



Virtual Secure Mode und Device Guard

Das jetzt nicht mehr neue Sicherheitskonzept von Microsoft namens Device Guard läuft ab Windows 10 Enterprise und Server 2016.

Mit dieser Technik können Hardware- und Softwarefeatures so eingesetzt, das z.B. nur vertrauenswürdige Anwendungen ausgeführt werden können.

Device Guard setzt isolierten Umgebungen (Container) ein, um Anwendungen auszuführen und voneinander zu trennen. Als Basis für die Umsetzung dient der Hypervisor von Microsoft.

Hardwarevoraussetzungen:

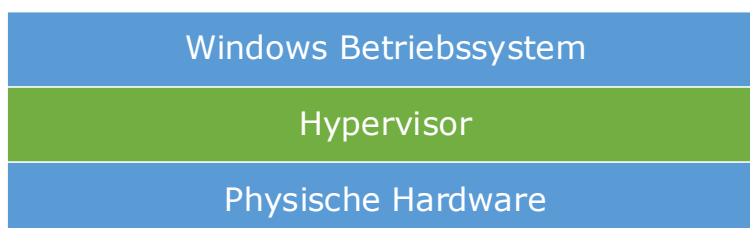
- UEFI ab v2.3.1 im einheitlichen Modus
- Windows 64-Bit
- Second Layer Address Translation SLAT
- Virtualisierungserweiterung z.B. Intel VT oder AMD-V
- Optional: Trusted Platform Modul TPM
- Optional: Andere Bootmedien sollten deaktiviert werden

Für die Nutzung von Device Guard muss zuerst der Virtual Secure Mode aktiviert werden. Der VSM ist eine Funktion der die Virtualisierungserweiterung des Prozessors einsetzt, um so die Sicherheit zu erhöhen. Geschützt werden dabei unter anderem die Daten im Speicher und es wird sichergestellt, dass jede Instanz nur auf die eigenen Daten zugreifen kann.

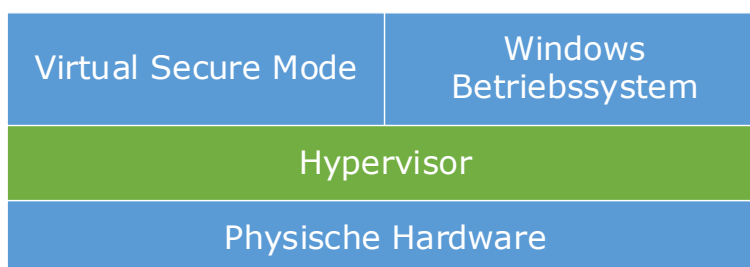
Auch der Credential Guard benötigt für die Ausführung den Virtual Secure Mode, aber der ist heute nicht mein Thema.

Der Aufbau wäre bildlich so darzustellen:

Der Hypervisor dient als Zwischenschicht, um so die sicherheitsrelevanten Prozesse getrennt vom Betriebssystem auslagern zu können.



Auf diese Weise ist die Virtual Secure Mode Instanz geschützt und wird getrennt vom Betriebssystem (Kernel) verwaltet.



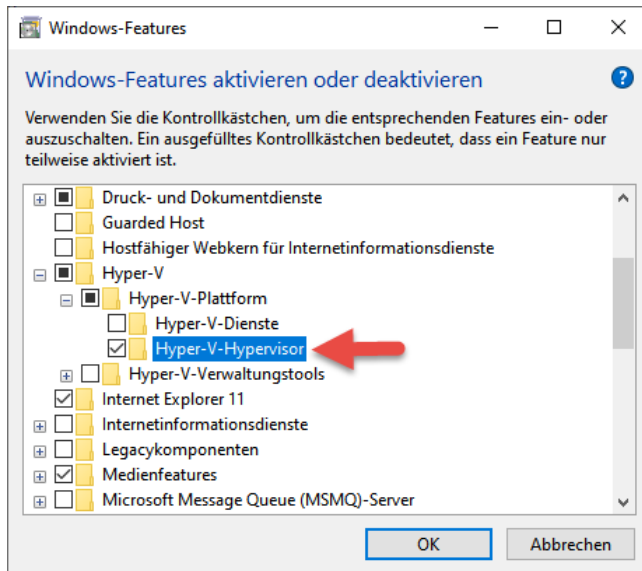


Virtual Secure Mode und Device Guard

Kommen wir zur Aktivierung des Virtual Secure Mode. Zunächst muss der Hypervisor installiert werden. Die Managementtools werden nicht benötigt.

