Advanced UEFI security features for Surface Pro 3

Article • 01/25/2023 • 4 minutes to read

This article describes how to install and configure the v3.11.760.0 UEFI update to enable additional security options for Surface Pro 3 devices.

To address more granular control over the security of Surface devices, the v3.11.760.0 UEFI update provides additional security options that allow you to disable specific hardware devices or to prevent starting from those devices. After the UEFI update is installed on a device, you can configure it manually or automatically by running a script.

Manually install the UEFI update

Before you can configure the advanced security features of your Surface device, you must first install the v3.11.760.0 UEFI update. This update is installed automatically if you receive your updates from Windows Update. For more information about how to configure Windows to update automatically by using Windows Update, see How to configure and use Automatic Updates in Windows .

To update the UEFI on Surface Pro 3, you can download and install the Surface UEFI updates as part of the Surface Pro 3 Firmware and Driver Pack. These firmware and driver packs are available from the Surface Pro 3 page on the Microsoft Download Center. You can find out more about the firmware and driver packs at Download drivers and firmware for Surface . The firmware and driver packs are available as both self-contained Windows Installer (.msi) and archive (.zip) formats. You can find out more about these two formats and how you can use them to update your drivers at Manage and deploy Surface driver and firmware updates.

Manually configure additional security settings

Note

To enter firmware setup on a Surface device, begin with the device powered off, press and hold the **Volume Up** button, then press and release the **Power** button, then release the **Volume Up** button after the device has begun to boot.

After the v3.11.760.0 UEFI update is installed on a Surface device, an additional UEFI menu named **Advanced Device Security** becomes available. If you click this menu, the following options are displayed:

Option	Description	Available settings (default listed in bold)
Network Boot	Enables or disables the ability of your Surface device to boot from the network (also known as PXE boot).	**Enabled**, Not Bootable
Side USB	Enables or disables the USB port on the side of the Surface device. Additionally, the USB port can be enabled, but not allow booting.	**Enabled**, Not Bootable, Disabled
Docking Port	Enables or disables the ports on the Surface docking station. Additionally, the docking port can be enabled, but block booting from any USB or Ethernet port in the docking station.	**Enabled**, Not Bootable, Disabled
Front Camera	Enables or disables the camera on the front of the Surface device.	**Enabled**, Disabled
Rear Camera	Enables or disables the camera on the rear of the Surface device.	**Enabled**, Disabled
On Board Audio	Enables or disables audio on the Surface device.	**Enabled**, Disabled
microSD	Enables or disables the microSD slot on the Surface device.	**Enabled**, Disabled
WiFi	Enables or disables the built-in Wi-Fi transceiver in the Surface device. This also disables Bluetooth.	**Enabled**, Disabled
Bluetooth	Enables or disables the built-in Bluetooth transceiver in the Surface device.	**Enabled**, Disabled

Automate additional security settings

As an IT professional with administrative privileges, you can automate the configuration of UEFI settings by leveraging Surface Pro 3 Firmware Tools (476 KB) available from the Microsoft Download Center. These tools install a .NET assembly that can be called from any custom application or script.

Prerequisites

- The sample scripts below leverage the previously mentioned extension and therefore assume that the tool has been installed on the device being managed.
- The scripts must be run with administrative privilege.
- The Windows PowerShell command **Set-ExecutionPolicy Unrestricted** must be called prior to running sample scripts if they are not digitally signed.

Sample scripts

Note

The UEFI password used in the sample scripts below is presented in clear text. We strongly recommend saving the scripts in a protected location and running them in a controlled environment.

Show all configurable options:

PowerShell

```
# Load the extension
[System.Reflection.Assembly]::Load("SurfaceUefiManager,
Version=1.0.5483.22783, Culture=neutral, PublicKeyToken=20606f4b5276c705")

# Get the collection of all configurable settings
$uefiOptions = [Microsoft.Surface.FirmwareOption]::All()

foreach ($uefiOption in $uefiOptions)
{
    Write-Host "Name:" $uefiOption.Name
    Write-Host " Description =" $uefiOption.Description
    Write-Host " Current Value =" $uefiOption.CurrentValue
    Write-Host " Default Value =" $uefiOption.DefaultValue
    Write-Host " Proposed Value =" $uefiOption.ProposedValue

    # This gives usage and validation information
    Write-Host " Allowed Values =" $uefiOption.FriendlyRegEx
    Write-Host " Regular Expression =" $uefiOption.RegEx
    Write-Host
}
```

Set or change UEFI password:

PowerShell

```powershell
# Load the extension
[System.Reflection.Assembly]::Load("SurfaceUefiManager,
Version=1.0.5483.22783, Culture=neutral, PublicKeyToken=20606f4b5276c705")

# Must supply UEFI administrator Password if set
# If it is not currently set this is ignored
[Microsoft.Surface.FirmwareOption]::Unlock("1234")

$Password = [Microsoft.Surface.FirmwareOption]::Find("Password")

# Set New value to 12345
$Password.ProposedValue = "12345"
```

Check status of proposed changes:

PowerShell

```powershell
# Load the extension
[System.Reflection.Assembly]::Load("SurfaceUefiManager,
Version=1.0.5483.22783, Culture=neutral, PublicKeyToken=20606f4b5276c705")

# Check update status
$updateStatus = [Microsoft.Surface.FirmwareOption]::UpdateStatus
$updateIteration = [Microsoft.Surface.FirmwareOption]::UpdateIteration
Write-Host "Last Update Status =" $updateStatus
Write-Host "Last Update Iteration =" $updateIteration

# Get the individual results for the last proposed update
# If the device has never had an update attempt this will be an empty list
$details = [Microsoft.Surface.FirmwareOption]::UpdateStatusDetails
Write-Host $details.Count "Settings were proposed"
if ($details.Count -gt 0)
{
    Write-Host "Result Details"
    foreach ($detail in $details.GetEnumerator())
    {
        Write-Host " " $detail.Key "=" $detail.Value
    }
}
```

Revert UEFI to default values:

PowerShell

```
# Load the extension
[System.Reflection.Assembly]::Load("SurfaceUefiManager,
Version=1.0.5483.22783, Culture=neutral, PublicKeyToken=20606f4b5276c705")

# Must supply UEFI administrator Password if set
# If it is not currently set this is ignored
[Microsoft.Surface.FirmwareOption]::Unlock("1234")

# Get the collection of all configurable settings
$uefiOptions = [Microsoft.Surface.FirmwareOption]::All()

# Reset all options to the factory default
foreach ($uefiOption in $uefiOptions)
{
    $uefiOption.ProposedValue = $uefiOption.DefaultValue
}
```

Status code interpretation

- 00 - The proposed update was a success
- 02 - One of the proposed values had an invalid value
- 03 - There was a proposed value set that was not recognized
- 0F - The unlock password did not match currently set password