Microsoft Security Features and Firmware Configurations

Spring 2018 UEFI Seminar and Plugfest
March 26-30, 2018
Presented by Scott Anderson and Jeremiah Cox
Facilitated by Michael Anderson
• Contains pre-release information shared under UEFI Forum NDA
• Please do not record or share this presentation outside of UEFI Forum
• VBS & HVCI everywhere
• UEFI CA and Baby Duck
• Device Firmware Configuration Interface (DFCI)
Virtualization Based Security & Hypervisor Protected Code Integrity (VBS & HVCI)
What are these technologies?

**Virtualization-based security** (VBS) is the ‘foundation’ for many features we have shipped since the start of Windows 10. VBS uses the hypervisor to create this virtual secure mode, and to enforce restrictions which protect vital system and operating system resources, or to protect security assets such as authenticated user credentials.

**Hypervisor-protected code integrity** (HVCI) has also been shipping since the start of Windows 10, but previously focused on Enterprise customers only and we are bringing it now across all SKU’s. It introduces requirements for *kernel* mode drivers only and HVCI compatibility has been a requirement for all new driver submissions since RS1.
Memory Integrity Protection

Leverages virtualization page tables managed by VTL1 to eliminate W^X memory in VTL0 kernel-mode.

**Normal Mode (VTL0)**
- User mode
- Kernel mode
- NT Kernel Pool
- Secure Kernel

**Secure Mode (VTL1)**
- Isolated Process C
- Isolated Process D
- C Secure Kernel Data
- D Secure Kernel Data
- Kernel Code Integrity Policy

**SLAT is used to gate enforce RX only**
HVIC running in SK validates code pages
If valid set GPA bits to
R=1 W=0 KMX=UMX=1

**Mode-Based Execute (MBE) Control**
Extended-Extended Page Tables (EPT)
- XU for user pages
- XS for supervisor pages
- KMX and UMX hardware bits.
Kernel Control Flow Integrity

Kernel CFG is used to enforce runtime code flow integrity for kernel drivers.

- **Kernel Control Flow Guard** improves protection against control flow hijacking for kernel code.
- Paired with HVCI to ensure both code integrity and control flow integrity.

**Compile time**

```c
void Foo(...) {
    // SomeFunc is address-taken
    // and may be called indirectly
    Object->FuncPtr = SomeFunc;
}
```

Metadata is automatically added to the image which identifies functions that may be called indirectly.

**Kernel Runtime**

```c
void Bar(...) {
    // Compiler-inserted check to verify call target is valid
    _guard_check_icall(Object->FuncPtr);
    Object->FuncPtr(xyz);
}
```

A lightweight check is inserted prior to indirect calls which will verify that the call target is valid at runtime.

- **Image Load**
  - Update valid call target data with metadata from Driver image

- **HVCI**
  - HVCI validates and maps pages
  - CFG bitmap is protected by HV

- **Indirect Call**
  - Perform O(1) validity check
  - Terminate process if invalid target

- Terminate process if invalid target
How can you as a partner validate?

Hardware:

a) Make sure the device you’re testing with meets the hardware and firmware configuration requirements for VBS and HVCI: [https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-vbs](https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-vbs)

b) Specifically on silicon type, this is from Intel’s 7th generation, Kabylake or later

c) Example PCs would be:
   - Surface Laptop, Book 2, Surface Pro 2017 and newer
   - Commercial Laptops: Dell XPS Ultrabooks, Dell Latitudes, HP Elitebooks 1000 series, Lenovo Carbon X1’s

1. Use the Device Guard and Credential Guard hardware readiness tool to enable

2. Once complete, to validate that VBS and HVCI are enabled, check by going to Start → Run → MSINFO32 and you will see the following come up:

<table>
<thead>
<tr>
<th>Virtualization-based security</th>
<th>Running</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtualization-based security Required Security Properties</td>
<td></td>
</tr>
<tr>
<td>Virtualization-based security Available Security Properties</td>
<td></td>
</tr>
<tr>
<td>Virtualization-based security Services Configured</td>
<td>Base Virtualization Support, Secure Boot, UEFI Code Readonly, SMM Security ...</td>
</tr>
<tr>
<td>Virtualization-based security Services Running</td>
<td>Hypervisor enforced Code Integrity</td>
</tr>
<tr>
<td>Device Encryption Support</td>
<td>Hypervisor enforced Code Integrity</td>
</tr>
<tr>
<td>A hypervisor has been detected. Features required for Hyper-V will not be displayed.</td>
<td>Elevation Required to View</td>
</tr>
</tbody>
</table>
Links for more details