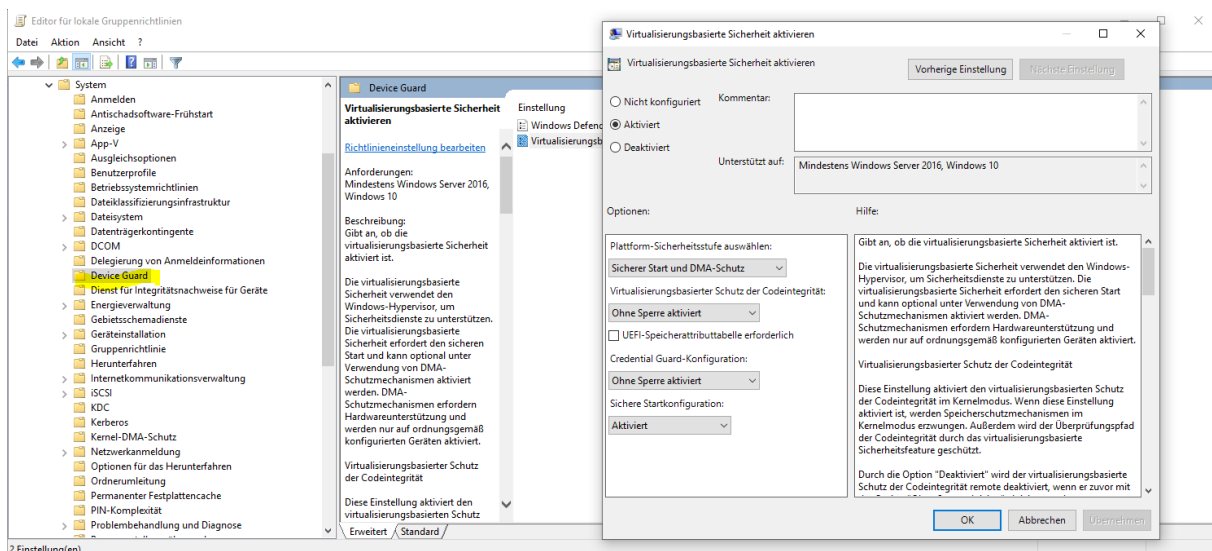
Install-WindowsFeature -Name Hyper-V

-IncludeManagementTools



Neustart des Systems.

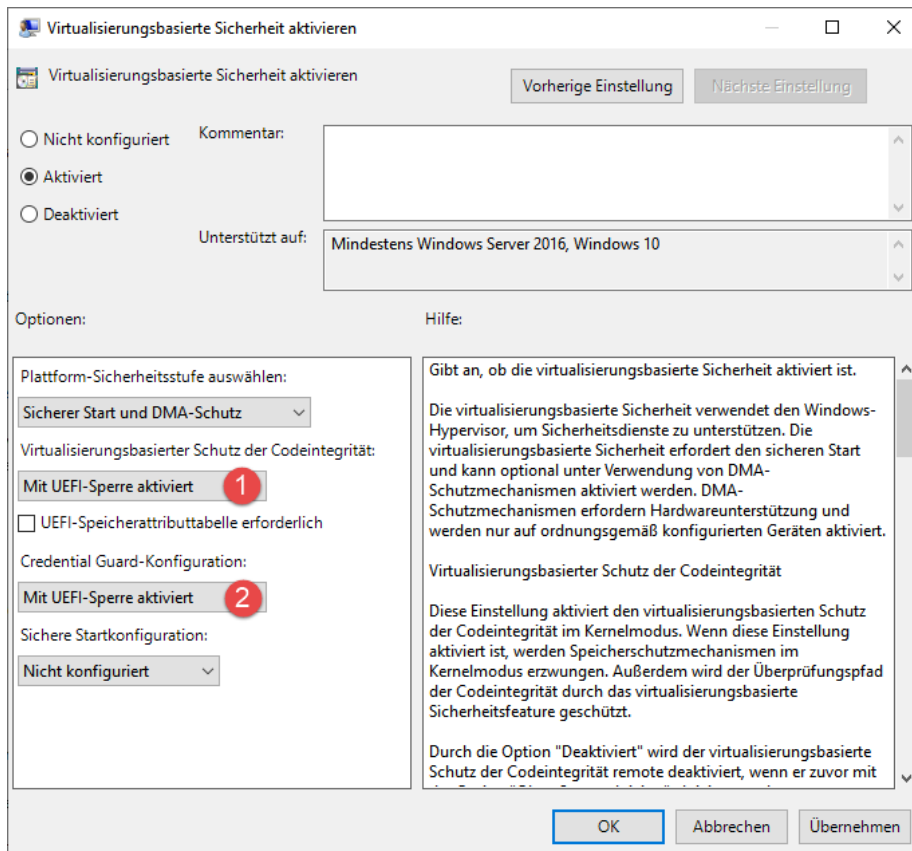
Zum konfigurieren des Secure Modes muss noch eine Gruppenrichtlinie erstellt bzw. aktiviert werden.



