OPEN SOURCE FIRMWARE CONFERENCE 2021 NOV. 30TH & DEC. 1ST

The Firmware Supply-Chain Security is broken!

Can we fix it?

SPEAKER

Alex Matrosov

Alex Ermolov

Kai Michaelis

Richard Hughes

BINARLY - Pasadena, CA

https://www.binarly.io

Protect Devices from emerging firmware and hardware threats using modern artificial intelligence.

Provide an advanced analytics platform for Security Operations Center and Incident Response teams for enhanced visibility into the enterprise device infrastructure.

founders@binarly.io

IMMUNE - Bochum, Germany

https://www.immu.ne/

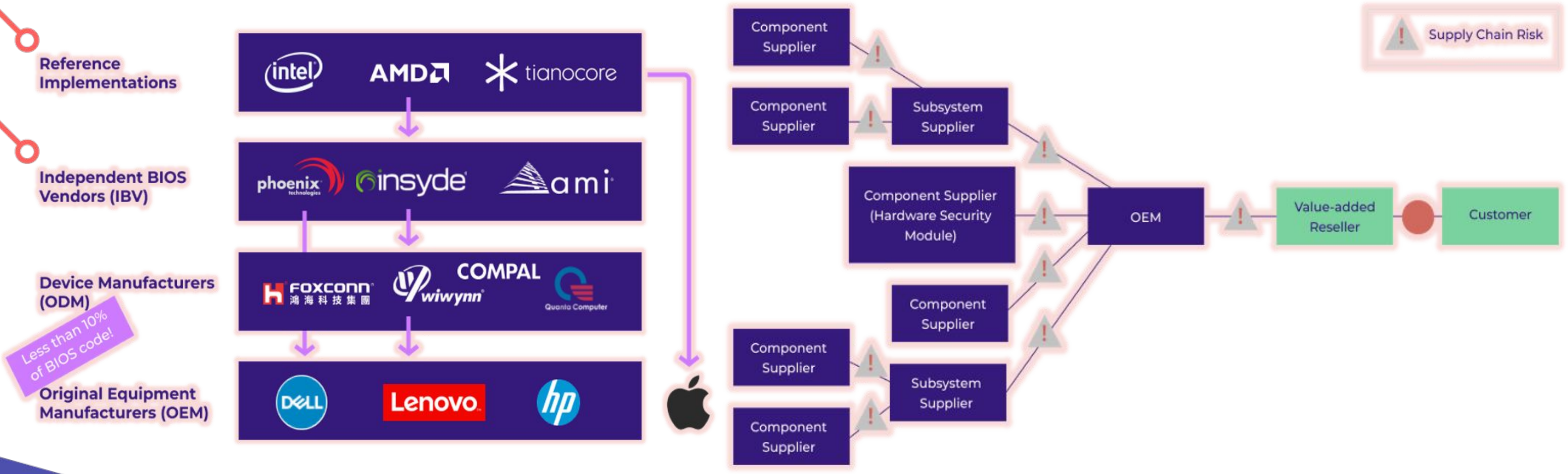
Provies **attestation as a service** for data center and edge compute.

In depth analysis of firmware, provisioned hardware configuration and boot chain.

Provides **firmware monitoring and risk assessment** across the whole server fleet.