HVCI Driver Compatibility:
https://blogs.msdn.microsoft.com/windows_hardware_certification/2015/05/22/driver-compatibility-with-device-guard-in-windows-10/

HVCI Compatibility Test:
https://docs.microsoft.com/en-us/windows.hardware/test/hlk/testref/b972fc52-2468-4462-9799-6a1898808c86

Device Guard and Credential Guard hardware readiness tool:
UEFI submission change to Dev Center
Windows Hardware Dev Center dashboard

The Windows Hardware Dev Center dashboard has replaced Sysdev for most hardware tasks and will completely replace it by the end of June 2018.

Hardware Dev Center dashboard

This site provides the following services:

- Create HCK and HLK device certification submissions.
- Create HCK, HLK, and attested signing submissions.
- Publish to Windows Update and create shipping labels with promotions.
- Share your driver with another company (Resell).
- Customize your driver after initial certification.
- Manage your users and legal agreements.

Sysdev

Please continue to use Sysdev for the following:

- WLK device certification submissions
- System submissions
- Device metadata
- Bug management
- Remote debugging
- Win32 app certification submissions
- Certified Products List

Sign in to the dashboard

Sign in to Sysdev
<table>
<thead>
<tr>
<th>Shared Product ID</th>
<th>Name</th>
<th>State</th>
<th>Certification type</th>
<th>Created date</th>
<th>Permission</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>11520150640793274</td>
<td>TestinDCAT_T1_T2_CL_USO_3_CL_USO_3_Driver2_v64.cab</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793273</td>
<td>TestinDCAT_T1_T2_CL_USO_1_OptionalNotElevated_Driver</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793272</td>
<td>TestinDCAT_T1_T2_CL_USO_1_OptionalNotElevated_Driver</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793271</td>
<td>TestinDCAT_T1_T2_CL_USO_3_CL_USO_3_MultipleDevices_x</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793270</td>
<td>TestinDCAT_T1_T2_CL_USO_3_CL_USO_3_MultipleDevices_A</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793269</td>
<td>TestinDCAT_T1_T2_CL_USO_2_CL_USO_2_REPUBLISH_v64.cab</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793268</td>
<td>TestinDCAT_T1_T2_CL_USO_2_REPUBLISH_ARM64.c</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793267</td>
<td>TestinDCAT_T1_T2_CL_USO_1_CL_USO_1_EXPIRED_v64.cab</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793266</td>
<td>TestinDCAT_T1_T2_CL_USO_1_CL_USO_1_EXPIRED_ARM64.c</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793265</td>
<td>TestinDCAT_T1_T2_CL_DU_17_DU_WITHOptional_Critical_I</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793264</td>
<td>TestinDCAT_T1_T2_CL_DU_17_DU_WITHOptional_Critical_I</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793263</td>
<td>TestinDCAT_T1_T2_CL_DU_17_DU_WITHOptional_Critical_D</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793234</td>
<td>TestinDCAT_T1_T2_CL_DU_17_DU_WITHOptional_Critical_D</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793232</td>
<td>TestinDCAT_T1_T2_CL_DU_17_DU_WITHOptional_Critical_D</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793231</td>
<td>TestinDCAT_T1_T2_CL_DU_17_DU_WITHOptional_Critical_D</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793230</td>
<td>TestinDCAT_T1_T2_CL_DU_17_DU_WITHOptional_Critical_D</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793229</td>
<td>TestinDCAT_T1_T2_CL_DU_16_CL_USO_16_InstalledDriver</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793228</td>
<td>TestinDCAT_T1_T2_CL_DU_16_CL_USO_16_InstalledDriver</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
<tr>
<td>11520150640793227</td>
<td>TestinDCAT_T1_T2_CL_DU_16_CL_USO_16_InstalledDriver</td>
<td>Completed</td>
<td>Attestation</td>
<td>12/13/2017</td>
<td>Author</td>
<td>SpiralOrbit Hardware Dev Center</td>
</tr>
</tbody>
</table>
## File Signing Services