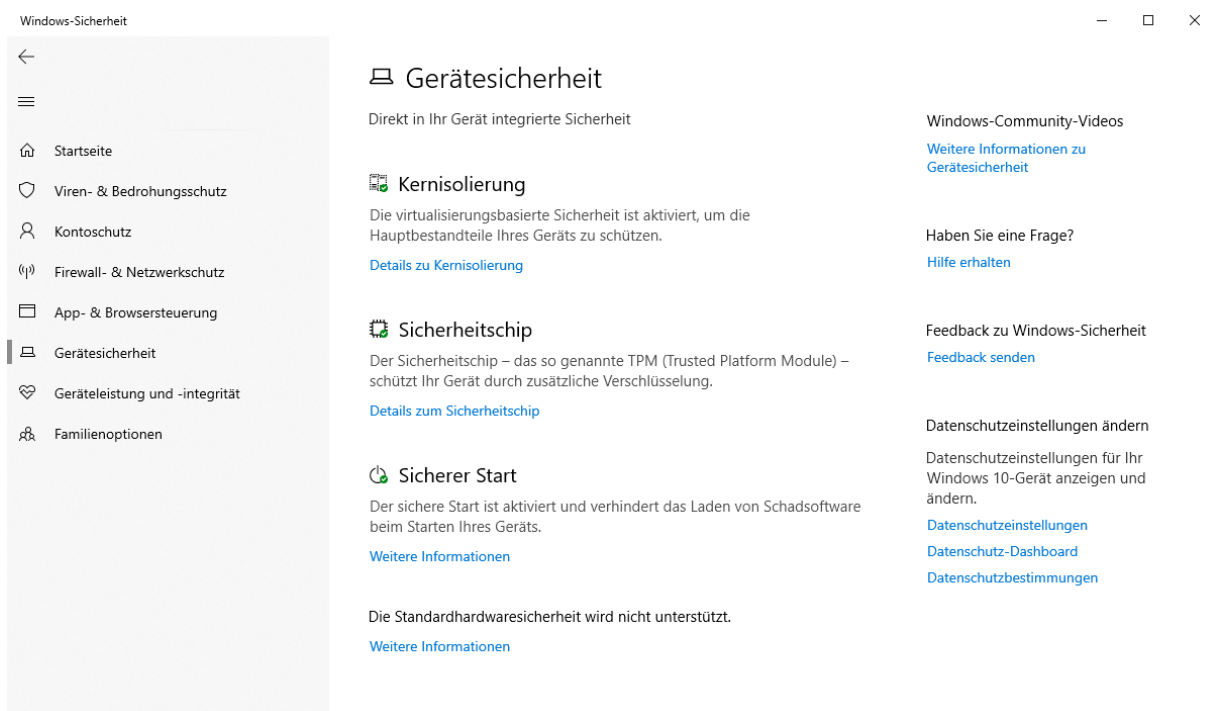


Virtual Secure Mode und Device Guard

Würden diese beiden Optionen aktiviert werden, würde man damit eine Remoteabschaltung verhindern.



Device Guard ist nun für den Einsatz bereit.





Virtual Secure Mode und Device Guard

Prüfen ob Device Guard läuft:

```
Get-CimInstance -ClassName Win32_DeviceGuard -Namespace root\Microsoft\Windows\DeviceGuard
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\Administrator> Get-CimInstance -ClassName Win32_DeviceGuard -Namespace root\Microsoft\Windows\DeviceGuard

AvailableSecurityProperties           : {1}
CodeIntegrityPolicyEnforcementStatus : 0
InstanceIdentifier                   : 4ff40742-2649-41b8-bdd1-e80fad1cce80
RequiredSecurityProperties            : {1, 2, 3}
SecurityServicesConfigured           : {1, 2}
SecurityServicesRunning               : {0}
UsermodeCodeIntegrityPolicyEnforcementStatus : 0
Version                               : 1.0
VirtualizationBasedSecurityStatus    : 2
PSComputerName                        :
```

Properties	Description	Valid values
AvailableSecurityProperties	This field helps to enumerate and report state on the relevant security properties for Device Guard.	<ul style="list-style-type: none">○ 0. If present, no relevant properties exist on the device.○ 1. If present, hypervisor support is available.○ 2. If present, Secure Boot is available.○ 3. If present, DMA protection is available.○ 4. If present, Secure Memory Overwrite is available.○ 5. If present, NX protections are available.○ 6. If present, SMM mitigations are available. <p>Note: 4, 5, and 6 were added as of Windows 10, version 1607.</p>
InstanceIdentifier	A string that is unique to a particular device.	Determined by WMI.
RequiredSecurityProperties	This field describes the required security properties to enable virtualization-based security.	<ul style="list-style-type: none">○ 0. Nothing is required.○ 1. If present, hypervisor support is needed.



Virtual Secure Mode und Device Guard

		<ul style="list-style-type: none"> ○ 2. If present, Secure Boot is needed. ○ 3. If present, DMA protection is needed. ○ 4. If present, Secure Memory Overwrite is needed. ○ 5. If present, NX protections are needed. ○ 6. If present, SMM mitigations are needed. <p>Note: 4, 5, and 6 were added as of Windows 10, version 1607.</p>
SecurityServicesConfigured	This field indicates whether the Credential Guard or HVCI service has been configured.	<ul style="list-style-type: none"> ○ 0. No services configured. ○ 1. If present, Credential Guard is configured. ○ 2. If present, HVCI is configured.
SecurityServicesRunning	This field indicates whether the Credential Guard or HVCI service is running.	<ul style="list-style-type: none"> ○ 0. No services running. ○ 1. If present, Credential Guard is running. ○ 2. If present, HVCI is running.
Version	This field lists the version of this WMI class.	The only valid value now is 1.0.
VirtualizationBasedSecurityStatus	This field indicates whether VBS is enabled and running.	<ul style="list-style-type: none"> ○ 0. VBS is not enabled. ○ 1. VBS is enabled but not running. ○ 2. VBS is enabled and running.
PSComputerName	This field lists the computer name.	All valid values for computer name.