contact@immu.ne

Supply Chain complexity keeps 1-days unfixed for years



Firmware Vulnerability Disclosure Time

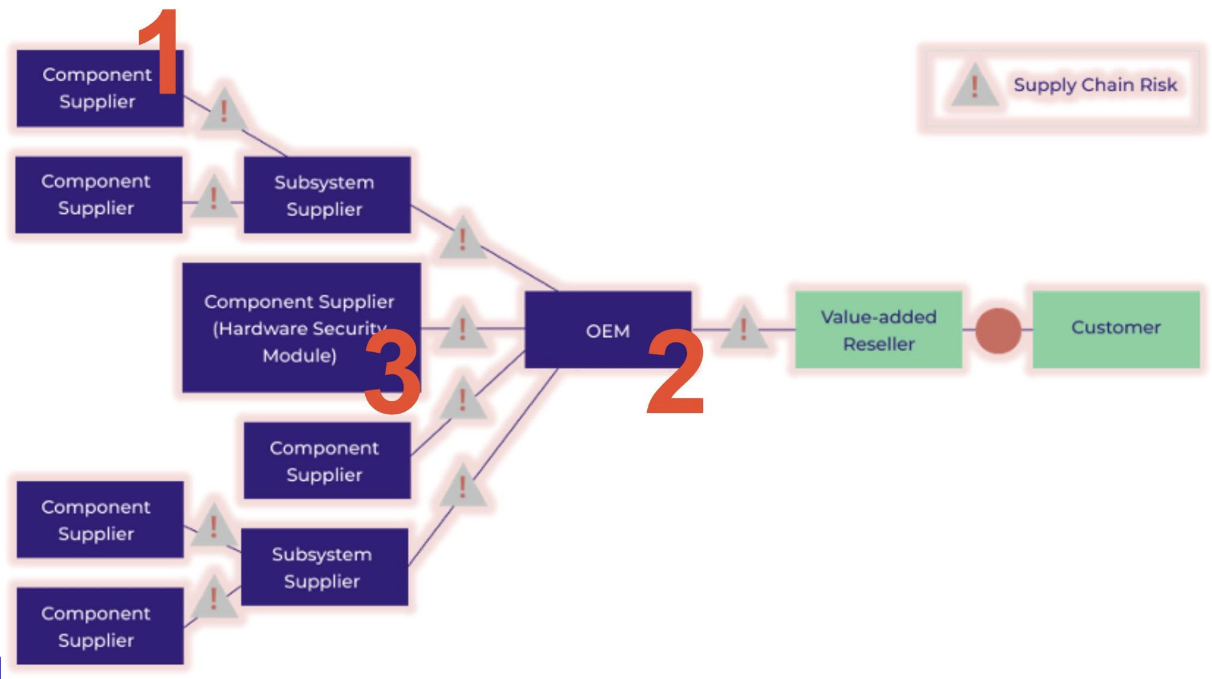
Combination of vulnerabilities creates a successful attack vector for compromising firmware:

1. **Privileges Escalation**
2. **FW storage write-protection bypass**
3. **HW-based trusted boot bypass**

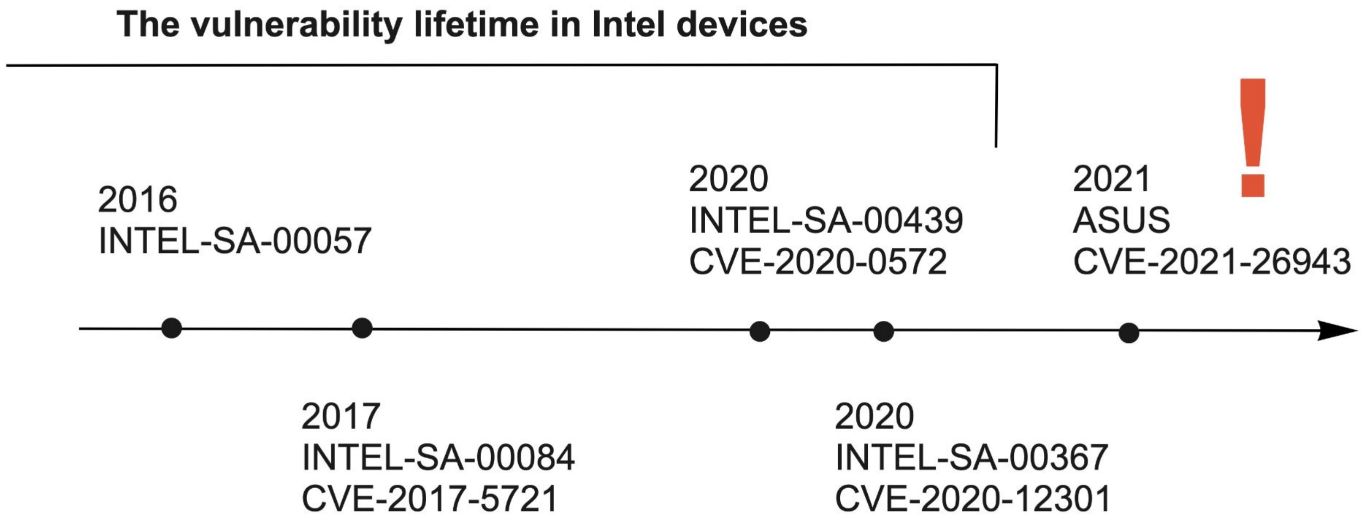
Patch delivery timeline could be different for each vulnerability in the attack vector.

