IMPLEMENTING MICROSOFT DEVICE GUARD FOR ISO 27001, PCI, AND FEDRAMP

North America | Latin America | Europe
877.224.8077 | info@coalfire.com | coalfire.com

INTRODUCTION

The threat of a cyber-attack is a constant factor all organizations must consider when developing their information security posture. Malware attacks are created on a daily basis and can inflict damage on a company's reputation and finances. Device Guard is a new security feature for Windows Server 2016 and the Windows 10 operating system, providing advanced protection against malware and improving upon hardware and software system integrity hardening by allowing only trusted applications to run in an organizational environment. Device Guard helps organizations to better protect their operating system from being attacked by malicious outsiders and strengthen their security posture.

In order to help customers implement this new capability for compliance with ISO, PCI, or FedRAMP, Microsoft worked closely with Coalfire, a recognized third-party IT compliance firm, to define each security and compliance objective in relation to the capabilities of Device Guard, and provide an example use case for each compliance and security objective. In addition, Appendix A contains mappings between Device Guard and the security control requirements present in ISO 27001, PCI DSS, and FedRAMP.

OVERVIEW OF DEVICE GUARD

The most common approach to malicious code protection is to deploy anti-virus or endpoint protection mechanisms to detect 'blacklisted' malware or malicious code after it has been introduced into the organization. This practice is certainly effective, but the time frame in between introduction and remediation can cause significant damage to the system and in turn to the organization. Microsoft now offers customers a new security mechanism in the form of Device Guard. In lieu of the aforementioned 'blacklisting' methodology, Device Guard utilizes a "whitelisting" process that only allows trusted binaries to run in the organizational environment. This method works as a prevention tool as well as a detection tool, providing customers a proactive approach to help prevent malware or malicious code from penetrating the operating system. Device Guard can be used as a standalone feature or deployed in combination with other Windows Server 2016 and Windows 10 features to provide additional levels of security.

DEVICE GUARD CODE INTEGRITY AND VIRTUALIZATION-BASED SECURITY

The Device Guard feature is based on powerful Code Integrity functionality within the Windows Server 2016 and Windows 10 kernel. This Code Integrity functionality now provides customers the ability to configure user-mode and kernel mode code integrity policies in a way that meets their organizational objectives, organizations can now choose what trusted publishers or programs they will allow in their system environment, including existing unsigned and signed Win32 applications. Code Integrity provides customers the ability to trust specific types of program binaries, whether by publisher (i.e., all signed Microsoft binaries) or at a more granular level to prohibit the execution of any untrusted binary associated with an unnecessary system function. In order to provide customers some flexibility, the Code Integrity policy can also be used to limit the execution of outdated versions of a particular trusted binary (i.e. a critical system component with known historical vulnerabilities), but permit the execution of updated versions of that trusted binary, to ensure system administrators do not have to manually reconfigure the Code Integrity policy every time a patch or update is released for a specific trusted binary.

In addition to leveraging new Code Integrity functionality, Device Guard can also be used in conjunction with new virtualization-based security (VBS) capabilities within Windows Server 2016 and Windows 10 to implement strong protections against malware. When VBS is enabled, it strengthens either the default kernel-mode code integrity policy, or the configurable code integrity policy that customers deploy. VBS capabilities can be used to isolate Windows services critical to the security and integrity of the system from user and kernel modes, including from host administrator action, essentially acting as a barricade to most malware execution paths.

VBS is used by the Hyper-V hypervisor to restrict sections of physical memory from Windows kernel access in order to provide a secure place to store code integrity policies, credentials, signatures, and other system-critical artifacts. This prevents attackers from compromising system-critical functions even if they compromise the Windows kernel. With VBS, even if malware gains access to the kernel, the effects can be severely limited, because the hypervisor can prevent the malware from executing code. Code Integrity checks for kernel components and device drivers are performed in a secure environment which is resistant to attack from kernel mode software, and page permissions for kernel mode are set and maintained by the hypervisor. Even if there are vulnerabilities that allow memory modification, like a buffer overflow, the modified memory cannot be executed.