Registry – die Einstellungen finden wir hier:

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\DeviceGuard



Virtual Secure Mode und Device Guard

Das Ziel ist es nun, nur noch die Anwendungen laufen zu lassen, die über eine Richtlinie vorab definiert wurden (Whitelisting).

Solche Richtlinien heißen „Codeintegritätsrichtlinien“. Codeintegritätsrichtlinien schützen uns vor ungewollten und gefährlichen Anwendungen.

Code-Integrität = Windows Defender Application Control

Microsoft stellt eine Anleitung zum Aufbau einer CI Policy bereit:

<https://docs.microsoft.com/en-us/powershell/module/configci/new-cipolicy?view=win10-ps>

<https://docs.microsoft.com/en-us/powershell/module/configci/set-ruleoption?view=win10-ps>

<https://docs.microsoft.com/de-de/windows/security/threat-protection/windows-defender-application-control/create-initial-default-policy>

Nach dem Erstellen einer solchen Richtlinie, muss diese auf dem oder die Computer bereitgestellt gestellt werden.

Bei einer zentralisierten Bereitstellung sollte ein Netzwerkpfad angegeben werden. Handelt es sich um eine Standalone Maschine z.B. einem Kiosk-Rechner, so kann die Policy lokal liegen, aber bitte mit den entsprechenden NTFS-Rechten geschützt.

Windows Defender-Anwendungssteuerung bereitstellen

Windows Defender-Anwendungssteuerung bereitstellen

Vorherige Einstellung Nächste Einstellung

Nicht konfiguriert Kommentar:

Aktiviert

Deaktiviert

Unterstützt auf: Mindestens Windows Server 2016, Windows 10

Optionen:

Dateipfad für Codeintegritätsrichtlinie:

Hilfe:

Windows Defender-Anwendungssteuerung bereitstellen

Mit dieser Richtlinieneinstellung können Sie eine Codeintegritätsrichtlinie auf einem Computer bereitstellen, um zu steuern, was auf diesem Computer ausgeführt werden darf.

Wenn Sie eine Codeintegritätsrichtlinie bereitstellen, wird der im Kernelmodus und auf dem Windows-Desktop ausführbare Code auf Grundlage der Richtlinie durch Windows eingeschränkt. Zum Aktivieren dieser Richtlinie muss der Computer neu gestartet werden.

Der Dateipfad muss entweder ein UNC-Pfad (z. B. \\ServerName\ShareName\SIPolicy.p7b) oder ein gültiger lokaler Pfad (z. B. C:\FolderName\SIPolicy.p7b) sein. Das lokale Computerkonto (LOCAL SYSTEM) muss über eine Zugriffsberechtigung für die Richtliniendatei verfügen.

Bei Verwendung einer signierten, geschützten Richtlinie wird die Funktion nicht vom Computer entfernt, wenn die Richtlinieneinstellung deaktiviert wird. Führen Sie stattdessen

OK Abbrechen Übernehmen



Virtual Secure Mode und Device Guard

Beispiel:

Möchten ich einen Kiosk-Rechner schützen, der fertig installiert und konfiguriert ist, so kann ich mir aus dem aktuellen Stand der Maschine eine Policy erstellen lassen:

Die Powershell mit administrativen Rechten öffnen.

Schritt 1)

Erstellen einer neuen Policy.

```
$dg_policy = "C:\Policy\CI\policy.xml"
```

```
$dg_policyBIN = "C:\Policy\CI\policy.p7b"
```

```
New-CIPolicy -FilePath $dg_policy -UserPEs -Level Publisher -Fallback Hash
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

"Überprüfung wird ausgeführt... Dies kann einen Moment dauern."
C:\Windows\WinSxS\x86_netFx4-system_core_ni_b03f5f7f11d50a3a_4.0.15713.0_none_86fabcf23d8eb51c\system.core.ni.d11
```

Schritt 2)

Nun deaktivieren wir den Audit Mode

```
Set-RuleOption -FilePath "C:\Policy\CI\policy.xml" -Option 3 -Delete
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Set-RuleOption -FilePath "C:\Policy\CI\policy.xml" -Option 3 -Delete
PS C:\WINDOWS\system32>
```

Schritt 3)

Konvertieren die neue Policy *.xml in eine binäre Datei.

```
convertfrom-CIPolicy $dg_policy $dg_policyBIN
```

```
#ConvertFrom-CIPolicy -XmlFilePath "C:\Policy\CI\policy.xml" -BinaryFilePath  
"C:\Policy\CI\policy.p7b"
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\WINDOWS\system32> ConvertFrom-CIPolicy -XmlFilePath "C:\Policy\CI\policy.xml" -BinaryFilePath "C:\Policy\CI\policy.p7b"
C:\Policy\CI\policy.p7b
PS C:\WINDOWS\system32>
```



