Windows Engineering Guidance – Manufacturing for Device Guard

Microsoft Corporation

April 2015

Some information relates to pre-released product, which may be substantially modified before it's commercially released. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Copyright

Table of Contents

1 Revision History

Version 1.0 – February 2015

Description
Document Created

2 Contact Information

For Device Guard Questions, please reach out to your Microsoft Representative or send an email to dgext@microsoft.com

3 Overview

With Windows 10, Device Guard enables PCs to be protected against malware by introducing a collective set of restrictions on a device across several technologies. The core value proposition of Device Guard is the restriction of the Windows OS to supporting the execution of code signed by trusted signers defined in Code Integrity Policy. The security posture and robustness of this enforcement is complimented by the device hardware configuration as well as enablement of security features in Windows. These include the enablement and configuration of:

- User Mode Code Integrity
- New Kernel Code Integrity rules, which include new WHQL signing constraints
- Secure Boot with db/dbx restrictions
- Virtualization Based Security, including hypervisor based code integrity

The enablement of Device Guard has explicit requirements on both the hardware capabilities of a devices as well as the configuration of the device prior to enabling the features.

This whitepaper is covers the definition of a “Device Guard Capable”, “Device Guard Ready”, and “Device Guard Enabled” PC and the role that the factory floor and Custom Customer image processes have on the ability to address the needs of Enterprises and their users.

4 Introducing Device Guard

Device Guard is a new security brand in Windows 10 that defines a restrictive system configuration offering malware resiliency. This resiliency is achieved through a combination of hardware configuration and Windows configuration, and ultimately enforces code integrity from device power on through the runtime lifecycle of Windows.

Device Guard at its core represents a combination of User and Kernel Code Integrity enforcement along with configurations that aid in the protection of the code integrity policies and the agents of the system which enforce them. These include the use of hardware features of a device, such as the TPM and IOMMU as well as functional changes in Windows 10 such as the availability of Virtualization Based Security.

In conjunction with the Device Guard feature, customers may be interested in employing enhancements in Windows 10 as it relates to Anti-Malware, App Locker, and other related technologies. These enhancements will not be covered in this whitepaper.

As Device Guard is a collection of features, there is an effective sliding scale in the security posture reflected by the individual configurations of a system. The aim of providing Device Guard ready systems to customers is to minimize the labor required for direct on-device configuration of hardware capabilities, while also ensuring devices with features enabled can have a secure posture through their chain of custody leading to possession by an end user. At this time, Device Guard is a feature of the Windows 10 Enterprise Client SKU only.

5 Designing for Device Guard

Device Guard makes use of several modern PC assets. These include:

Technical Requirement	Motivation	Expected Configuration
Virtualization Extensions, such as Intel VT-x, AMD-V	The use of virtualization based security in Windows is the foundation of enhanced Credential Protection and Kernel robustness	Enabled by default
IOMMU, Such as Intel VT-d, AMD-Vi	Support for the IOMMU in Windows 10 enhances system resiliency against both local and network attacks	Enabled by default
X64 vs. x86	Device Guard requires the Virtualization Based Security features, which in turn require the Windows Hypervisor. As the Windows Hypervisor is only supported on x64 devices with x64 Windows, there is no support for Device Guard on x86 Windows installations.	X64 devices, and x64 OS installation
Trusted Platform Module (TPM)	The TPM 2.0 provides full integration with Attestation and can be used for additional key protections for the Virtualization Based Security offering. The availability of a TPM 2.0 is highly encouraged for Device Guard Capable systems.	TPM2.0 recommended
Boot Options	Ethernet, USB, CD, and other boot methods SHOULD be configured as disabled after a customer image is installed. This prevents other operating systems from booting.	Boot options restricted by default
Supervisor Password	The accessibility of device configuration through UEFI can permit a physically present user to bypass the configuration of the device	Supervisor Password set where possible as provided by customer
Secure Boot	Secure Boot ensures the safety and security of the preboot environment. A restrictive policy tailored to a customer's needs (an appropriately modified UEFI db, dbx) can further enhance security by removing threat vectors for undesirable code to execute	UEFI db restricted to explicit reduced set of roots

6 Device Guard Readiness States

Devices fall into 3 Device Guard readiness categories:

1) Device Guard Ready
2) Device Guard Capable
3) Not supported for Device Guard

Device Guard device security features are listed in section 5.

Device Guard Capable devices support these features, but the configuration state of the device is such that a physically present, privileged user must change them to the required state before the device is in the Device Guard Ready state. This imposes tremendous deployment friction for Enterprise IT, as each device must be individually modified.

Device Guard Ready devices are immediately capable of enabling all Device Guard features through central administration tasks, such as Group Policy or Device Management. Device Guard Ready machines that are imaged with a Customer Custom Desktop image can be provided to a customer with Device Guard fully enabled.