DEVICE GUARD IMPLEMENTATIONS

The Device Guard feature leverages these new Code Integrity and VBS capabilities to provide customers the ability to implement several different levels of restrictions:

> **Audit Mode**: Code Integrity functionality runs in a passive audit mode. Code Integrity policies present on the machine will not be enforced to restrict process execution based on whether or not a binary is allowed but instead will log violations of the Code Integrity policy (i.e. execution of unapproved binaries) to the Windows Event Log. This provides system administrators or system monitors the ability to identify untrusted processes running on the machine.
>
> **Enforcement Mode**: Code Integrity functionality runs in active enforcement mode, ensuring that code that is not approved in accordance with the organization’s Code Integrity policy is not permitted to execute, ensuring the execution of trusted binaries only.
>
> **Enforcement Mode with hardware enforcement and VBS:** Code Integrity runs in an active enforcement mode and leverages CPU virtualization extensions to run Code Integrity services alongside the kernel in a Windows hypervisor-protected container. This ensures that kernel mode code that is not approved in accordance with the organization’s Code Integrity policy is not permitted to execute, even if an attacker compromises the Windows kernel or host administrator privileges. Hardware enforcement also provides Code Integrity assurance during startup, restart and other system state changes, providing a host the ability to verify during boot that the Code Integrity policy present on the machine is trusted and unchanged from before the system state change, and has not been modified by any offline or any lower-level attack vectors.

USING DEVICE GUARD WITH CREDENTIAL GUARD

Device Guard can be deployed in combination with Credential Guard to provide additional levels of security and assurance for an organization environment.

Credential Guard is another new feature of Windows Server 2016 and Windows 10 that leverages VBS in a similar way to Device Guard. Credential Guard provides additional protection to Active Directory domain users by storing credential artifacts within the same type of VBS virtualization container that hosts code integrity. By isolating these credential artifacts from the active user mode and kernel mode, they have a much lower risk of being stolen in the event a malicious attacker compromises the Windows machine and tries to harvest the credential artifacts to be used for further attacks.

USING DEVICE GUARD FOR COMPLIANCE WITH PCI, ISO 27001, AND FEDRAMP

In addition to providing customers a more powerful and effective way of administering malicious code protections, Device Guard can also help customers meet compliance with several common compliance frameworks. The remainder of this document is aimed at providing Device Guard control and requirement applicability across three common compliance frameworks: ISO 27001, PCI DSS, and FedRAMP.

Although compliance does not directly equate to security, many customers are required to adhere to different compliance standards as part of doing business in organization environments. This new solution is broadly applicable to numerous different controls within ISO 27001, PCI DSS, and FedRAMP, and provide customers an easier and more efficient way to meet applicable control requirements that are already in place.

Device Guard can help customers manage multiple technical assurance, process execution prevention, and malicious code protection requirements within ISO 27001, PCI DSS, and FedRAMP.

IDENTIFICATION OF AUTHORIZED AND UNAUTHORIZED PROCESSES AND [ATTEMPTED] PROCESS EXECUTION

Deploying Device Guard in any mode (Audit Mode, Enforcement Mode, or Enforcement Mode with VBS) provides monitoring capabilities for all executed binaries. Although Code Integrity policies present on the machine – whether those present by default or those configured by the customer – will not be enforced to restrict binary execution in Audit Mode, violations of the Code Integrity policy will still be logged to the Windows Event Log. This provides system administrators or system monitors the ability to identify untrusted binaries and processes running on the machine. Audit Mode functionality is perfect for a passive, low-risk environment, such as an organization workstation pool or shared publication servers where corporate IT functions are only monitoring for violations of company Acceptable Use Policies or productivity expectations.