A single vulnerability could be replaced by another one (not necessarily equal impact) to support the attack vector.

Supply Chain security failures points



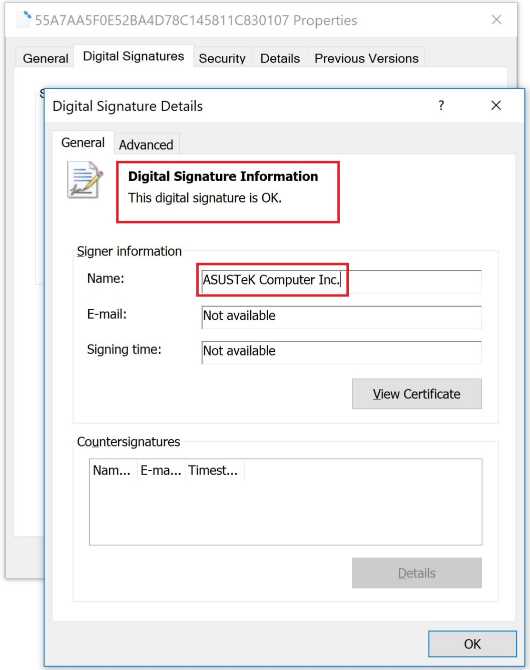
AMI UsbRt vulnerability is a perfect example of supply chain complexity



! Any device that uses the AMI UsbRt could be vulnerable to the very same bug

From reporting the firmware vulnerability to the fix usually takes around 6-9 months or more

Lack of transparency into supply chain



55A7AA5F0E52BA4D78C145811C830107 Properties

Digital Signature Details

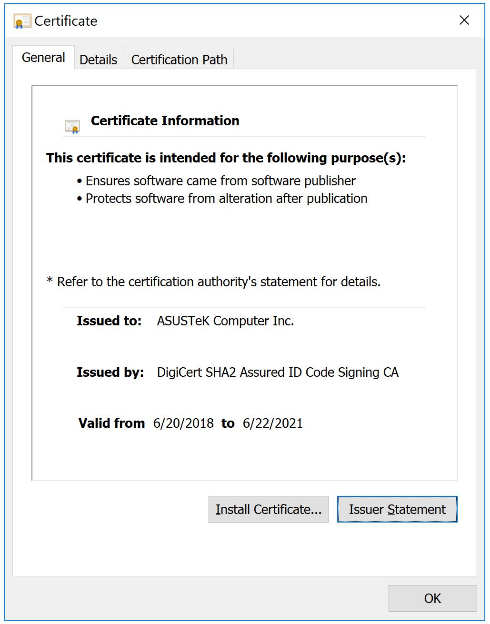
Digital Signature Information
This digital signature is OK.

Signer information

Name:	ASUSTeK Computer Inc.
E-mail:	Not available
Signing time:	Not available

Countersignatures

Nam...	E-ma...	Timest...



Certificate

Certificate Information

This certificate is intended for the following purpose(s):

- Ensures software came from software publisher
- Protects software from alteration after publication

* Refer to the certification authority's statement for details.

Issued to: ASUSTeK Computer Inc.

