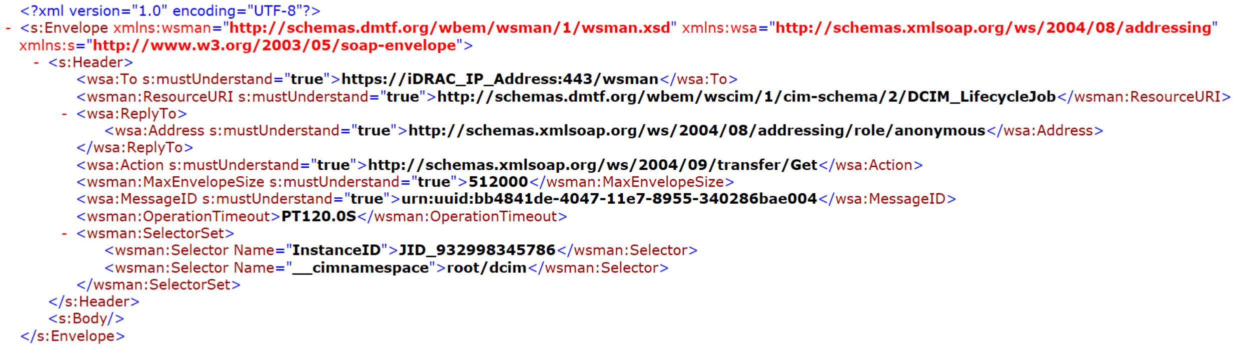
SOAP Response for DCIM_BIOSService.CreateTargetedConfigJob()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService">
  - <s:Header>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSService/CreateTargetedConfigJobResponse</wsa:Action>
        <wsa:RelatesTo>urn:uuid:b10fd351-2b4d-11e7-bce6-000c2982e5a5</wsa:RelatesTo>
        <wsa:MessageID>uuid:f63ce055-4e25-1e25-800b-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
      - <n1:CreateTargetedConfigJob_OUTPUT>
          - <n1:Job>
              - <wsa:EndpointReference>
                    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
                  - <wsa:ReferenceParameters>
                        <wsman:ResourceURI>http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LifecycleJob</wsman:ResourceURI>
                      - <wsman:SelectorSet>
                            <wsman:Selector Name="InstanceID">JID_932998345786</wsman:Selector>
                            <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
                        </wsman:SelectorSet>
                    </wsa:ReferenceParameters>
                </wsa:EndpointReference>
            </n1:Job>
            <n1:ReturnValue>4096</n1:ReturnValue>
        </n1:CreateTargetedConfigJob_OUTPUT>
    </s:Body>
  </s:Envelope>
```

4. Management application requests to query the job status.

SOAP Request for GET Job Status.



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  - <s:Header>
        <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
        <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_LifecycleJob</wsman:ResourceURI>
      - <wsa:ReplyTo>
            <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:Action s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</wsa:Action>
        <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
        <wsa:MessageID s:mustUnderstand="true">urn:uuid:bb4841de-4047-11e7-8955-340286bae004</wsa:MessageID>
        <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
      - <wsman:SelectorSet>
            <wsman:Selector Name="InstanceID">JID_932998345786</wsman:Selector>
            <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        </wsman:SelectorSet>
    </s:Header>
    <s:Body/>
  </s:Envelope>
```

5. Repeat the Step-1 procedure to verify that the value of `BIOS.Setup.1-1:SecureBootPolicy` has changed.

## *Viewing the Secure Boot certificate*

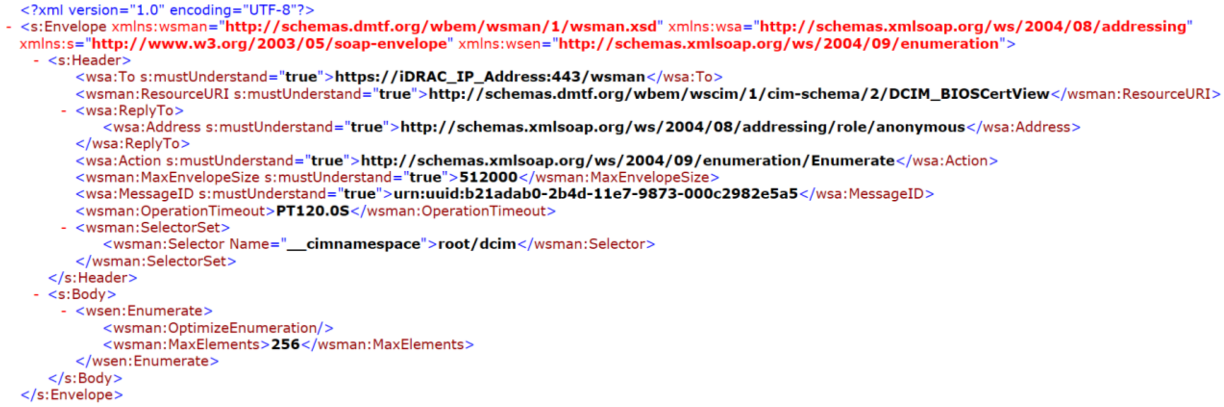
Management application can enumerate the DCIM_BIOSCertView class to see all the stored Secure Boot Certificate information, and can send the get request for particular instance to see that particular certificate information.

If the `BIOS.Setup.1-1:SecureBootPolicy` enumeration attribute configured as Custom**,** the DCIM_BIOSCertView will return only sorted custom Certificate , If the management application wants to see the standard certificates are stored in iDRAC `BIOS.Setup.1-1:SecureBootPolicy` attribute must be changed to Standard.

For more information, see the *Dell_BIOSandBOOTManagementProfile_4.0.0* Session 7.6 in the Profile Document available on http://en.community.dell.com/techcenter/systems-management/w/wiki/1906.dcim-library-profile.

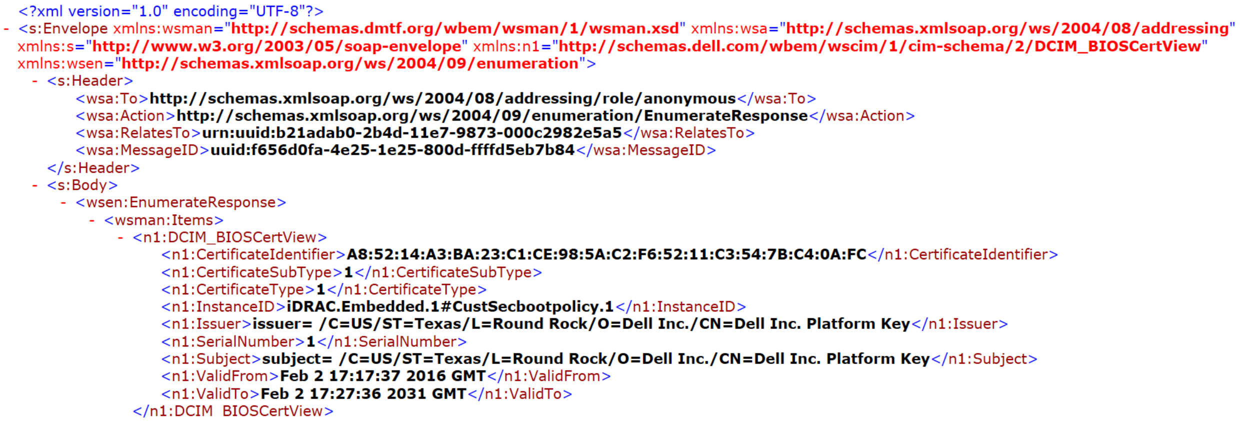
1. Management application sends the SOAP request for enumerating the DCIM_BIOSCertView.