Virtual Secure Mode und Device Guard

Mit diesem Befehl wird nach signierten Binärdaten gesucht, ausführbaren Dateien oder DLLs.

```
$dg_policy = "C:\Policy\CI\policy.xml"
```

```
$dg_policyBIN = "C:\Policy\CI\policy.bin"
```

```
New-CIPolicy -Level PcaCertificate -FilePath $dg_policy -UserPES
```

```
convertfrom-CIPolicy $dg_policy $dg_policyBIN
```

```
Administrator: Windows PowerShell ISE
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe
Unbenannt1.ps1* X DG_Readiness_Tool_v3.6.ps1*
1 $dg_policy = "C:\Policy\CI\policy.xml"
2 $dg_policyBIN = "C:\Policy\CI\policy.bin"
3 New-CIPolicy -Level PcaCertificate -FilePath $dg_policy -UserPES
4 convertfrom-CIPolicy $dg_policy $dg_policyBIN
5
"Überprüfung wird ausgeführt... Dies kann einen Moment dauern."
C:\Windows\WinSxS\x86_mscorlib_b77a5c561934e089_4.0.15713.110_none_5f1850654cf5ab48\mscorlib.dll.
Skript/Auswahl wird ausgeführt. Drücken Sie "Strg+Unterbrechen", um den Vorgang zu beenden, und "", um den Debugger zu öffnen.
Ln 6 Spalte 1 100%
```

Die Events in der Ereignisanzeige finden wir an diesen Stellen:

The screenshot shows the Windows Event Viewer interface. The left pane displays the event log hierarchy, with 'DeviceGuard' expanded under 'Operational'. The main pane shows a list of three events:

Ebene	Datum und Uhrzeit	Quelle	Ereignis-ID	Aufgabenkatego...
Informationen	30.06.2019 15:52:06 Uhr	DeviceGuard	7000	Keine
Informationen	30.06.2019 15:51:33 Uhr	DeviceGuard	7000	Keine
Informationen	30.06.2019 15:51:33 Uhr	DeviceGuard	7000	Keine

The details pane for event ID 7000 from DeviceGuard shows the following text:

Die Gruppenrichtlinie wurde erfolgreich von Device Guard verarbeitet: Virtualisierungsbasierte Sicherheit = Aktiviert, Sicherer Start = Ein, DMA-Schutz = Ein, Virtualisierungsbasierte Codeintegrität = Aktiviert, Credential Guard = Aktiviert, Neustart erforderlich = Nein, Status = 0x0.

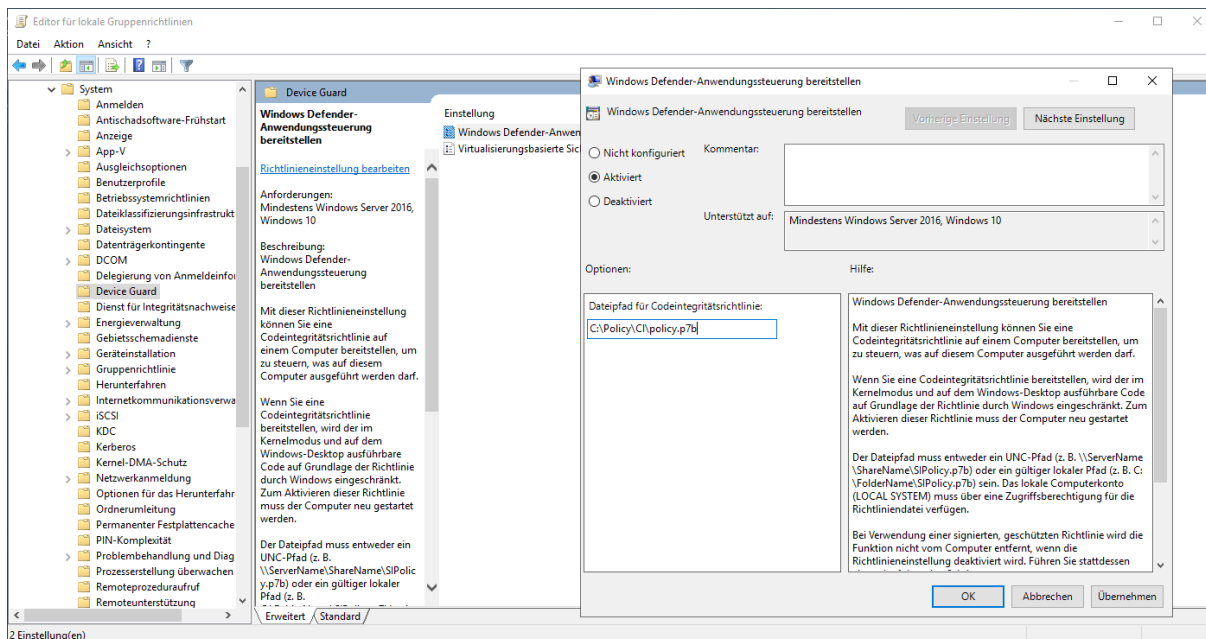
Below the text, the following properties are listed:

- Protokollname: Microsoft-Windows-DeviceGuard/Operational
- Quelle: DeviceGuard
- Ereignis-ID: 7000
- Ebene: Informationen
- Benutzer: SYSTEM
- Vorgangscodename: Info
- Protokolliert: 30.06.2019 15:52:06 Uhr
- Aufgabenkategorie: Keine
- Schlüsselwörter: DELL7450
- Weitere Informationen: [Onlinehilfe](#)