Issued by: DigiCert SHA2 Assured ID Code Signing CA

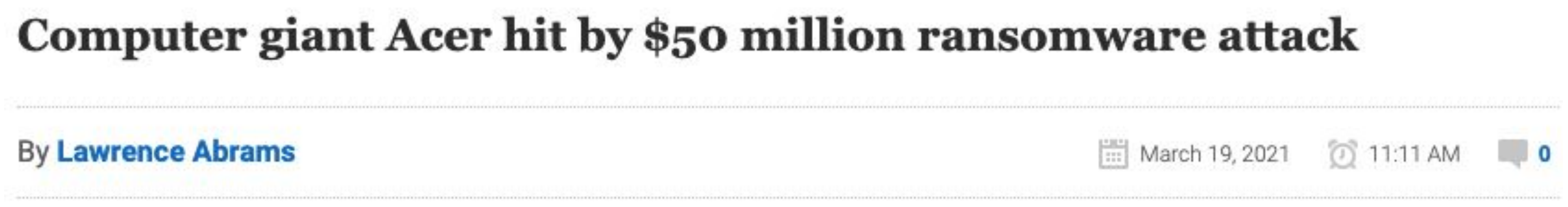
Valid from 6/20/2018 **to** 6/22/2021

An Investigative Update of the Cyberattack

By Sudhakar Ramakrishna

May 7, 2021 | Security SolarFocus

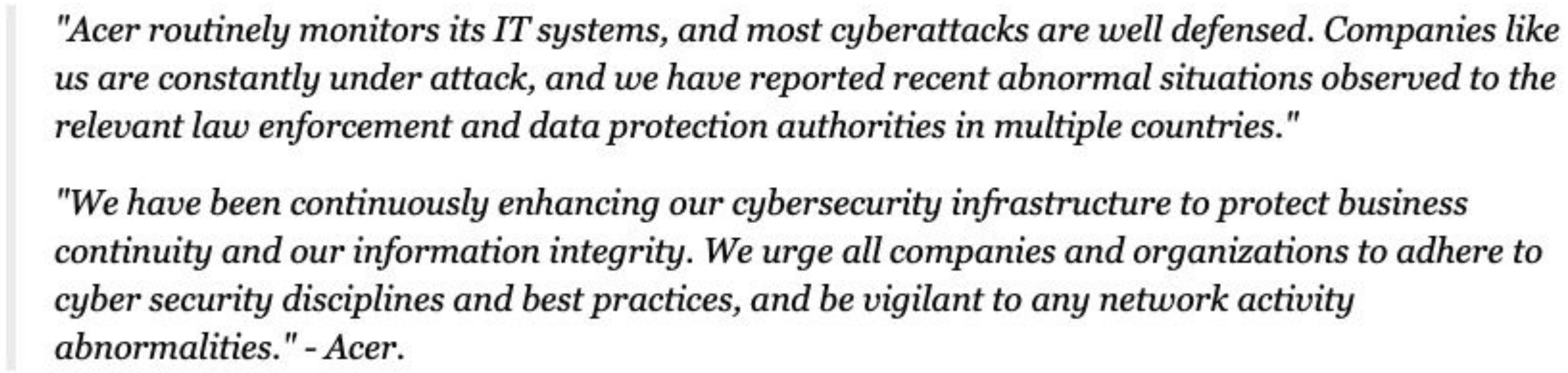
Lack of security in the supply chain



Computer giant Acer hit by $50 million ransomware attack

By Lawrence Abrams March 19, 2021 11:11 AM 0

Lack of security in the supply chain

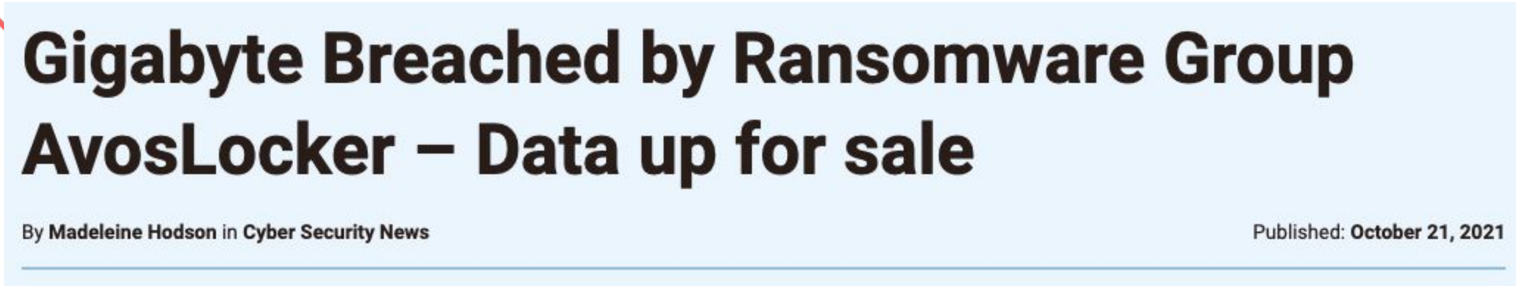


"Acer routinely monitors its IT systems, and most cyberattacks are well defensed. Companies like us are constantly under attack, and we have reported recent abnormal situations observed to the relevant law enforcement and data protection authorities in multiple countries."