   SOAP Request for Enumerate the DCIM_BIOSCertView



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertView</wsman:ResourceURI>
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:b21adab0-2b4d-11e7-9873-000c2982e5a5</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
    - <wsen:Enumerate>
        <wsman:OptimizeEnumeration/>
        <wsman:MaxElements>256</wsman:MaxElements>
      </wsen:Enumerate>
    </s:Body>
  </s:Envelope>
```

SOAP Response for Enumerate the DCIM_BIOSCertView

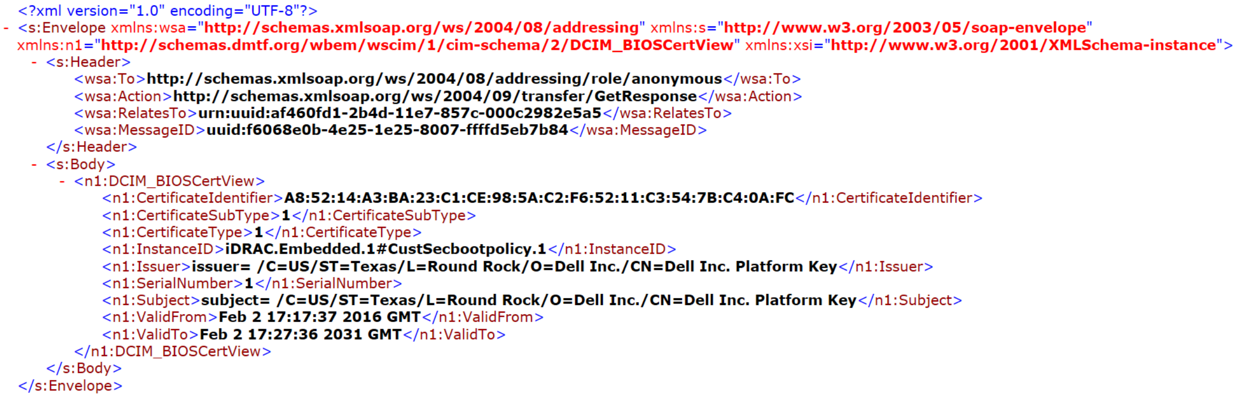


```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertView"
  xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration">
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:b21adab0-2b4d-11e7-9873-000c2982e5a5</wsa:RelatesTo>
      <wsa:MessageID>uuid:f656d0fa-4e25-1e25-800d-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <wsen:EnumerateResponse>
      - <wsman:Items>
        - <n1:DCIM_BIOSCertView>
            <n1:CertificateIdentifier>A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC</n1:CertificateIdentifier>
            <n1:CertificateSubType>1</n1:CertificateSubType>
            <n1:CertificateType>1</n1:CertificateType>
            <n1:InstanceID>iDRAC.Embedded.1#CustSecbootpolicy.1</n1:InstanceID>
            <n1:Issuer>issuer= /C=US/ST=Texas/L=Round Rock/O=Dell Inc./CN=Dell Inc. Platform Key</n1:Issuer>
            <n1:SerialNumber>1</n1:SerialNumber>
            <n1:Subject>subject= /C=US/ST=Texas/L=Round Rock/O=Dell Inc./CN=Dell Inc. Platform Key</n1:Subject>
            <n1:ValidFrom>Feb 2 17:17:37 2016 GMT</n1:ValidFrom>
            <n1:ValidTo>Feb 2 17:27:36 2031 GMT</n1:ValidTo>
          </n1:DCIM  BIOSCertView>
```

2. Management application sends the SOAP request to get the DCIM_BIOSCertView for one particular instance.

SOAP Request for Get the DCIM_BIOSCertView
SOAP Response for Get the DCIM_BIOSCertView

```
  <?xml version="1.0" encoding="UTF-8"?>
  <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
```



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertView" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:af460fd1-2b4d-11e7-857c-000c2982e5a5</wsa:RelatesTo>
      <wsa:MessageID>uuid:f6068e0b-4e25-1e25-8007-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:DCIM_BIOSCertView>
        <n1:CertificateIdentifier>A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC</n1:CertificateIdentifier>
        <n1:CertificateSubType>1</n1:CertificateSubType>
        <n1:CertificateType>1</n1:CertificateType>
        <n1:InstanceID>iDRAC.Embedded.1#CustSecbootpolicy.1</n1:InstanceID>
        <n1:Issuer>issuer= /C=US/ST=Texas/L=Round Rock/O=Dell Inc./CN=Dell Inc. Platform Key</n1:Issuer>
        <n1:SerialNumber>1</n1:SerialNumber>
        <n1:Subject>subject= /C=US/ST=Texas/L=Round Rock/O=Dell Inc./CN=Dell Inc. Platform Key</n1:Subject>
        <n1:ValidFrom>Feb 2 17:17:37 2016 GMT</n1:ValidFrom>
        <n1:ValidTo>Feb 2 17:27:36 2031 GMT</n1:ValidTo>
      </n1:DCIM_BIOSCertView>
    </s:Body>
  </s:Envelope>
```

## Importing the Secure Boot certificate

Management applications can use the `DCIM_BIOSCertService.ImportBootCertificate()` method from the Dell_BIOSandBOOT management profile to import the Secure boot Certificates.

`The BIOS.Setup.1-1:SecureBootPolicy` enumeration attribute current value should be in “Custom” to perform import operation. Secure Boot Certificate can be imported from NFS (0), CIFS (2), HTTP (5), or HTTPS (6) shares as specified by the “ShareType” parameter. The share where the certificate file, as specified by “FileName”, should be saved is specified in the “ShareName” parameter. In the case of CIFS, credentials to access the share must be specified by the “UserName” and “Password” parameters. The certificate type, Certificate Sub Type of the certificate to be imported must be as specified in “CertificateType”, “CertificateSubType”.

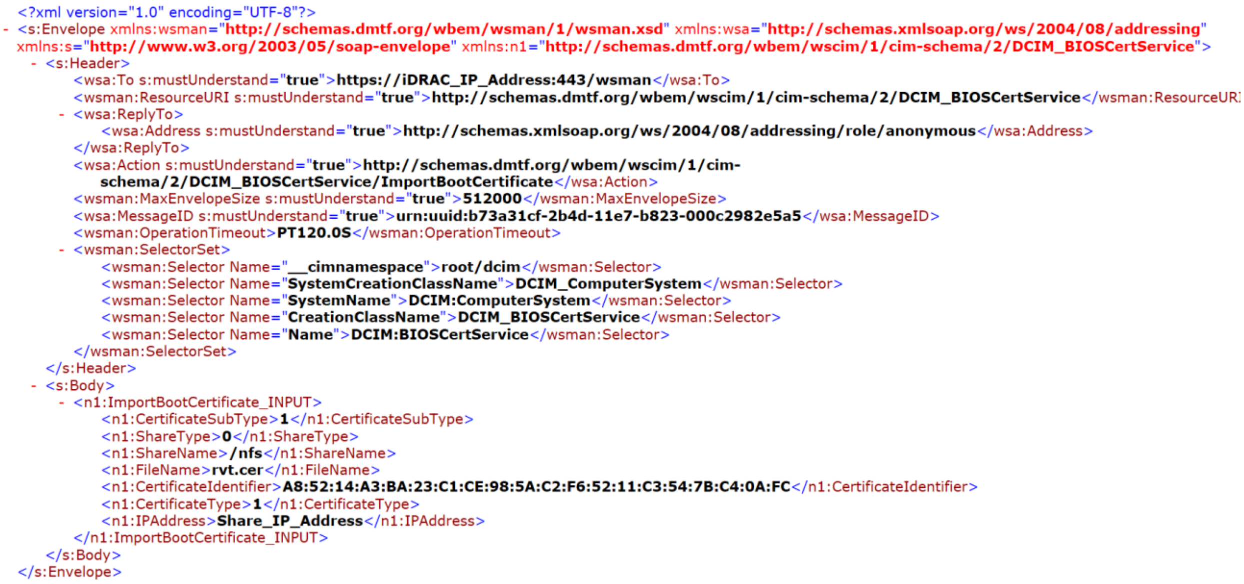
For more information, see the *Dell_BIOSandBOOTManagementProfile_4.0.0* Session 8.9 in Profile Document available at http://en.community.dell.com/techcenter/systems-management/w/wiki/1906.dcim-library-profile.

Here is an example WS-Man workflow by using SOAP to import the Secure Boot Certificate. In this workflow, the current value of `BIOS.Setup.1-1:SecureBootPolicy` is checked. If it is Standard, use the `DCIM_BIOSService.SetAttribute()` and `DCIM_BIOSService.CreateTargetedConfigJob()` method to change it to Custom. And call the `ImportBootCertificate()` method to import the Secure Boot certificate.