ENFORCEMENT OF TECHNICAL PROCESS EXECUTION PREVENTION, MEMORY PROTECTION, AND STRONG CODE INTEGRITY ASSURANCE

Deploying Device Guard in Enforcement Mode or Enforcement Mode with VBS provides customers the ability to implement strong internal Code Integrity controls – preventing organizational users or malicious attackers from being able to execute applications, programs, or processes that have not been authorized. Code Integrity policies on the machine are actively enforced to restrict process execution of any sort that is not explicitly permitted by the organization. This provides system administrators the ability to effectively lock down the environment from any successful process execution-based intrusion attempt, whether internal or external. Enforcement Mode functionality is perfect for active, high-risk environments such as organization server environments where there is concern of advanced persistent threat or a high chance of machine compromise due to malicious code.

APPENDIX A: DEVICE GUARD MAPPING TO PCI, ISO 27001, AND FEDRAMP

Device Guard Security and Compliance Capability	ISO 27001: 2013	PCI DSS 3.2	FedRAMP; NIST 800-53 Revision 4
Identification of Authorized and Unauthorized Processes and [Attempted] Process Execution Note: Provided by Implementing Device Guard in any mode	A.12.2.1 – Controls Against Malware A.12.4.1 – Event Logging A.12.4.2 – Protection of Log Information A.12.4.3 – Administrator and Operator Logs	5.1 – Anti-Virus Software 5.1.1 – Anti-Virus Software – Detection and Prevention 5.2 – Anti-Virus Software – Updates and Monitoring 10.2.7 – Audit Object Actions 10.3 – Audit Trail 10.3.1 – Audit Trail – User Identification 10.3.2 – Audit Trail – Event Type 10.3.3 – Audit Trail – Date and Time 10.3.4 – Audit Trail – Success or Failure 10.3.5 – Audit Trail – Event Origination 10.3.6 – Audit Trail – Component, Resource, Data Identity 10.5 – Audit Trail Protection 10.5.2 – Audit Trail Protection – Prevent Unauthorized Modification	AU-2 – Audit Events AU-3 – Content of Audit Records AU-9 – Protection of Audit Information AU-12 – Audit Generation CM-7 (5) – Least Functionality | Authorized Software / Whitelisting SI-3 – Malicious Code Protection SI-3 (1) – Malicious Code Protection | Central Management
Enforcement of Technical Process Execution Prevention, Memory Protection, and Strong Code Integrity Assurance Note: Provided by Implementing Device	A.12.2.1 – Controls Against Malware A.12.5.1 – Installation of Software on Operational Systems A.14.1.2 – Securing Application Services on Public Networks	2.2 – System Configuration Settings and Hardening 2.2.2 – Enable Only Necessary Services, Protocols, Daemons 2.2.3 – Enable Additional Control for Insecure Services, Protocols, Daemons	AC-4 – Information Flow Enforcement AC-6 (10) – Least Privilege | Prohibit Non-Privileged Users from Executing Privileged Functions CM-2 – Baseline Configuration CM-5 (3) – Access Restrictions for Change | Signed Components CM-6 – Configuration Settings

Device Guard Security and Compliance Capability	ISO 27001: 2013	PCI DSS 3.2	FedRAMP; NIST 800-53 Revision 4
Guard in Enforcement Mode or Enforcement Mode with VBS	A.14.1.3 – Protection of Application Services Transactions	2.2.4 – Configure System to Prevent Misuse 2.2.5 – Remove Unnecessary System Functionality 5.1 – Anti-Virus Software 5.1.1 – Anti-Virus Software – Detection and Prevention 5.2 – Anti-Virus Software – Updates and Monitoring 5.3 – Anti-Virus Software – Prevent Disablement or Alteration 6.5.2 – Buffer Overflows 11.5 – Change Detection Mechanism	CM-7 – Least Functionality CM-7 (2) – Least Functionality | Prevent Program Execution CM-7 (5) – Least Functionality | Authorized Software / Whitelisting CM-8 (3) – Information System Component Inventory | Automated Unauthorized Component Detection SC-4 – Information in Shared Resources SC-5 – Denial of Service Protection SC-39 – Process Isolation SI-3 – Malicious Code Protection SI-3 (1) – Malicious Code Protection | Central Management SI-6 – Security Verification SI-7 – Software, Firmware, and Information Integrity SI-10 – Information Input Validation SI-16 – Memory Protection