<table>
<thead>
<tr>
<th>ID</th>
<th>Product Name</th>
<th>Type</th>
<th>Submitted</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1152921504607396159</td>
<td>UEFI Bootloader NXA 1.8</td>
<td>UEFI</td>
<td>12/1/2017</td>
<td>Preparation</td>
</tr>
<tr>
<td>1152921504607395848</td>
<td>LSA 1</td>
<td>LSA</td>
<td>11/3/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607395758</td>
<td>UEFI Bootloader UST 2.3</td>
<td>UEFI</td>
<td>5/1/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607393274</td>
<td>UEFI SpiralOrbit</td>
<td>UEFI</td>
<td>11/11/2017</td>
<td>Review</td>
</tr>
<tr>
<td>1152921504607393273</td>
<td>UEFI Bootloader ASO 3.1</td>
<td>UEFI</td>
<td>10/1/2017</td>
<td>Validation</td>
</tr>
<tr>
<td>1152921504607393272</td>
<td>LSA 2</td>
<td>LSA</td>
<td>6/8/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607393271</td>
<td>UEFI Bootloader ASG 1.2</td>
<td>UEFI</td>
<td>11/4/2017</td>
<td>Signing</td>
</tr>
<tr>
<td>1152921504607393270</td>
<td>UEFI Bootloader NXP 4.5</td>
<td>UEFI</td>
<td>12/2/2017</td>
<td>Waiting for Upload</td>
</tr>
<tr>
<td>11529215046073333270</td>
<td>UEFI Bootloader AST 5.6</td>
<td>UEFI</td>
<td>11/3/2017</td>
<td>Signing</td>
</tr>
</tbody>
</table>

Viewing page 1 of 61
Submit new UEFI Signing Request
<table>
<thead>
<tr>
<th>ID</th>
<th>Product Name</th>
<th>Type</th>
<th>Submitted</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1152921504607396159</td>
<td>UEFI Bootloader NXA 1.8</td>
<td>UEFI</td>
<td>12/1/2017</td>
<td>Preparation</td>
</tr>
<tr>
<td>1152921504607395848</td>
<td>LSA 1</td>
<td>LSA</td>
<td>11/3/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607395758</td>
<td>UEFI Bootloader UST 2.3</td>
<td>UEFI</td>
<td>5/1/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607393274</td>
<td>UEFI SpiralOrbit</td>
<td>UEFI</td>
<td>11/11/2017</td>
<td>Review</td>
</tr>
<tr>
<td>1152921504607393273</td>
<td>UEFI Bootloader ASO 3.1</td>
<td>UEFI</td>
<td>10/1/2017</td>
<td>Validation</td>
</tr>
<tr>
<td>1152921504607393272</td>
<td>LSA 2</td>
<td>LSA</td>
<td>6/8/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607393271</td>
<td>UEFI Bootloader ASG 1.2</td>
<td>UEFI</td>
<td>11/4/2017</td>
<td>Signing</td>
</tr>
<tr>
<td>1152921504607393270</td>
<td>UEFI Bootloader NXP 4.5</td>
<td>UEFI</td>
<td>12/2/2017</td>
<td>Waiting for Upload</td>
</tr>
<tr>
<td>1152921504607333270</td>
<td>UEFI Bootloader AST 5.6</td>
<td>UEFI</td>
<td>11/3/2017</td>
<td>Signing</td>
</tr>
</tbody>
</table>
New UEFI submission

Turnaround time:
The average turnaround time for UEFI submissions is one week. First time UEFI submissions and shims will take more time to complete, as they require deeper review and additional communication. To minimize turnaround times, we recommend avoiding duplicate submissions and aggregating UEFI packages.

UEFI submission requirements:
The submission has been tested according to the guidelines for Pre-submission testing for UEFI submissions. The submission adheres to the latest Microsoft UEFI CA Signing policy updates. By clicking Submit you agree that your submission adheres to these guidelines.

Name *

UEFI Name

Drag your packages here (.cab) or browse your files

Submit
New UEFI submission

Turnaround time:
The average turnaround time for UEFI submissions is one week. First time UEFI submissions and shims will take more time to complete, as they require deeper review and additional communication. To minimize turnaround times, we recommend avoiding duplicate submissions and aggregating UEFI packages.

UEFI submission requirements:
The submission has been tested according to the guidelines for Pre-submission testing for UEFI submissions.
The submission adheres to the latest Microsoft UEFI CA Signing policy updates.
By clicking Submit you agree that your submission adheres to these guidelines.