1. To check and change the `BIOS.Setup.1-1:SecureBootPolicy` attribute to Custom, refer to and follow the steps mentioned in Changing the Secure policy.

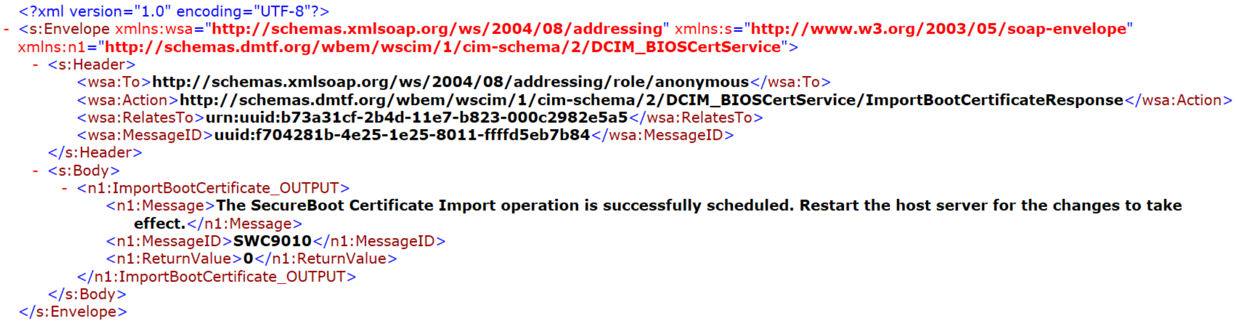
2. Management application sends the SOAP request to import the Secure Boot certificate from external share.

SOAP Request for DCIM_BIOSCertService.ImportBootCertificate()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService</wsman:ResourceURI
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-
        schema/2/DCIM_BIOSCertService/ImportBootCertificate</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:b73a31cf-2b4d-11e7-b823-000c2982e5a5</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        <wsman:Selector Name="SystemCreationClassName">DCIM_ComputerSystem</wsman:Selector>
        <wsman:Selector Name="SystemName">DCIM:ComputerSystem</wsman:Selector>
        <wsman:Selector Name="CreationClassName">DCIM_BIOSCertService</wsman:Selector>
        <wsman:Selector Name="Name">DCIM:BIOSCertService</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
    - <n1:ImportBootCertificate_INPUT>
        <n1:CertificateSubType>1</n1:CertificateSubType>
        <n1:ShareType>0</n1:ShareType>
        <n1:ShareName>/nfs</n1:ShareName>
        <n1:FileName>rvt.cer</n1:FileName>
        <n1:CertificateIdentifier>A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC</n1:CertificateIdentifier>
        <n1:CertificateType>1</n1:CertificateType>
        <n1:IPAddress>Share_IP_Address</n1:IPAddress>
      </n1:ImportBootCertificate_INPUT>
    </s:Body>
  </s:Envelope>
```

SOAP Response for DCIM_BIOSCertService.ImportBootCertificate()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService">
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService/ImportBootCertificateResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:b73a31cf-2b4d-11e7-b823-000c2982e5a5</wsa:RelatesTo>
      <wsa:MessageID>uuid:f704281b-4e25-1e25-8011-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:ImportBootCertificate_OUTPUT>
        <n1:Message>The SecureBoot Certificate Import operation is successfully scheduled. Restart the host server for the changes to take
          effect.</n1:Message>
        <n1:MessageID>SWC9010</n1:MessageID>
        <n1:ReturnValue>0</n1:ReturnValue>
      </n1:ImportBootCertificate_OUTPUT>
    </s:Body>
  </s:Envelope>
```

3. Restart the host server for the changes to take effect.

## *Exporting the Secure Boot certificate*

Management applications can use the `DCIM_BIOSCertService.ExportBootCertificate()` method from Dell_BIOSandBOOT management profile to export the Secure boot Certificates.

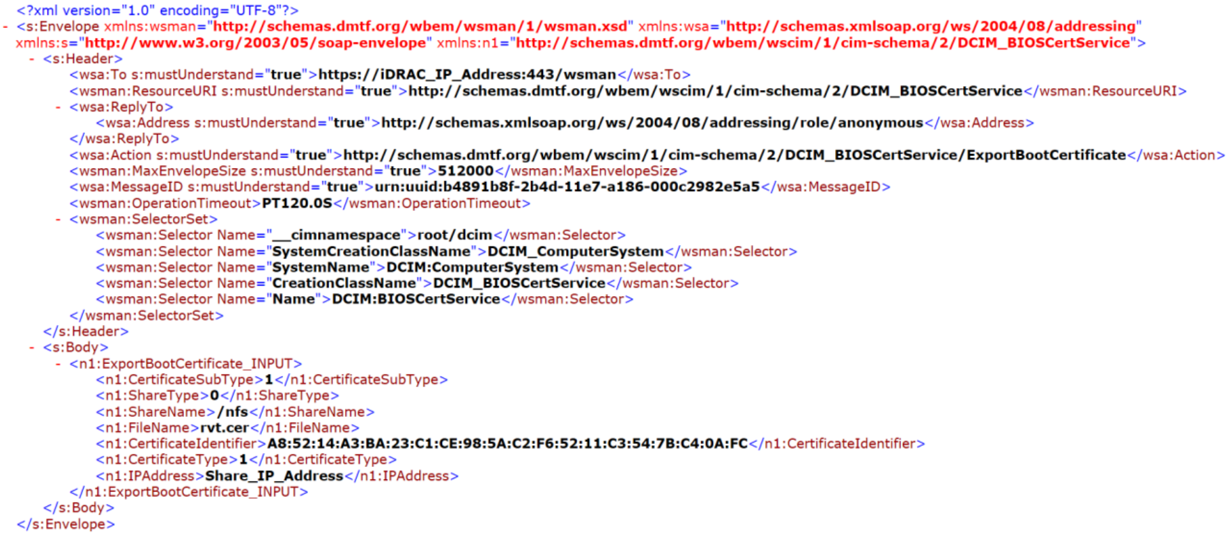
The `BIOS.Setup.1-1:SecureBootPolicy` enumeration attribute current value must be in “Custom” to perform export operation. Secure Boot certificate can be exported to NFS (0), CIFS (2), HTTP (5), or HTTPS (6) shares, as specified by the “ShareType” parameter. The share where the certificate file as specified by “FileName” should reside as specified in the “ShareName” parameter. In the case of CIFS shares, credentials to access the share should also be specified by the “UserName” and “Password” parameters. The certificate type, Certificate Sub Type and Certificate Identifier of the certificate to be exported as specified in “CertificateType”, “CertificateSubType” and “CertificateIdentifier”. For more information, see the *Dell_BIOSandBOOTManagementProfile_4.0.0* Session 8.8 Profile Document available on http://en.community.dell.com/techcenter/systems-management/w/wiki/1906.dcim-library-profile.

Here is an example WS-Man workflow by using SOAP to import Secure Boot Certificate. In this workflow the current value of **BIOS.Setup.1-1:SecureBootPolicy** is checked.  If it is Standard, use the `DCIM_BIOSService.SetAttribute()` and `DCIM_BIOSService.CreateTargetedConfigJob()` method to change it to Custom. And call the `ExportBootCertificate()` method  to export the certificate to external share.

1. To check and change the `BIOS.Setup.1-1:SecureBootPolicy` attribute to Custom, refer to and follow the steps mentioned in Changing the Secure policy.

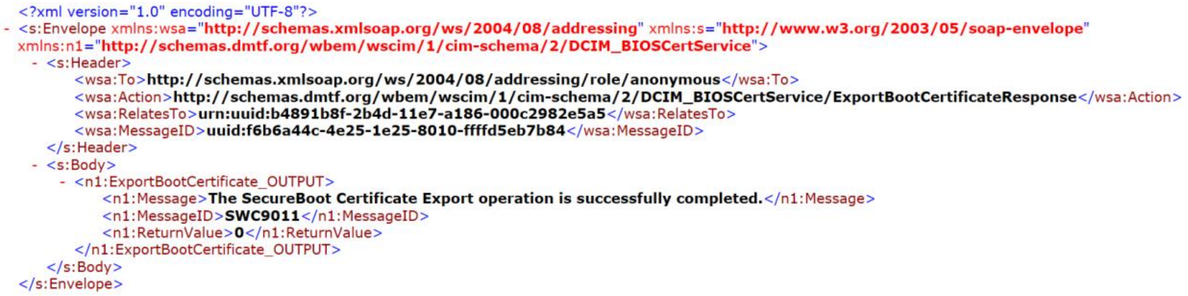
2. Management application sends the SOAP request to export the Secure Boot certificate from external share.