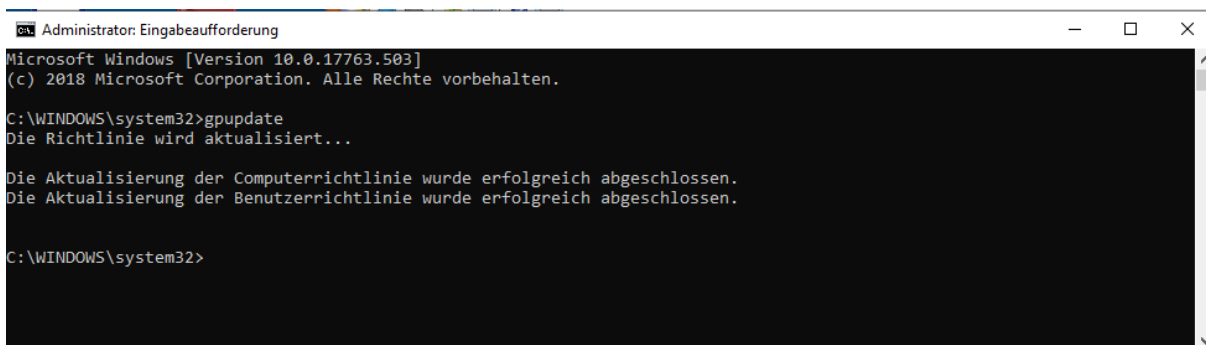


Virtual Secure Mode und Device Guard

Nachdem die erhobenen Informationen als .xml Datei vorliegt, wird diese im letzten Schritt in eine Binärdatei *.p7b umgewandelt. Die Policy ist nun fertig für den Einsatz in einer Gruppenrichtlinie.



Gruppenrichtlinienupdate durchführen und neu starten. Doppelt hält besser!



Starte ich jetzt eine Applikation die nicht in der Policy als Whitelisting enthalten ist, so wird die Ausführung blockiert.





Virtual Secure Mode und Device Guard

Ihre Organisation hat diese App mithilfe der Windows Defender-Anwendungssteuerung blockiert.

C:\Users\Joern\Desktop\CCleaner_5.8.exe

Weitere Informationen erhalten Sie vom Support.

In Zwischenablage kopieren

Schließen

Im Log erscheint eine entsprechende Fehlermeldung.

The screenshot shows the Windows Event Viewer window titled 'Ereignisanzeige'. The left pane shows the tree view with 'Operational' selected under 'DeviceGuard'. The main pane displays a list of events with the following data:

Ebene	Datum und Uhrzeit	Quelle	Ereignis-ID	Aufgabenkategorie
Informationen	30.06.2019 18:44:02 Uhr	CodeIntegrity	3084 (20)	
Informationen	30.06.2019 18:42:25 Uhr	CodeIntegrity	3089 (1)	
Fehler	30.06.2019 18:42:25 Uhr	CodeIntegrity	3077 (18)	
Informationen	30.06.2019 18:42:24 Uhr	CodeIntegrity	3089 (1)	
Fehler	30.06.2019 18:42:24 Uhr	CodeIntegrity	3033 (1)	
Informationen	30.06.2019 18:26:23 Uhr	CodeIntegrity	3089 (1)	
Fehler	30.06.2019 18:26:23 Uhr	CodeIntegrity	3077 (18)	

The selected event (ID 3077) is expanded to show the following details:

Ereignis 3077, CodeIntegrity

Allgemein Details

Code Integrity determined that a process (\Device\HarddiskVolume4\Windows\System32\svchost.exe) attempted to load \Device\HarddiskVolume4\Users\Joern\Desktop\CCleaner_5.8.exe that did not meet the Enterprise signing level requirements or violated code integrity policy.

Protokollname: Microsoft-Windows-CodeIntegrity/Operational
Quelle: CodeIntegrity Protokolliert: 30.06.2019 18:42:25 Uhr
Ereignis-ID: 3077 Aufgabenkategorie: (18)
Ebene: Fehler Schlüsselwörter:
Benutzer: DELL7450\Joern Computer: DELL7450
Vorgangscod: (7274496)
Weitere Informationen: [Onlinehilfe](#)



Virtual Secure Mode und Device Guard

Update der Policy.xml

Soll auf der Maschine eine neue Software installiert werden, was ja aktuell nicht mehr funktionieren würde, so muss man Code Integrity wieder in den Audit Mode versetzen.

Set-RuleOption -FilePath C:\Policy\CI\policy.xml -Option 3

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\WINDOWS\system32> Set-RuleOption -FilePath C:\Policy\CI\policy.xml -Option 3
PS C:\WINDOWS\system32>
```

```
PS C:\Policy\ci> Set-RuleOption -help
0 Enabled:UMCI
1 Enabled:Boot Menu Protection
2 Required:WHQL
3 Enabled:Audit Mode
4 Disabled:Flight Signing
5 Enabled:Inherit Default Policy
6 Enabled:Unsigned System Integrity Policy
7 Allowed:Debug Policy Augmented
8 Required:EV Signers
9 Enabled:Advanced Boot Options Menu
10 Enabled:Boot Audit On Failure
11 Disabled:Script Enforcement
12 Required:Enforce Store Applications
13 Enabled:Managed Installer
14 Enabled:Intelligent Security Graph Authorization
15 Enabled:Invalidate EAs on Reboot
16 Enabled:Update Policy No Reboot
```

Als nächstes erstellen wir eine neue Policy namens „Install“ um mergen diese danach mit der bereits vorhanden Policy namens Policy.xml.

Zur Demonstration habe ich unter C:\ einen neuen Ordner namens Tools erstellt und 2 Portable Programme abgelegt, die gerade eben noch blockiert wurden. Diese möchte ich nun in meine bereits erstellte Policy mit aufnehmen, damit sie später benutzbar werden, also nicht mehr blockiert werden.