"We have been continuously enhancing our cybersecurity infrastructure to protect business continuity and our information integrity. We urge all companies and organizations to adhere to cyber security disciplines and best practices, and be vigilant to any network activity abnormalities." - Acer.

OPEN SOURCE FIRMWARE CONFERENCE

Lack of security in the supply chain



Gigabyte Breached by Ransomware Group AvosLocker – Data up for sale

By Madeleine Hodson in Cyber Security News

Published: October 21, 2021

Lack of transparency into supply chain

- Complex, intransparent firmware supply chain
  - Lots of 3rd party suppliers
  - Lack of widespread use of software BOM
- Extremely long disclosure timelines
  - From months to a year
- Security vs. regular support
  - No SLA: no updates
  - Limited ability to notify end users
- Needs outside effort to fix the update problem
  - LVFS finally creates concise story for non-Windows platforms

What are vulnerability classifications

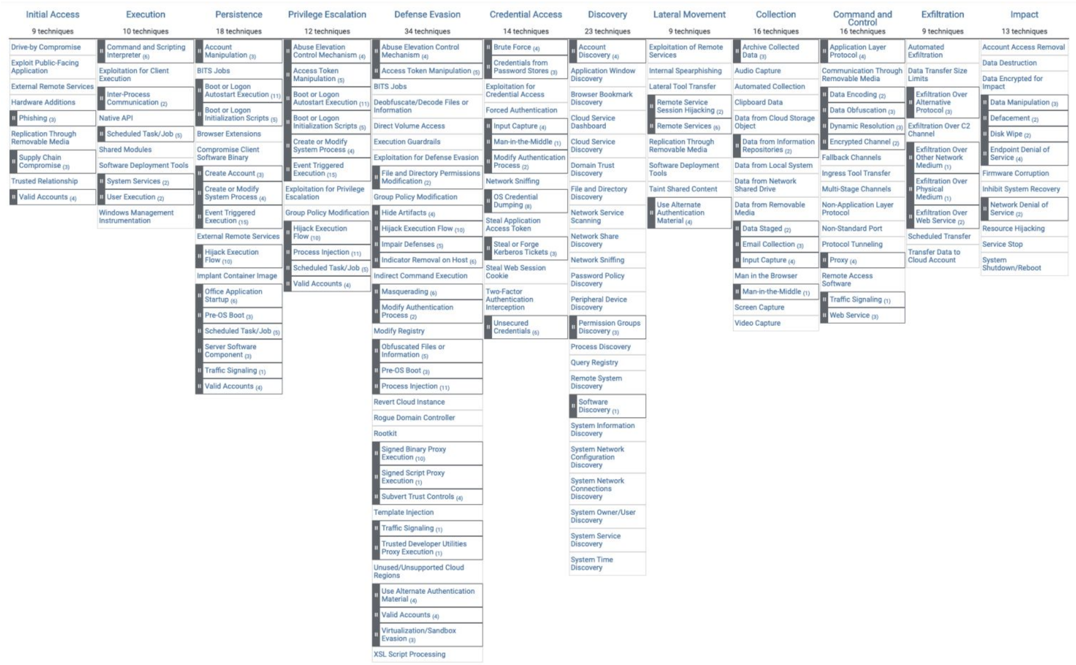
- Enumeration of (common) vulnerabilities
  - Common Weakness Enumeration (CWE)
  - MITRE ATT&CK Framework