SOAP Request  for DCIM_BIOSCertService.ExportBootCertificate()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService</wsman:ResourceURI>
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService/ExportBootCertificate</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:b4891b8f-2b4d-11e7-a186-000c2982e5a5</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        <wsman:Selector Name="SystemCreationClassName">DCIM_ComputerSystem</wsman:Selector>
        <wsman:Selector Name="SystemName">DCIM:ComputerSystem</wsman:Selector>
        <wsman:Selector Name="CreationClassName">DCIM_BIOSCertService</wsman:Selector>
        <wsman:Selector Name="Name">DCIM:BIOSCertService</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
    - <n1:ExportBootCertificate_INPUT>
        <n1:CertificateSubType>1</n1:CertificateSubType>
        <n1:ShareType>0</n1:ShareType>
        <n1:ShareName>/nfs</n1:ShareName>
        <n1:FileName>rvt.cer</n1:FileName>
        <n1:CertificateIdentifier>A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC</n1:CertificateIdentifier>
        <n1:CertificateType>1</n1:CertificateType>
        <n1:IPAddress>Share_IP_Address</n1:IPAddress>
      </n1:ExportBootCertificate_INPUT>
    </s:Body>
  </s:Envelope>
```

SOAP Response for DCIM_BIOSCertService.ExportBootCertificate()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService">
  - <s:Header>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService/ExportBootCertificateResponse</wsa:Action>
        <wsa:RelatesTo>urn:uuid:b4891b8f-2b4d-11e7-a186-000c2982e5a5</wsa:RelatesTo>
        <wsa:MessageID>uuid:f6b6a44c-4e25-1e25-8010-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
      - <n1:ExportBootCertificate_OUTPUT>
            <n1:Message>The SecureBoot Certificate Export operation is successfully completed.</n1:Message>
            <n1:MessageID>SWC9011</n1:MessageID>
            <n1:ReturnValue>0</n1:ReturnValue>
        </n1:ExportBootCertificate_OUTPUT>
    </s:Body>
  </s:Envelope>
```

## Resetting the Secure Boot certificate

Management applications can use the `DCIM_BIOSCertService.ResetBootCertificate()` method from the Dell_BIOSandBOOT management profile to reset or copy the standard policy certificate in to Cusom policy.

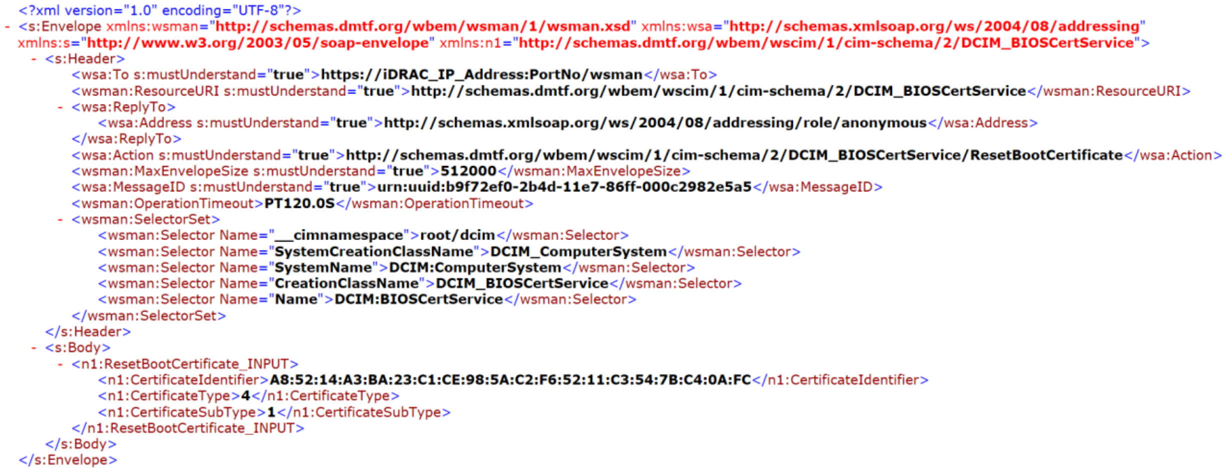
**The BIOS.Setup.1-1:SecureBootPolicy** enumeration attribute current value should be in “Custom” to perform delete operations. All or specific certificate can be reset as specified by “CertificateType”.

For more information, see the *Dell_BIOSandBOOTManagementProfile_4.0.0* Session 8.10 Profile Document  available http://en.community.dell.com/techcenter/systems-management/w/wiki/1906.dcim-library-profile.

Below is an example WSMAN workflow using SOAP to Delete the Secure Boot Certificate. In this workflow the current value of **BIOS.Setup.1-1:SecureBootPolicy** will be checked.  If it is Standard, Use the DCIM_BIOSService.SetAttribute() and DCIM_BIOSService.CreateTargetedConfigJob() method to change it to Custom. And called the ResetBootCertificate() method  to reset the certificate.

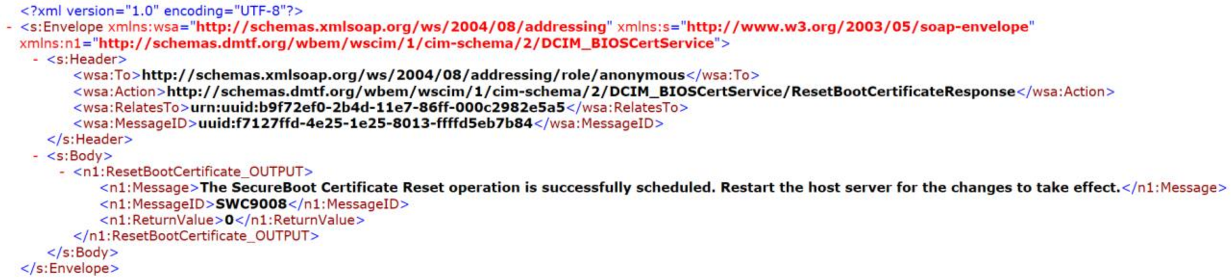
1. To check and change the `BIOS.Setup.1-1:SecureBootPolicy` attribute to Custom, refer to and follow the steps mentioned in Changing the Secure policy.
2. Management application sends the SOAP request to reset the Secure Boot certificate.

SOAP Request for DCIM_BIOSCertService.ResetBootCertificate()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService">
  - <s:Header>
        <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:PortNo/wsman</wsa:To>
        <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService</wsman:ResourceURI>
      - <wsa:ReplyTo>
            <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:Action s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService/ResetBootCertificate</wsa:Action>
        <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
        <wsa:MessageID s:mustUnderstand="true">urn:uuid:b9f72ef0-2b4d-11e7-86ff-000c2982e5a5</wsa:MessageID>
        <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
      - <wsman:SelectorSet>
            <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
            <wsman:Selector Name="SystemCreationClassName">DCIM_ComputerSystem</wsman:Selector>
            <wsman:Selector Name="SystemName">DCIM:ComputerSystem</wsman:Selector>
            <wsman:Selector Name="CreationClassName">DCIM_BIOSCertService</wsman:Selector>
            <wsman:Selector Name="Name">DCIM:BIOSCertService</wsman:Selector>
        </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
      - <n1:ResetBootCertificate_INPUT>
            <n1:CertificateIdentifier>A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC</n1:CertificateIdentifier>
            <n1:CertificateType>4</n1:CertificateType>
            <n1:CertificateSubType>1</n1:CertificateSubType>
        </n1:ResetBootCertificate_INPUT>
    </s:Body>
  </s:Envelope>
```

SOAP Response for DCIM_BIOSCertService.ResetBootCertificate()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService">
  - <s:Header>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService/ResetBootCertificateResponse</wsa:Action>
      <wsa:RelatesTo>urn:uuid:b9f72ef0-2b4d-11e7-86ff-000c2982e5a5</wsa:RelatesTo>
      <wsa:MessageID>uuid:f7127ffd-4e25-1e25-8013-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:ResetBootCertificate_OUTPUT>
        <n1:Message>The SecureBoot Certificate Reset operation is successfully scheduled. Restart the host server for the changes to take effect.</n1:Message>
        <n1:MessageID>SWC9008</n1:MessageID>
        <n1:ReturnValue>0</n1:ReturnValue>
      </n1:ResetBootCertificate_OUTPUT>
    </s:Body>
  </s:Envelope>
```

3. Restart the host server for the changes to take effect.

## Deleting Secure Boot certificate

Management applications can use the `DCIM_BIOSCertService.DeleteBootCertificate()` method from Dell_BIOSandBOOT management profile to Delete the Secure boot Certificates.