```
Administrator: Windows PowerShell
Wellknown      : False
Ekus           :
Exceptions     :
FileAttributes :
FileException  : False
UserMode      : True

Name           : C:\Program Files\Intel\Intel(R) Dynamic Platform and Thermal Framework\uninstall\sk-SK\setup.exe.dll Hash Page Sha256
Id            : ID_ALLOW_A_2CA6_0
TypeId        : Allow
Root          :
FileVersionRef :
AppIDRef      :
Wellknown     : False
Ekus         :
Exceptions    :
FileAttributes :
FileException : False
UserMode     : True

Name           : C:\Program Files\Intel\Intel(R) Dynamic Platform and Thermal Framework\uninstall\ru-RU\setup.exe.dll Hash Sha1
```



Virtual Secure Mode und Device Guard

New-CIPolicy -FilePath "C:\Install\Install.xml" -Level FilePublisher -Fallback Hash -UserPEs -ScanPath "C:\Tools\"

Der Parameter **-OmitPaths .\Install\NoScan** schließt zu scannende Ordner aus.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\WINDOWS\system32> Set-RuleOption -FilePath C:\Policy\CI\policy.xml -Option 3
PS C:\WINDOWS\system32> New-CIPolicy -FilePath Install.xml -Level FilePublisher -Fallback Hash -UserPEs -ScanPath "C:\Tools\"
"Überprüfung wurde erfolgreich abgeschlossen."
PS C:\WINDOWS\system32>
```

Den Merge Vorgang starten.

Merge-CIPolicy Policy.xml, Install.xml -OutputFilePath Policy2.xml

```
Administrator: Windows PowerShell
Wellknown      : False
Ekus           :
Exceptions     :
FileAttributes :
FileException  : False
UserMode      : True

Name           : C:\Program Files\Intel\Intel(R) Dynamic Platform and Thermal Framework\uninstall\sk-SK\setup.exe.dll Hash Page Sha256
Id             : ID_ALLOW_A_2CA6_0
TypeId        : Allow
Root          :
FileVersionRef :
AppIDRef      :
Wellknown     : False
Ekus          :
Exceptions    :
FileAttributes :
FileException  : False
UserMode     : True

Name           : C:\Program Files\Intel\Intel(R) Dynamic Platform and Thermal Framework\uninstall\ru-RU\setup.exe.dll Hash Sha1
```

Den Audit Mode wieder deaktivieren.

Set-RuleOption -FilePath "C:\Policy\CI\policy.xml" -Option 3 -Delete

Es ist auch klar zu erkennen, das die Policy nach dem mergen größer geworden ist.

The screenshot shows a File Explorer window displaying the contents of the 'C:\Policy\CI' directory. The files listed are:

Name	Änderungsdatum	Typ	Größe
Install.xml	30.06.2019 18:21	XML-Dokument	4 KB
policy.p7b	30.06.2019 17:54	PKCS #7-Zertifikate	1.249 KB
policy.xml	30.06.2019 18:37	XML-Dokument	3.546 KB
Policy2.xml	30.06.2019 18:27	XML-Dokument	3.595 KB

Below the File Explorer, a PowerShell terminal window shows the command:

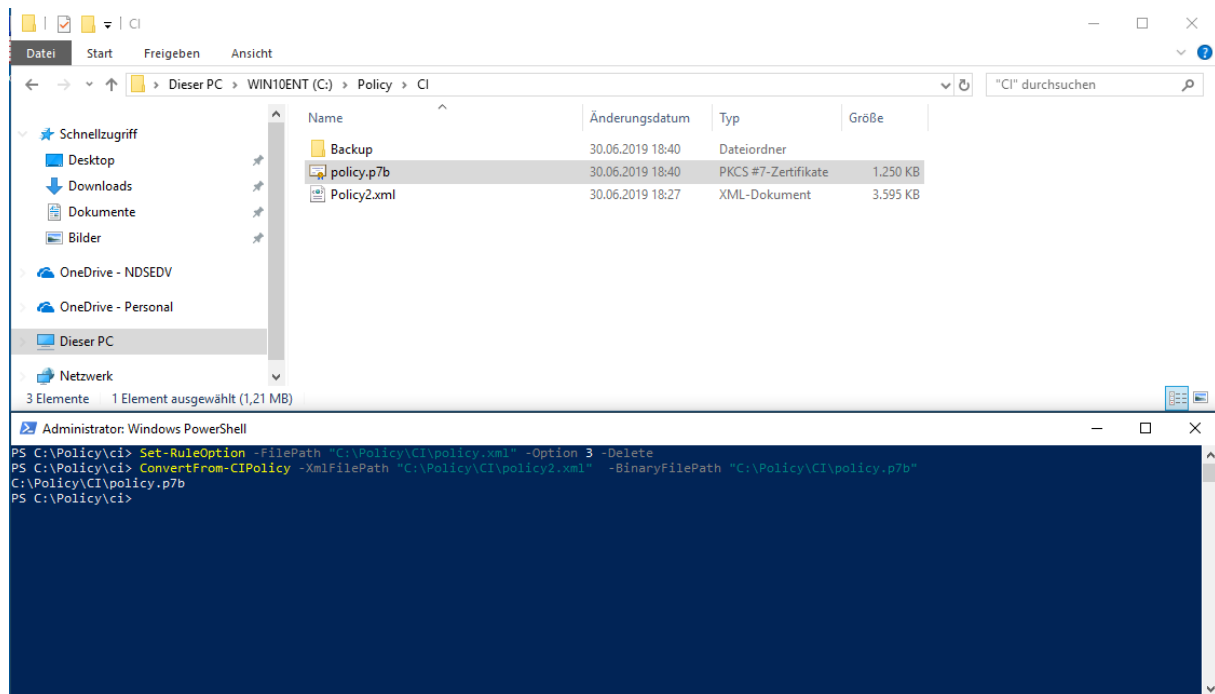
```
PS C:\Policy\ci> Set-RuleOption -FilePath "C:\Policy\CI\policy.xml" -Option 3 -Delete
PS C:\Policy\ci>
```



Virtual Secure Mode und Device Guard

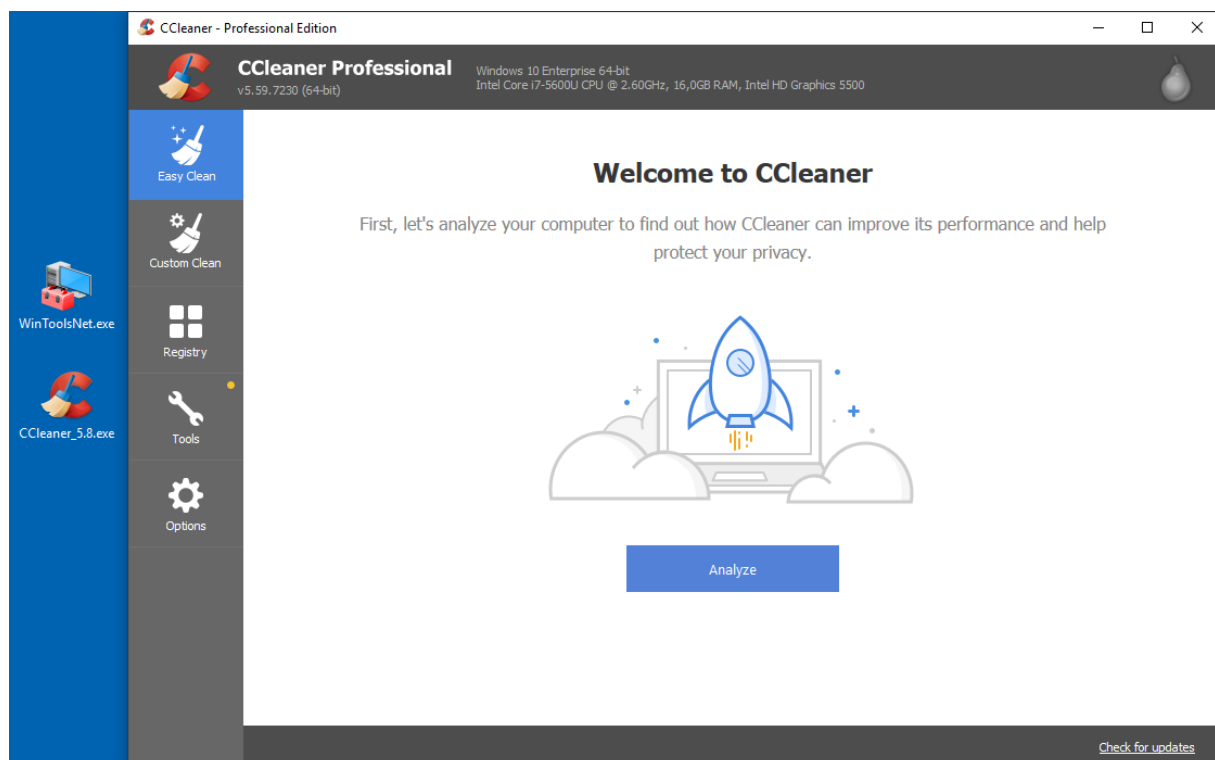
Abschließend wird die neue Policy2.xml wieder in eine binäre Datei umgewandelt.

```
ConvertFrom-CIPolicy -XmlFilePath "C:\Policy\CI\policy2.xml" -BinaryFilePath "C:\Policy\CI\policy.p7b"
```



Starte den Rechner einmal durch.

Voila, die beiden Apps lassen sich nun ausführen.





Virtual Secure Mode und Device Guard