Devices that do not support the platform security features, such as virtualization, are not capable of supporting Device Guard.

7 Code Integrity Configuration for Device Guard

The core enablement of Device Guard is the provisioning of Code Integrity policy for both user and Kernel mode in Windows that restrict what code can execute on a device. Code Integrity Policy is a customer configured option and is deliverable through conventional management channels including Group Policy.

Code Integrity is a binary encoded XML document, and in Windows 10 can be a signed document. The signing of the document aids in protection against administrative users modifying or removing policy.

For the Device Guard feature, devices should only have code integrity configured if provided by a customer for a customer provided image.

8 Virtualization Based Security

The Windows 10 hypervisor introduces new capabilities for enabling virtual trust levels. A new component referred to as 'Isolated User Mode' leverages these capabilities to segregate the runtime environment of a device into two distinct trust levels. The memory and execution isolation provided by the Hypervisor allow a device to run security services isolated from the running root OS.

For Virtualization Based Security, Windows 10 provides a kernel code integrity service and credential isolation service. The outsourcing of Kernel Code Integrity to a hypervisor hosted service provides increased robustness for the root OS against any code that might be present at the kernel layer of the OS.

Device Guard Ready machines MUST have all virtualization capabilities for a device enabled by default. This includes Virtualization Extensions (such as Intel VT-x) and IOMMU support (such as Intel VT-d).

9 Driver Compliance with Device Guard

Beginning with the release of Windows 10, all new Windows 10 kernel mode drivers must be submitted to and digitally signed by the Windows Hardware Developer Center Dashboard. Windows 10 will not load new kernel mode drivers which are not signed by the portal.

Additionally, starting on October 15, 2015, the portal will only accept driver submissions, including both kernel and user mode driver submissions, that have a valid Extended Validation ("EV") Code Signing Certificate.

These changes help make Windows more secure. These changes limit the risk of a driver publisher's signing keys being lost or stolen and also ensures that driver publishers are strongly authenticated.

10 Customer Image Configuration for Device Guard Enablement and Deployment

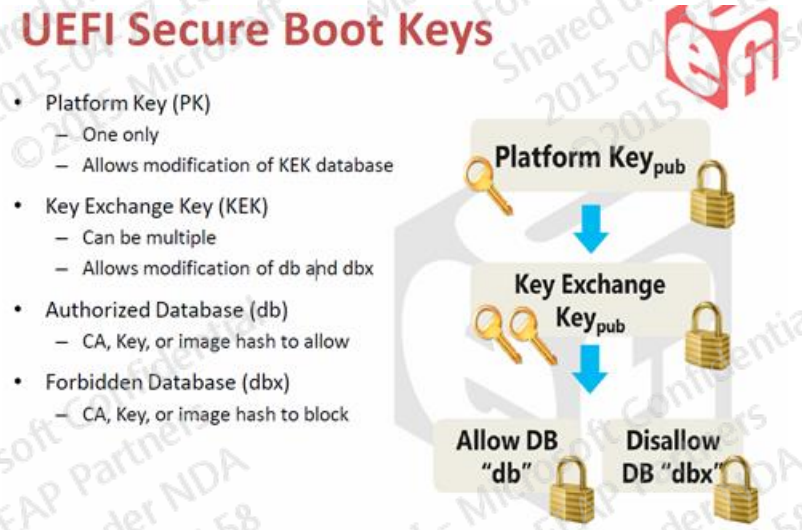
The enablement of Device Guard on a Device Guard Ready device is a core target for Customer Provided Image scenarios. This ensures the highest initial security posture of a device, and its readiness for immediate end user delivery in this configuration.

As part of the image building process, Customers will enable the Virtualization Based Security Feature, configure code integrity, and additional OS settings using enhanced deployment tools available in Windows 10. Additional documentation will follow for this process.

10.1 UEFI Secure Boot Configuration for Device Guard

Secure Boot ensures that all firmware images are authenticated against Authorized Database (db) and Forbidden Database (dbx) before allowing them to run. Windows 10 hardware requirements mandates the contents of db and dbx along with UEFI BIOS menu options. OEM can configure these to meet their Device Guard customer customization. Recommended configurations are provided in the below table.