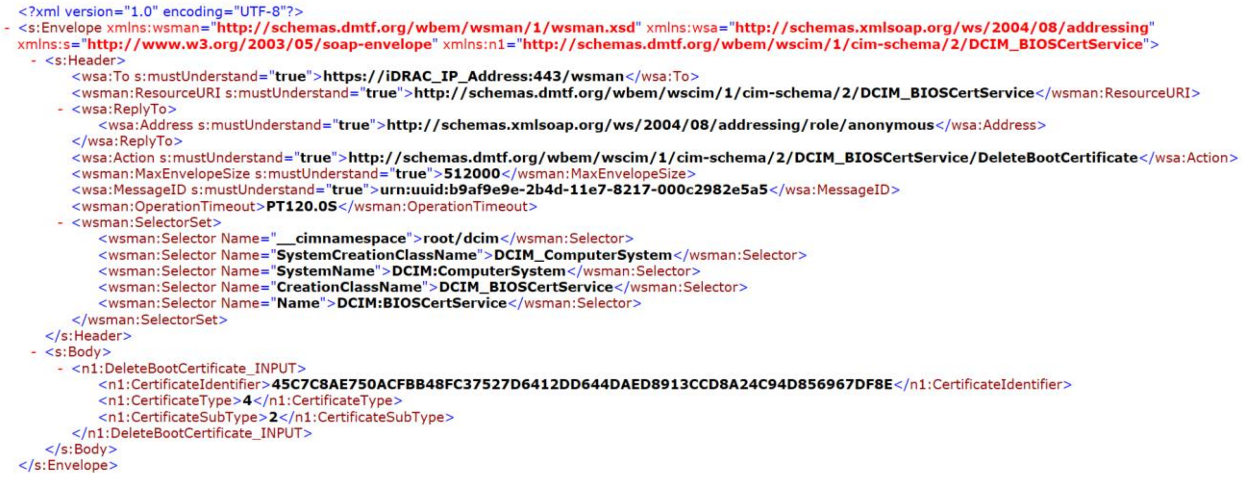
The `BIOS.Setup.1-1:SecureBootPolicy` enumeration attribute current value should be in “Custom” to perform delete operation. All or specific certificate can be deleted as specified by “CertificateType” and “CertificateSubType”.

For more information, see the *Dell_BIOSandBOOTManagementProfile_4.0.0* Session 8.11 Profile Document  available http://en.community.dell.com/techcenter/systems-management/w/wiki/1906.dcim-library-profile.

Here is an example WS-Man workflow by using the SOAP to delete the Secure Boot Certificate. In this workflow, the current value of `BIOS.Setup.1-1:SecureBootPolicy` is checked.  If it is Standard, use the `DCIM_BIOSService.SetAttribute()` and `DCIM_BIOSService.CreateTargetedConfigJob()` method to change it to Custom. The DeleteBootCertificate() method  is called to delete the certificate.

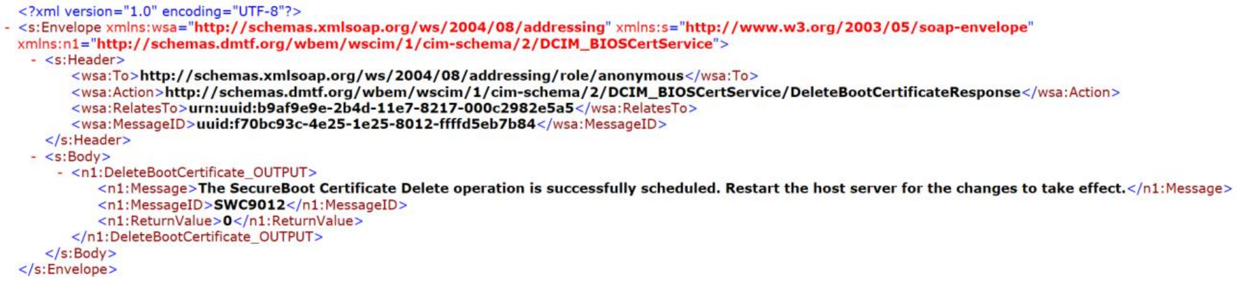
1. To check and change the `BIOS.Setup.1-1:SecureBootPolicy` attribute to Custom, refer to and follow the steps mentioned in Changing the Secure policy.
2. Management application sends the SOAP request to reset the Secure Boot certificate.

SOAP Request for DCIM_BIOSCertService.DeleteBootCertificate()



```
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService">
  - <s:Header>
      <wsa:To s:mustUnderstand="true">https://iDRAC_IP_Address:443/wsman</wsa:To>
      <wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService</wsman:ResourceURI>
    - <wsa:ReplyTo>
        <wsa:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
      </wsa:ReplyTo>
      <wsa:Action s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService/DeleteBootCertificate</wsa:Action>
      <wsman:MaxEnvelopeSize s:mustUnderstand="true">512000</wsman:MaxEnvelopeSize>
      <wsa:MessageID s:mustUnderstand="true">urn:uuid:b9af9e9e-2b4d-11e7-8217-000c2982e5a5</wsa:MessageID>
      <wsman:OperationTimeout>PT120.0S</wsman:OperationTimeout>
    - <wsman:SelectorSet>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        <wsman:Selector Name="SystemCreationClassName">DCIM_ComputerSystem</wsman:Selector>
        <wsman:Selector Name="SystemName">DCIM:ComputerSystem</wsman:Selector>
        <wsman:Selector Name="CreationClassName">DCIM_BIOSCertService</wsman:Selector>
        <wsman:Selector Name="Name">DCIM:BIOSCertService</wsman:Selector>
      </wsman:SelectorSet>
    </s:Header>
  - <s:Body>
    - <n1:DeleteBootCertificate_INPUT>
        <n1:CertificateIdentifier>45C7C8AE750ACFBB48FC37527D6412DD644DAED8913CCD8A24C94D856967DF8E</n1:CertificateIdentifier>
        <n1:CertificateType>4</n1:CertificateType>
        <n1:CertificateSubType>2</n1:CertificateSubType>
      </n1:DeleteBootCertificate_INPUT>
    </s:Body>
  </s:Envelope>
```

SOAP Response for DCIM_BIOSCertService.DeleteBootCertificate()



```xml
  <?xml version="1.0" encoding="UTF-8"?>
- <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService">
  - <s:Header>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsa:Action>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_BIOSCertService/DeleteBootCertificateResponse</wsa:Action>
        <wsa:RelatesTo>urn:uuid:b9af9e9e-2b4d-11e7-8217-000c2982e5a5</wsa:RelatesTo>
        <wsa:MessageID>uuid:f70bc93c-4e25-1e25-8012-ffffd5eb7b84</wsa:MessageID>
    </s:Header>
  - <s:Body>
    - <n1:DeleteBootCertificate_OUTPUT>
          <n1:Message>The SecureBoot Certificate Delete operation is successfully scheduled. Restart the host server for the changes to take effect.</n1:Message>
          <n1:MessageID>SWC9012</n1:MessageID>
          <n1:ReturnValue>0</n1:ReturnValue>
      </n1:DeleteBootCertificate_OUTPUT>
    </s:Body>
  </s:Envelope>
```