- Shared vocabulary
  - Exchange IOC, techniques

- Knowledge base
  - Red Teaming
  - Defence assessment

- Systematization
  - Research

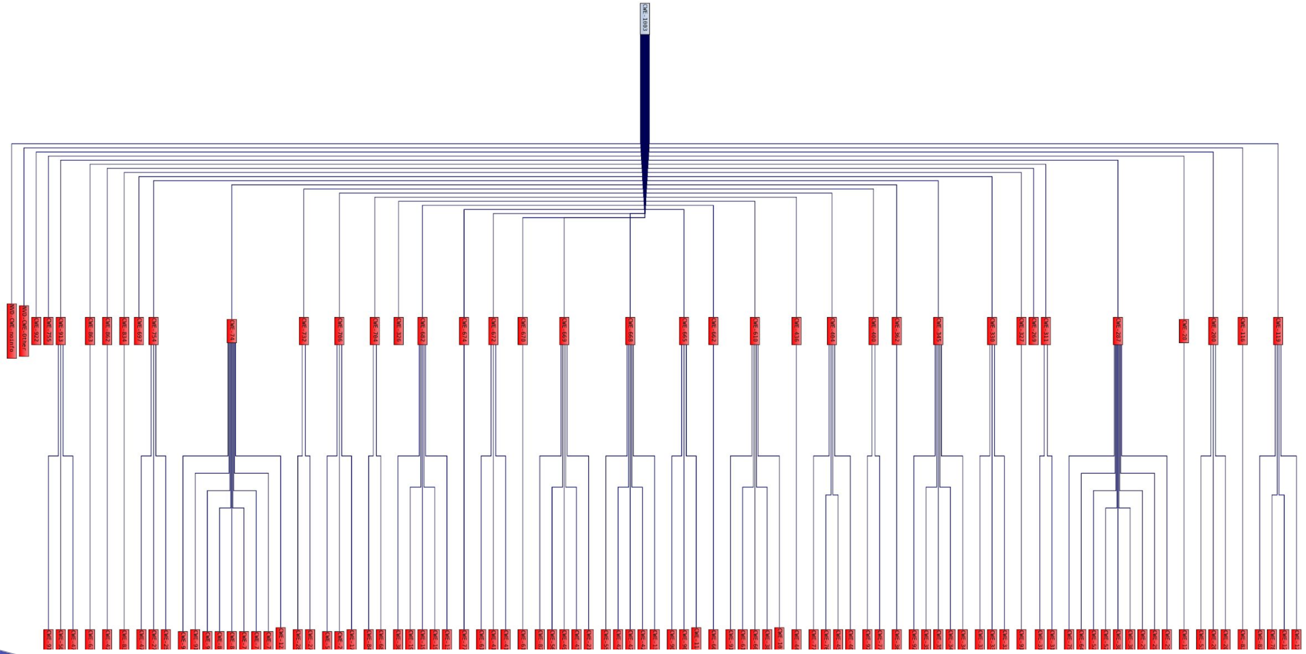
What are vulnerability classifications



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery
9 techniques	10 techniques	18 techniques	12 techniques	34 techniques	14 techniques	23 techniques
Drive-by Compromise	Command and Scripting Interpreter (6)	Account Manipulation (3)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)
Exploit Public-Facing Application	Exploitation for Client Execution	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Credentials from Password Stores (3)	Application Window Discovery
External Remote Services	Inter-Process Communication (2)	Boot or Logon Autostart Execution (11)	Boot or Logon Autostart Execution (11)	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery
Hardware Additions	Native API	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Service Dashboard
Phishing (3)	Scheduled Task/Job (5)	Browser Extensions	Create or Modify System Process (4)	Direct Volume Access	Input Capture (4)	Cloud Service Discovery
Replication Through Removable Media	Shared Modules	Compromise Client Software Binary	Event Triggered Execution (15)	Execution Guardrails	Man-in-the-Middle (1)	Domain Trust Discovery
Supply Chain Compromise (3)	Software Deployment Tools	Create Account (3)	Exploitation for Privilege Escalation	Exploitation for Defense Evasion	Modify Authentication Process (2)	File and Directory Discovery
Trusted Relationship	System Services (2)	Create or Modify System Process (4)	Group Policy Modification	File and Directory Permissions Modification (2)	Network Sniffing	Network Service Scanning
Valid Accounts (4)	User Execution (2)	Event Triggered Execution (15)	Hijack Execution Flow (10)	Group Policy Modification	OS Credential Dumping (8)	Network Share Discovery
	Windows Management Instrumentation	External Remote Services	Process Injection (11)	Hide Artifacts (4)	Steal Application Access Token	Network Sniffing
		Hijack Execution Flow (10)	Scheduled Task/Job (5)	Hijack Execution Flow (10)	Steal or Forge Kerberos Tickets (3)	Password Policy Discovery
		Implant Container Image	Valid Accounts (4)	Impair Defenses (5)	Steal Web Session Cookie	Peripheral Device Discovery
		Office Application Startup (6)		Indicator Removal on Host (6)	Two-Factor Authentication Interception	Permission Groups Discovery (3)
		Pre-OS Boot (3)		Indirect Command Execution	Unsecured Credentials (6)	Process Discovery
		Scheduled Task/Job (5)		Masquerading (6)		Query Registry
		Server Software Component (3)		Modify Authentication Process (2)		Remote System Discovery
		Traffic Signaling (1)		Modify Registry		Software Discovery (1)
		Valid Accounts (4)		Obfuscated Files or Information (5)		System Information Discovery
				Pre-OS Boot (3)		System Network Configuration Discovery
				Process Injection (11)		System Network Connections Discovery
				Revert Cloud Instance		System Owner/User Discovery
				Rogue Domain Controller		System Service Discovery
				Rootkit		System Time Discovery
				Signed Binary Proxy Execution (10)		
				Signed Script Proxy Execution (1)		
				Subvert Trust Controls (4)		
				Template Injection		
				Traffic Signaling (1)		
				Trusted Developer Utilities Proxy Execution (1)		
				Unused/Unsupported Cloud Regions		
				Use Alternate Authentication Material (4)		
				Valid Accounts (4)		
				Virtualization/Sandbox Evasion (3)		
				XSL Script Processing		