Hierarchy of Secure Boot Keys – that are used for validating firmware images



Key / db name	Description	Owner
Platform Key (PK)	PK enables the root trust anchor in the Platform from which the trust chain is built for Secure Boot.	OEM/ Platform Owner
Key Exchange Key (KEK)	PK authorizes KEK Vendors to provide the second stage trust anchors.	Microsoft
Authorized Database (db)	Db is the list of all UEFI image files anchored to the KEK.	Microsoft + Allowed to be configured by OEM based on Device Guard customer customization
Forbidden Database (dbx)	A blacklist of keys, signatures, and/or hashes of binaries whose trust has been revoked.	Microsoft + Allowed to be configured by OEM based on Device Guard customer customization

Recommended configuration of UEFI Secure Boot options: This table is delta/in-addition to requirements specified in “Hardware Compatibility Specification for Systems for Windows 10”

Key / db name	Configuration	Description
Authorized Database (db)	**Required** - Microsoft Windows Production PCA 2011 **Recommended** – Do not include the Microsoft UEFI CA **If needed** – OEM certificate that are required to load OEM firmware drivers/apps **If needed** – Enterprise/Customer certificate that are required to load Enterprise customer firmware drivers/apps.	Need for removing Microsoft UEFI CA – all the 3rd party UEFI firmware apps/drivers/SHIMs (3rd party OS loaders) are signed with Microsoft UEFI CA. If Microsoft UEFI CA is present in the Db then an admin user on the device can programmatically install non-windows OS and other blocked firmware drivers/apps and thus compromise a Device Guard enabled device.
Database (dbx)	**Required** – Firmware that is revoked by OEM or Enterprise customer if any	
Secure Boot Menu	**Required** – Option to turn off Secure Boot must not be present or locked depending on Device Guard customer customization **Required** – Option to customize Secure Boot keys must not be present or locked depending on Device Guard customer customization	Ability to turn off Secure Boot and/or customize Secure Boot keys by a physically present user can be exploited to install non-Windows OS and other blocked firmware drivers/apps and thus compromise a Device Guard enabled device.

Note: Setting up a device with above recommended configuration wont trip any of the HLK tests. You must continue to pass the existing HLK tests to meet “Hardware Compatibility Specification for Systems for Windows 10”.

10.2 UEFI BIOS Menu Lockdown

All the hardware features mentioned below must be shipped ON by default and allowed to be modified only after platform administrator authentication:

1. Virtualization Extensions
2. IOMMU
3. TPM
4. Secure Boot
5. Boot Order Lock

Platform administrator authentication	Preference	Description
NO BIOS MENU to disable or configure these firmware and hardware features	Ideal	Having no BIOS Menu is ideal as it restricts a physically present attacker to compromise a Device Guard enabled device. To help Enterprise IT admins to debug and to reconfigure, UEFI tools to reconfigure these option can be provided that are signed by OEM or Enterprise certificate that is trusted by UEFI Secure Boot DB.
BIOS PASSWORD or any other authentication	Acceptable	Supervisor Password or any other authentication set where possible as provided by the customer. This is not ideal since it involves disclosure of password and password management, but is acceptable since it restricts a physically present attacker.

10.3 VBS Stack Deployment

VBS deployment requires installing of IUM (Isolated user Mode) and Hyper-V. And individual components of VBS stack can be configured for deployment through registry settings and/or through GP (Group Policy).

Deployment	Description
OEM Deployment	Step1: Using DISM APIs and/or through Unattend.xml customer image to be configured to install IUM and Hyper-V Step2: Set registry keys for 1. Platform Security Level HKLM\SYSTEM\CurrentControlSet\Control\DeviceGuard LsaCfgFlags (DWORD) 1 - to select “Secure Boot” only option 2 - to select “Secure Boot and DMA protection” option 2. Virtualization Based Protection of Code Integrity HKLM\SYSTEM\CurrentControlSet\Control\DeviceGuard HypervisorEnforcedCodeIntegrity (DWORD) 1 - to enable 0 – to disable 3. LSA Credential Isolation HKLM\SYSTEM\CurrentControlSet\Control\Lsa LsaCfgFlags (DWORD) 1 - to enable 0 – to disable
Enterprise Deployment	IT Admin can also provision VBS within the enterprise through GP (Group Policy): GP -> Administrative Templates\System\Device Guard\Turn On Virtualization Based Security

10.4 Config CI Deployment

Config CI deployment requires the presence of the Config CI Policy (.p7b) file in a specific OS location as mentioned below:

<EFI System Partition>\Microsoft\Boot\

Config CI Policy file can be either signed or unsigned depending on customer customization. Signed policy provides the added benefit that policy file cannot be modified or tampered with, hence is recommended.

Deployment	Description
OEM Deployment	Config CI Policy that is provided by customer can be baked into the device by placing the file in the location as mentioned above. <EFI System Partition>\Microsoft\Boot\
Enterprise Deployment	IT Admin can also provision Config CI Policy within the enterprise either through GP (Group Policy) or through MDM (Mobile Device Management). GP -> Administrative Templates\System\Device Guard\Deploy Code Integrity policy MDM CSP: AppLocker/ApplicationLaunchRestrictions/Grouping/CodeIntegrity