PowerShell Get-Help on command 'Set-SecureBootUEFI'

PS C:\Users\wahid> Get-Help Set-SecureBootUEFI

```
NAME
    Set-SecureBootUEFI

SYNOPSIS
    Sets the Secure Boot-related UEFI variables.


SYNTAX
    Set-SecureBootUEFI [-AppendWrite] [-Content <Byte[]>] -Name {PK | KEK | db |
    dbx} [-OutputFilePath <String>] [-SignedFilePath <String>] -Time <String>
    [<CommonParameters>]

    Set-SecureBootUEFI [-AppendWrite] [-ContentFilePath <String>] -Name {PK | KEK
    | db | dbx} [-OutputFilePath <String>] [-SignedFilePath <String>] -Time
    <String> [<CommonParameters>]


DESCRIPTION
    The Set-SecureBootUEFI cmdlet takes a formatted content object that is created
    by running the Format-SecureBootUEFI cmdlet and a signed file, combines the
    two and then attempts to set the package in one of the Secure Boot variables.
```

The supported Secure Boot variables include Platform Key (PK), Key Exchange Key (KEK), Signature Database (DB), and Forbidden Signature Database (DBX).

If successful, this cmdlet returns a UEFIEnvironmentVariable object. Otherwise, it displays an error.

This cmdlet runs on both UEFI and BIOS (non-UEFI) computers. If the computer does not support Secure Boot or is a non-UEFI computer, this cmdlet displays the following:

`Cmdlet not supported on this platform.`

If you do not run Windows PowerShellr in administrator mode, this cmdlet displays the following:

`Unable to set proper privileges. Access was denied.`

If you supply a signed file to this cmdlet that is not valid, this cmdlet displays the following:

`Incorrect authentication data.`

PARAMETERS

-AppendWrite [<SwitchParameter>]

Indicates that the contents of the current variable are appended instead of overwritten.

-Content <Byte[]>

Specifies the byte contents of the variable that is being set.

-ContentFilePath <String>

Specifies the file that contains the contents that is being set to the environment variable.

If you specify only the name, the file must be in the current working directory. Otherwise, specify the full path of the file.

-Name <String>

Specifies the name of the UEFI environment variable. The acceptable values for this parameter are: PK, KEK, DB, and DBX.

-OutputFilePath <String>

Specifies the name of the file created that contains the contents of what is set. If you specify this parameter, instead of setting the variable, the cmdlet stores the contents in this file.

The file is created in the specified path location.

-SignedFilePath <String>

Specifies the signed data that is paired with the contents that are being set to the environment variable.

If you specify only the name, the file must be in the current working directory. Otherwise, specify the full path of the file.

-Time <String>

Specifies the timestamp that is used in the signature. Format this value as follows so that it is accepted as a DateTime object:

`"2011-11-01T13:30:00Z"`

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

------------- Example 1: Set the DBX UEFI variable -------------

```
PS C:\> $ObjectFromFormat = ( Format-SecureBootUEFI -Name DBX -SignatureOwner
12345678-1234-1234-1234-123456789abc -Algorithm SHA256 -Hash
0011223344556677889900112233445566778899001122334455667788990011
-SignableFilePath GeneratedFileToSign.bin -Time 2011-11-01T13:30:00Z
-AppendWrite )
PS C:\> .\signtool.exe sign /fd sha256 /p7 .\ /p7co 1.2.840.113549.1.7.1 /p7ce
DetachedSignedData /a /f PrivateKey.pfx GeneratedFileToSign.bin
PS C:\> $ObjectFromFormat | Set-SecureBootUEFI -SignedFilePath
GeneratedFileToSign.bin.p7
Name       : dbx
Bytes      : {161, 89, 192, 165...}
Attributes : NON VOLATILE
             BOOTSERVICE ACCESS
             RUNTIME ACCESS
             TIME BASED AUTHENTICATED WRITE ACCESS
```

This example sets the information obtained from the Format-SecureBootUEFI cmdlet to the DBX UEFI variable.

The first command supplies a path to the signed package to be authenticated. The file named GeneratedFileToSign.bin is a digest created by the Format-SecureBootUEFI cmdlet that needs to be signed according to the UEFI specification.

The second command runs the SignTool.exe tool from the current directory to sign the digest. The SignTool.exe tool can be downloaded from Windows Software Development Kit (SDK) for Windows 8 (https://go.microsoft.com/fwlink/p/?LinkId=236500)on MSDN.

The third command sets the information.

Example 2: Set the DBX UEFI variable by using a signed package

```
PS C:\> Set-SecureBootUEFI -ContentFilePath FormattedVariable.bin
-SignedFilePath GeneratedFileToSign.bin.p7
Name       : dbx
Bytes      : {161, 89, 192, 165...}
Attributes : NON VOLATILE
             BOOTSERVICE ACCESS
             RUNTIME ACCESS
             TIME BASED AUTHENTICATED WRITE ACCESS
```

This command sets the formatted data that was written to file FormattedVariable.bin to the DBX UEFI variable. This cmdlet supplies a path to the signed package to be authenticated.

- Example 3: Set the DBX UEFI variable by using unsigned data -

```
PS C:\> $objectFromFormat = ( Format-SecureBootUEFI -Name DB -SignatureOwner
12345678-1234-1234-1234-123456789abc -Time 2011-11-01T13:30:00Z
-CertificateFilePath db.cer -FormatWithCert )
PS C:\> $objectFromFormat | Set-SecureBootUEFI
Name       : db
Bytes      : {161, 89, 192, 165...}
Attributes : NON VOLATILE
             BOOTSERVICE ACCESS
             RUNTIME ACCESS
             TIME BASED AUTHENTICATED WRITE ACCESS
```

This example creates formatted data that is not signed and sets the unsigned data into the UEFI variable named db.

REMARKS

To see the examples, type: "get-help Set-SecureBootUEFI -examples".

For more information, type: "get-help Set-SecureBootUEFI -detailed".

For technical information, type: "get-help Set-SecureBootUEFI -full".

For online help, type: "get-help Set-SecureBootUEFI -online"