3. Restart the host server for the changes to become effective.

# Example commands by using winrm

## *Viewing the Secure Boot Certificate*

```
winrm e http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/dcim_bioscertview -
u:idrac_username -p:idrac_password -r:https://idrac_ip/wsman -skipcncheck -skipcacheck -
encoding:utf-8 -a:basic
```

## *Importing the Secure Boot Certificate*

```
 winrm invoke ImportBootCertificate http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/root/dcim/DCIM_BIOSCertService?SystemCreationClassName=DCIM_ComputerSystem+Creati
onClassName=DCIM_BIOSCertService+SystemName=DCIM:ComputerSystem+Name=DCIM:BIOSCertService
@{UserName="ShareUserName";Password="Share_password";ShareType="2";ShareName="TestShare";I
PAddress="ShareIP";FileName="pk.cer";CertificateType="3";CertificateIdentifier="A8:52:14:A
3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC";CertificateSubType="1"} -
r:https://iDRAC_IP/wsman -u:iDRAC_UserName -p:iDRAC_PassWord -SkipCNcheck -SkipCAcheck -
SkipRevocationCheck -encoding:utf-8 -a:basic -format:pretty
```

## *Exporting the Secure Boot Certificate*

```
winrm invoke exportbootcertificate http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/root/dcim/dcim_bioscertservice?systemcreationclassname=dcim_computersystem+creati
onclassname=dcim_bioscertservice+systemname=dcim:computersystem+name=dcim:bioscertservice
@{username="shareusername";password="sharepassword";sharetype="2";sharename="testshare";ip
address="share_ip";filename="pk.cer";certificatetype="1";certificateidentifier="a8:52:14:a
3:ba:23:c1:ce:98:5a:c2:f6:52:11:c3:54:7b:c4:0a:fc";certificatesubtype="1"} -
r:https://idrac_ip/wsman -u:idrac_username -p:idrac_password -skipcncheck -skipcacheck -
skiprevocationcheck -encoding:utf-8 -a:basic -format:pretty
```

## *Resetting the Secure Boot Certificate*

```
 winrm invoke resetbootcertificate http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/root/dcim/dcim_bioscertservice?systemcreationclassname=dcim_computersystem+creati
onclassname=dcim_bioscertservice+systemname=dcim:computersystem+name=dcim:bioscertservice
@{certificatetype="1"} -r:https://idrac_ip/wsman -u:idrac_username -p:idrac_password -
skipcncheck -skipcacheck -skiprevocationcheck -encoding:utf-8 -a:basic -format:pretty
```

## *Deleting the Secure Boot Certificate*

```
winrm invoke deletebootcertificate http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/root/dcim/dcim_bioscertservice?systemcreationclassname=dcim_computersystem+creati
onclassname=dcim_bioscertservice+systemname=dcim:computersystem+name=dcim:bioscertservice
@{certificatetype="0"} -r:https://idrac_ip/wsman -u:idrac_username -p:idrac_password -
skipcncheck -skipcacheck -skiprevocationcheck -encoding:utf-8 -a:basic -format:pretty
```

## *Setting the Secure Boot Attribute*

```
winrm i setattributes http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/root/dcim/dcim_biosservice?__cimnamespace=root/dcim+systemcreationclassname=dcim_
computersystem+systemname=dcim:computersystem+creationclassname=dcim_biosservice+name=dcim
:biosservice -u:idrac_username -p:idrac_password -r:https://idrac_ip/wsman -skipcncheck -
skipcacheck -skiprevocationcheck -encoding:utf-8 -a:basic @{target="bios.setup.1-
1";attributename="securebootmode";attributevalue="auditmode"}
```

```
winrm i createtargetedconfigjob http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/root/dcim/dcim_biosservice?__cimnamespace=root/dcim+systemcreationclassname=dcim_
computersystem+systemname=dcim:computersystem+creationclassname=dcim_biosservice+name=dcim
:biosservice -u:idrac_username -p:idrac_password -r:https://idrac_ip/wsman -skipcncheck -
skipcacheck -skiprevocationcheck -encoding:utf-8 -a:basic @{target="bios.setup.1-
1";rebootjobtype="1";scheduledstarttime="time_now"}
```

# Secure Boot certificate management using Redfish

## *Viewing the Secure Boot settings resource*

The Secure Boot certificate management URI in Redfish can be located under the SecureBoot settings resource as listed here:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot |
| iDRAC privilege | login |
| HTTP request method | GET |

**Output**:

```
{
        "@odata.context": "/redfish/v1/$metadata#SecureBoot.SecureBoot",
        "@odata.id": "/redfish/v1/Systems/System.Embedded.1/SecureBoot",
        "@odata.type": "#SecureBoot.v1_0_0.SecureBoot",
        "Actions": {
                "#SecureBoot.ResetKeys": {
                        "ResetKeysType@Redfish.AllowableValues": [
                                "ResetAllKeysToDefault",
                                "DeleteAllKeys",
                                "DeletePK",
                                "ResetPK",
                                "ResetKEK",
                                "ResetDB",
                                "ResetDBX"
                        ],
                        "target":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Actions/SecureBoot.ResetKeys"
                },
                "Oem": {}
        },
        "Description": "UEFI Secure Boot",
        "Id": "SecureBoot",
        "Name": "UEFI Secure Boot",
        "Oem": {
                "Dell": {
                        "@odata.type": "#DellSecureBoot.v1_0_0.DellSecureBoot",
                        "Certificates": {
                                "@odata.id":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates"
                        }
                }
        },
        "SecureBootCurrentBoot": "Disabled",
        "SecureBootEnable": false,
        "SecureBootMode": "Deployed Mode"
}
```

## *Modifying the Secure Boot settings resource*

The SecureBoot settings resource can be modified by performing the PATCH operation as listed here:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot |
| iDRAC privilege | System Control |
| HTTP request method | PATCH |

**HTTP Request Body**:

```
{
        "SecureBootEnable": true
}
```

**Output:**

```
{
        "@Message.ExtendedInfo": [
                {
                        "Message": "Successfully Completed Request",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "Base.1.0.Success",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "None",
                        "Severity": "OK"
                },
                {
                        "Message": "The operation is successfully completed.",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "iDRAC.1.6.SYS430",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "No response action is required.However, to make them
immediately effective, restart the host server.",
                        "Severity": "Informational"
                }
        ]
}
```

## *Viewing the Certificate store collection*

As part the Secure Boot certificate management, the certificates are stores in the various certificate stores such as PK, KEK, DB and DBX. These can we located under the following resource ID:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates |
| iDRAC privilege | login |
| HTTP request method | GET |

**Output:**

```
{
        "@odata.context":
"/redfish/v1/$metadata#DellCertificateStoreCollection.DellCertificateStoreCollection",
        "@odata.id": "/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates",
        "@odata.type": "#DellCertificateStoreCollection.DellCertificateStoreCollection",
        "Description": "DellCertificateStoreCollection",
        "Members": [
                {
                        "@odata.id":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/PK"
                },
                {
                        "@odata.id":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/KEK"
                },
                {
                        "@odata.id":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DB"
                },
                {
                        "@odata.id":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DBX"
                }
        ],
        "Members@odata.count": 4,
        "Name": "DellCertificateStoreCollection"
}
```

## *Viewing the Certificate and Hash collection in Certificate store*

The HTTP GET operation on each of the certificate store resources such as PK, KEK, DB, and DBX list all the Certificate and the Hash resources under that particular certificate store as listed here:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/PK |
| iDRAC privilege | login |
| HTTP request method | GET |