Name *
Spiral Orbit Boot Loader UEFI

SPOBootLoader.cab 15 of 5,234 bytes
Uploading
Pause

Submit
Spiral Orbit Boot Loader UEFI

Submission Details

- Submission ID: 1152921504607393274
- Created Date: 12/19/2017
- Company: Spiral Orbit
- Created By: John Doe

Download Signed Package

This link will be seen after signing and finalize are completed.
View Existing UEFI Submissions
<table>
<thead>
<tr>
<th>ID</th>
<th>Product Name</th>
<th>Type</th>
<th>Submitted</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1152921504607396159</td>
<td>UEFI Bootloader NXA 1.8</td>
<td>UEFI</td>
<td>12/1/2017</td>
<td>Preparation</td>
</tr>
<tr>
<td>1152921504607395848</td>
<td>LSA 1</td>
<td>LSA</td>
<td>11/3/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607395758</td>
<td>UEFI Bootloader UST 2.3</td>
<td>UEFI</td>
<td>5/1/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607393274</td>
<td>UEFI SpiralOrbit</td>
<td>UEFI</td>
<td>11/11/2017</td>
<td>Review</td>
</tr>
<tr>
<td>1152921504607393273</td>
<td>UEFI Bootloader ASO 3.1</td>
<td>UEFI</td>
<td>10/1/2017</td>
<td>Validation</td>
</tr>
<tr>
<td>1152921504607393272</td>
<td>LSA 2</td>
<td>LSA</td>
<td>6/8/2017</td>
<td>Completed</td>
</tr>
<tr>
<td>1152921504607393271</td>
<td>UEFI Bootloader ASG 1.2</td>
<td>UEFI</td>
<td>11/4/2017</td>
<td>Signing</td>
</tr>
<tr>
<td>1152921504607393270</td>
<td>UEFI Bootloader NXP 4.5</td>
<td>UEFI</td>
<td>12/2/2017</td>
<td>Waiting for Upload</td>
</tr>
<tr>
<td>1152921504607393270</td>
<td>UEFI Bootloader AST 5.6</td>
<td>UEFI</td>
<td>11/3/2017</td>
<td>Signing</td>
</tr>
</tbody>
</table>
UEFI Bootloader UST 2.3

**Submission Details**

- **Submission ID**: 1152921504607395758
- **Created Date**: 5/1/2017
- **Company**: SpiralOrbit
- **Created By**: John Doe

[Download Signed Package]
UEFI CA – ARM Signing
ARM Signing

• Microsoft is relaxing the policy to permit signing of AARCH64 EFI modules

• Some new, global prerequisites to follow...
Baby Duck – Secure Boot Settings
Signing Challenges

• Multi-signing
  – Overhead to create/manage IHV and ISV and OEM CAs
    • Long lived certs are a challenge
  – OEM/enterprise management of certs
  – Device and System compatibility
    • Increased size
Signing Solution Proposal

• Replace with .RSRC section + Opus Info
  – UEFI CA signer and process remains as-is
  – Opus added by Microsoft UEFI CA to signature
    • specifies Product Vendor from name on file
  – Action to UEFI Submitters: add .RSRC section (trivial)
    • specifies Product Name, File Name & Version
    • Provides improved revocation of insecure versions without flooding dbx with hashes or rolling signing keys
    • This will be a requirement for signing by the UEFI CA going forward
    • No compat issue with downlevel
  – Action for System Firmware:
    • New authentication of EFI images based upon Opus & .RSRC
    • Will be a requirement of future Windows OS
Deployment Challenges

• Unenlightened algorithm for reliable db configuration is impractical

• Per-device targeting
  – Only current option is per-device-unique PK/KEK, does not scale

• Resistance to consistent workflows
Deployment Alternatives

• Infeasible heuristic replaced with explicit profiles & custom mode
• Targeting: SMBIOS Vendor, Model, Serial #, & UUID
• End-user workflow minimized
Building it

Are all blocking issues mitigated?

What is the value prop?
• Replace the UEFI CA with stricter trust 🛠

Cost/benefit justified?
Device Firmware Configuration Interface (DFCI)

Contact SAUEFI@Microsoft.com
Discussion of non-public Windows plans
Questions?
Thanks for attending the Spring 2018 UEFI Plugfest

For more information on the UEFI Forum and UEFI Specifications, visit http://www.uefi.org

Questions regarding this presentation mailto:SAUEFI@Microsoft.com

presented by