Lateral Movement	Collection	Command and Control	Exfiltration	Impact
9 techniques	16 techniques	16 techniques	9 techniques	13 techniques
Exploitation of Remote Services	Archive Collected Data (3)	Application Layer Protocol (4)	Automated Exfiltration	Account Access Removal
Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Lateral Tool Transfer	Automated Collection	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Remote Service Session Hijacking (2)	Clipboard Data	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Remote Services (6)	Data from Cloud Storage Object	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Replication Through Removable Media	Data from Information Repositories (2)	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Software Deployment Tools	Data from Local System	Fallback Channels	Exfiltration Over Web Service (2)	Endpoint Denial of Service (4)
Taint Shared Content	Data from Network Shared Drive	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Use Alternate Authentication Material (4)	Data from Removable Media	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
	Data Staged (2)	Non-Application Layer Protocol		Network Denial of Service (2)
	Email Collection (3)	Non-Standard Port		Resource Hijacking
	Input Capture (4)	Protocol Tunneling		Service Stop
	Man in the Browser	Proxy (4)		System Shutdown/Reboot
	Man-in-the-Middle (1)	Remote Access Software		
	Screen Capture	Traffic Signaling (1)		
	Video Capture	Web Service (3)		

What are vulnerability classifications

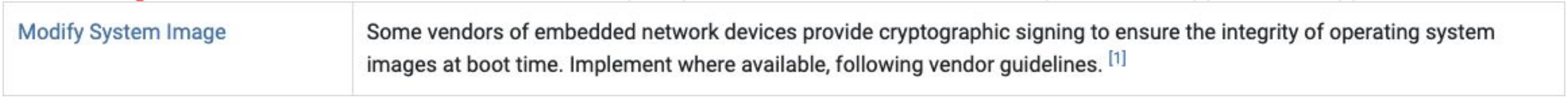


Why Another Spec?

- CWE
  - CWE-1236: Improper Neutralization of Formula Elements in a CSV File
  - CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
  - CWE-1310: Missing Ability to Patch ROM Code
- MITRE ATT&CK:
  - T1495: Firmware Corruption
  - T1542: Pre-OS Boot

Why Another Spec?

- CWE
  - CWE-1236: Improper Neutralization of Formula Elements in a CSV File
  - CWE-79: Improper Neutralization of Input During Web Page