**Output:**

```
{
       "@odata.context":
"/redfish/v1/$metadata#DellCertificateCollection.DellCertificateCollection",
       "@odata.id": "/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/PK",
       "@odata.type": "#DellCertificateCollection.DellCertificateCollection",
       "Certificates": [
              {
                     "@odata.id":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/PK/iDRAC.Embedded.1%23CustS
ecbootpolicy.1",
                     "CertificateSubtype": "Certificate",
                     "CertificateType": "PK",
                     "IssuerCommonName_CN": "Dell Inc. Platform Key",
                     "IssuerCountryCode_CC": "US",
                     "IssuerLocality_L": "Round Rock",
                     "IssuerOrganization_O": "Dell Inc.",
                     "IssuerState_S": "Texas",
                     "SecureBootPolicy": "Custom",
                     "SerialNumber": "18E0E033DB57CD984ABB23689D61BE4D",
                     "SubjectCommonName_CN": "Dell Inc. Platform Key",
                     "SubjectCountryCode_CC": "US",
                     "SubjectLocality_L": "Round Rock",
                     "SubjectOrganization_O": "Dell Inc.",
                     "SubjectState_S": "Texas",
                     "Thumbprint":
"A8:52:14:A3:BA:23:C1:CE:98:5A:C2:F6:52:11:C3:54:7B:C4:0A:FC",
                     "ValidFrom": "Feb  2 17:17:37 2016 GMT",
                     "ValidTo": "Feb  2 17:27:36 2031 GMT"
              }
       ],
       "Certificates@odata.count": 1,
       "Description": "DellCertificateCollection",
       "Hash": [],
       "Hash@odata.count": 0,
       "Name": "DellCertificateCollection"
}
```

## Uploading the Certificate to Certificate store

The certificate file can be uploaded to the certificate store by performing a HTTP POST operation with a Content-Type as “multipart/form-data” and body to have the certificate file uploaded as listed here:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DB |
| iDRAC privilege | System Control |
| HTTP request method | POST |

**HTTP Request Header:**

**Content-Type:** `“multipart/form-data"`

**Output:**

```
{
        "@Message.ExtendedInfo": [
                {
                        "Message": "Successfully Completed Request",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "Base.1.0.Success",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "None",
                        "Severity": "OK"
                },
                {
                        "Message": "The operation successfully completed.",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "iDRAC.1.6.SYS413",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "No response action is required.",
                        "Severity": "Informational"
                }
        ]
}
```

## *Uploading the Hash to Certificate store*

The Hash file can be uploaded to the certificate store by performing a HTTP POST operation with a Content-Type as “multipart/form-data” and body to have the .efi file. The hash value to be generated and the text part “CryptographicHash” which specifies the hash algorithm to be used such as SHA256, SHA384, and SHA512 as listed here:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DB |
| iDRAC privilege | System Control |
| HTTP request method | POST |

**HTTP Request Header:**

Content-Type: “multipart/form-data"

**Multipart Text:**

CryptographicHash: A string providing the Cryptographic Hash value of SHA256, SHA384, and SHA512.

**Output:**

```
{
        "@Message.ExtendedInfo": [
                {
                        "Message": "Successfully Completed Request",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "Base.1.0.Success",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "None",
                        "Severity": "OK"
                },
                {
                        "Message": "The operation successfully completed.",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "iDRAC.1.6.SYS413",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "No response action is required.",
                        "Severity": "Informational"
                }
        ]
}
```

## *Viewing the Certificate or Hash*

The individual certificate or hash information under each certificate store can be viewed by performing the HTTP GET operation on each of the instances as shown here:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DB/iDRAC.Embedded.1%23CustSecbootpolicy.3 |
| iDRAC privilege | Login |
| HTTP request method | GET |

**Output:**

```
{
        "@odata.context": "/redfish/v1/$metadata#DellCertificate.DellCertificate",
        "@odata.id":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DB/iDRAC.Embedded.1%23CustS
ecbootpolicy.3",
        "@odata.type": "#DellCertificate.v1_0_0.DellCertificate",
        "CertificateSubtype": "Certificate",
        "CertificateType": "DB",
        "Description": "SecureBoot Certificate",
        "Id": "iDRAC.Embedded.1#CustSecbootpolicy.3",
        "IssuerCommonName_CN": "Microsoft Root Certificate Authority 2010",
        "IssuerCountryCode_CC": "US",
        "IssuerLocality_L": "Redmond",
        "IssuerOrganization_O": "Microsoft Corporation",
        "IssuerState_S": "Washington",
        "Name": "SecureBoot Certificate",
        "SecureBootPolicy": "Custom",
        "SerialNumber": "61077656000000000008",
        "SubjectCommonName_CN": "Microsoft Windows Production PCA 2011",
        "SubjectCountryCode_CC": "US",
        "SubjectLocality_L": "Redmond",
        "SubjectOrganization_O": "Microsoft Corporation",
        "SubjectState_S": "Washington",
        "Thumbprint": "58:0A:6F:4C:C4:E4:B6:69:B9: EB: DC: 1B:2B:3E:08:7B:80:D0:67:8D",
        "ValidFrom": "Oct 19 18:41:42 2011 GMT",
        "ValidTo": "Oct 19 18:51:42 2026 GMT"
}
```

| | |
|---|---|
| Resource ID | //redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DBX/iDRAC.Embedded.1%23CustSecbootpolicy.5 |
| iDRAC privilege | Login |
| HTTP request method | GET |

**Output:**

```
{
       "@odata.context": "/redfish/v1/$metadata#DellCertificate.DellCertificate",
       "@odata.id":
"/redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DBX/iDRAC.Embedded.1%23Cust
Secbootpolicy.5",
       "@odata.type": "#DellCertificate.v1_0_0.DellCertificate",
       "CertificateSubtype": "SHA256",
       "CertificateType": "DBX",
       "Description": "SecureBoot Certificate",
       "Hash": "416E3E4A6722A534AFBA9040B6D6A69CC313F1E48E7959F57BF248D543D00245",
       "Id": "iDRAC.Embedded.1#CustSecbootpolicy.5",
       "Name": "SecureBoot Certificate",
       "SecureBootPolicy": "Custom"
}
```

## *Downloading the Certificate or Hash*

The individual certificate or hash under each certificate store can be downloaded by performing the HTTP GET operation on each of the instances with the Accept header as given here:

### Certificate:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DB/iDRAC.Embedded.1%23CustSecbootpolicy.3 |
| iDRAC privilege | Login |
| HTTP request method | GET |

**HTTP Request Header:**

**Accept:** `application/pkix-cert`

**Output:**

The certificate file will be downloaded.

### Hash:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DB/iDRAC.Embedded.1%23CustSecbootpolicy.5 |
| iDRAC privilege | Login |
| HTTP request method | GET |

**HTTP Request Header:**

**Accept:** `application/octet-stream`

**Output:**

The hash file will be downloaded.

## *Deleting the Certificate or Hash*

The individual certificate or hash under each certificate store can be removed by performing the HTTP DELETE operation on each of the instances as given here:

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Certificates/DB/iDRAC.Embedded.1%23CustSecbootpolicy.3 |
| iDRAC privilege | System Control |
| HTTP request method | DELETE |

**Output:**

```
{
        "@Message.ExtendedInfo": [
                {
                        "Message": "Successfully Completed Request",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "Base.1.0.Success",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "None",
                        "Severity": "OK"
                },
                {
                        "Message": "The operation successfully completed.",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "iDRAC.1.6.SYS413",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "No response action is required.",
                        "Severity": "Informational"
                }
        ]
}
```

## *Resetting the Secure Boot Keys*

All the Secure Boot certificates can be reset to default, all the Secure Boot certificates can be deleted, the Secure Boot certificate stores can deleted, or  reset by performing the POST on the following URI with allowable values as "ResetAllKeysToDefault", "DeleteAllKeys", "DeletePK", "ResetPK", "ResetKEK", "ResetDB", and "ResetDBX".

| | |
|---|---|
| Resource ID | /redfish/v1/Systems/System.Embedded.1/SecureBoot/Actions/SecureBoot.ResetKeys |
| iDRAC privilege | System Control |
| HTTP request method | POST |

