

Secureboot certificate management by 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 of the Secure Boot certificate management, the certificates are stored in the various certificate stores such as PK, KEK, DB and DBX. These can be 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%23CustSecbootpolicy.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

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

here:

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%23CustSecbootpolicy.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": "6107765600000000000008",
  "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%23CustSecbootpolicy.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 be 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"
    }
  ]
}
```