### HTTP Request Body:

```
{
        "ResetKeysType": "ResetAllKeysToDefault"
}
```

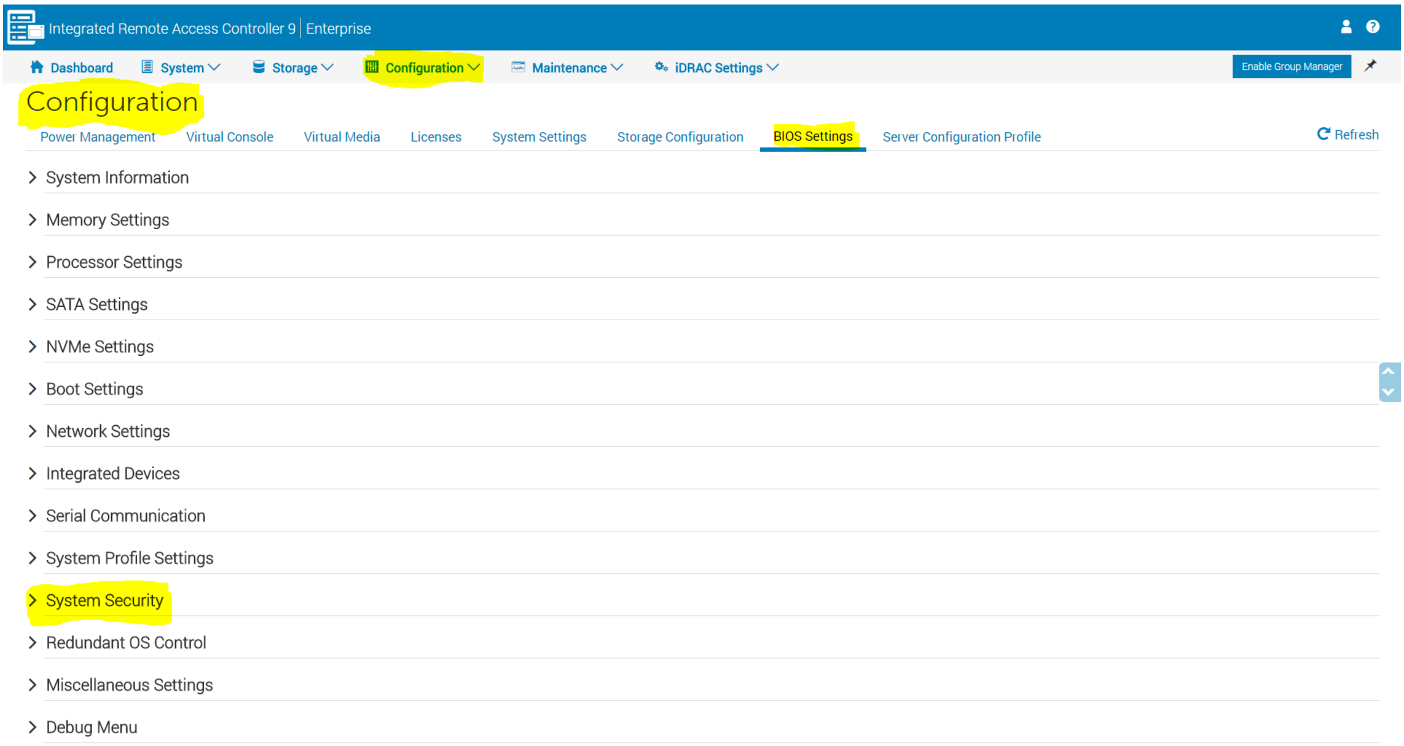
**Output:**

```
{
        "@Message.ExtendedInfo": [
                {
                        "Message": "Successfully Completed Request",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "Base.1.0.Success",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "None",
                        "Severity": "OK"
                },
                {
                        "Message": "The operation successfully completed.",
                        "MessageArgs": [],
                        "MessageArgs@odata.count": 0,
                        "MessageId": "iDRAC.1.6.SYS413",
                        "RelatedProperties": [],
                        "RelatedProperties@odata.count": 0,
                        "Resolution": "No response action is required.",
                        "Severity": "Informational"
                }
        ]
}
```

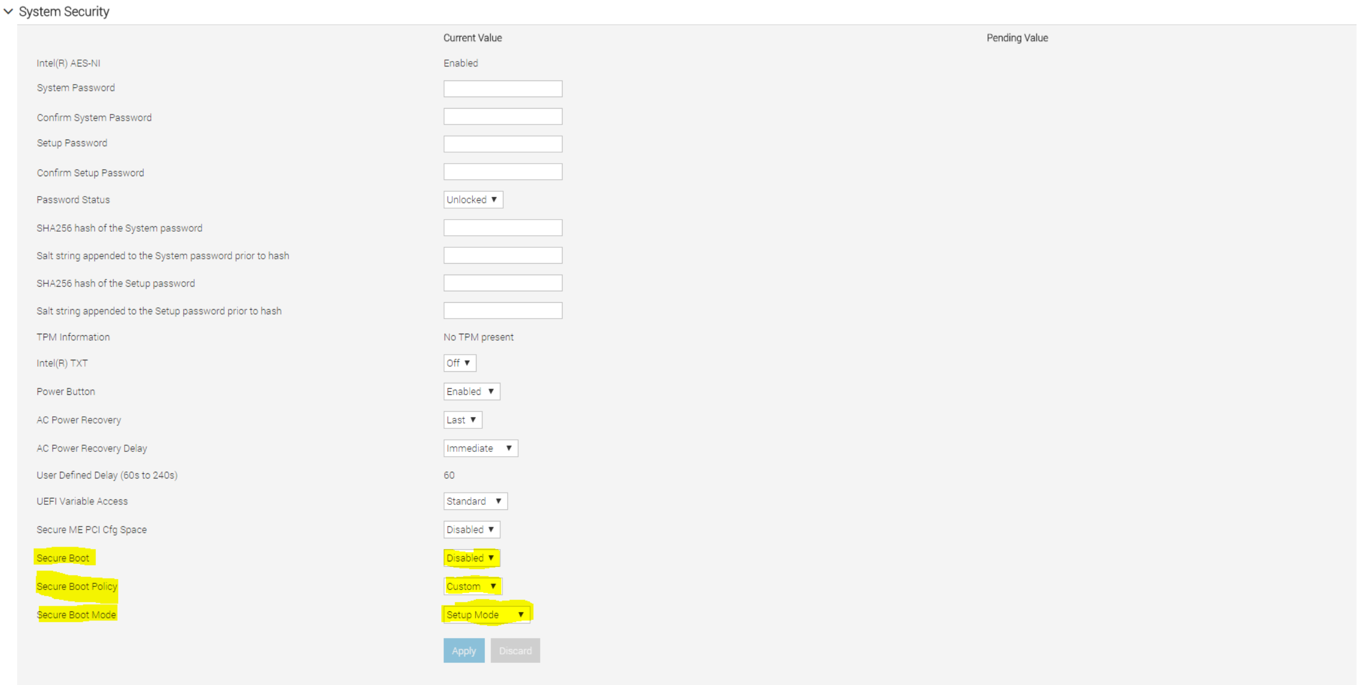
# Secure Boot certificate management using iDRAC UI

Secure Boot configuration in iDRAC GUI can be reached by traversing the GUI through **Configuration** > **BIOS Settings** > **System Security**. The GUI provides the options to configure Secure Boot to Enabled/Disabled, Secure Boot Policy to Standard/Custom, and Secure Boot Mode to Setup Mode/ User Mode / Deployed Mode/ Audit Mode.

*Currently, the iDRAC GUI does not provide options to configure or manage the Secure Boot certificates.*



These changes become effective only when you restart the host server. After making the required changes, make sure the host is restarted so that the requested changes are applied.



System Security

| | Current Value | Pending Value |
|---|---|---|
| Intel(R) AES-NI | Enabled | |
| System Password | | |
| Confirm System Password | | |
| Setup Password | | |
| Confirm Setup Password | | |
| Password Status | Unlocked ▼ | |
| SHA256 hash of the System password | | |
| Salt string appended to the System password prior to hash | | |
| SHA256 hash of the Setup password | | |
| Salt string appended to the Setup password prior to hash | | |
| TPM Information | No TPM present | |
| Intel(R) TXT | Off ▼ | |
| Power Button | Enabled ▼ | |
| AC Power Recovery | Last ▼ | |
| AC Power Recovery Delay | Immediate ▼ | |
| User Defined Delay (60s to 240s) | 60 | |
| UEFI Variable Access | Standard ▼ | |
| Secure ME PCI Cfg Space | Disabled ▼ | |
| Secure Boot | Disabled ▼ | |
| Secure Boot Policy | Custom ▼ | |
| Secure Boot Mode | Setup Mode ▼ | |