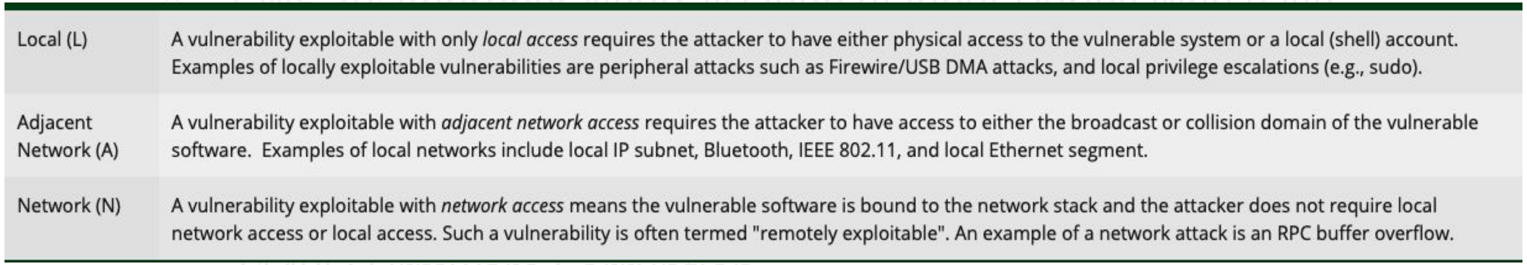


Modify System Image	Some vendors of embedded network devices provide cryptographic signing to ensure the integrity of operating system images at boot time. Implement where available, following vendor guidelines. [1]

- MITRE ATT&CK:
  - T1495: Firmware Corruption
  - T1542: Pre-OS Boot

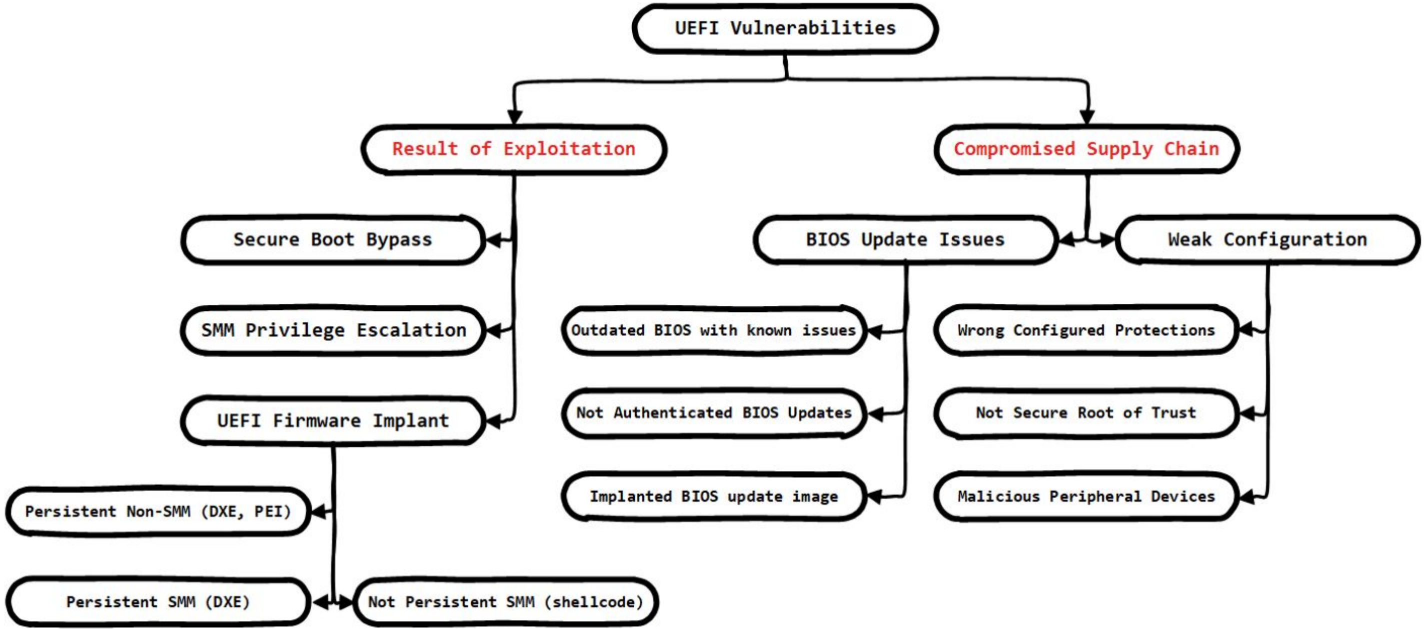
Why Another Spec?

- CWE
  - CWE-1236: Improper Neutralization of Formula Elements in a CSV File
  - CWE-79: Improper Neutralization of Input During Web Page
  - T1495: Firmware Corruption
  - T1542: Pre-OS Boot
-



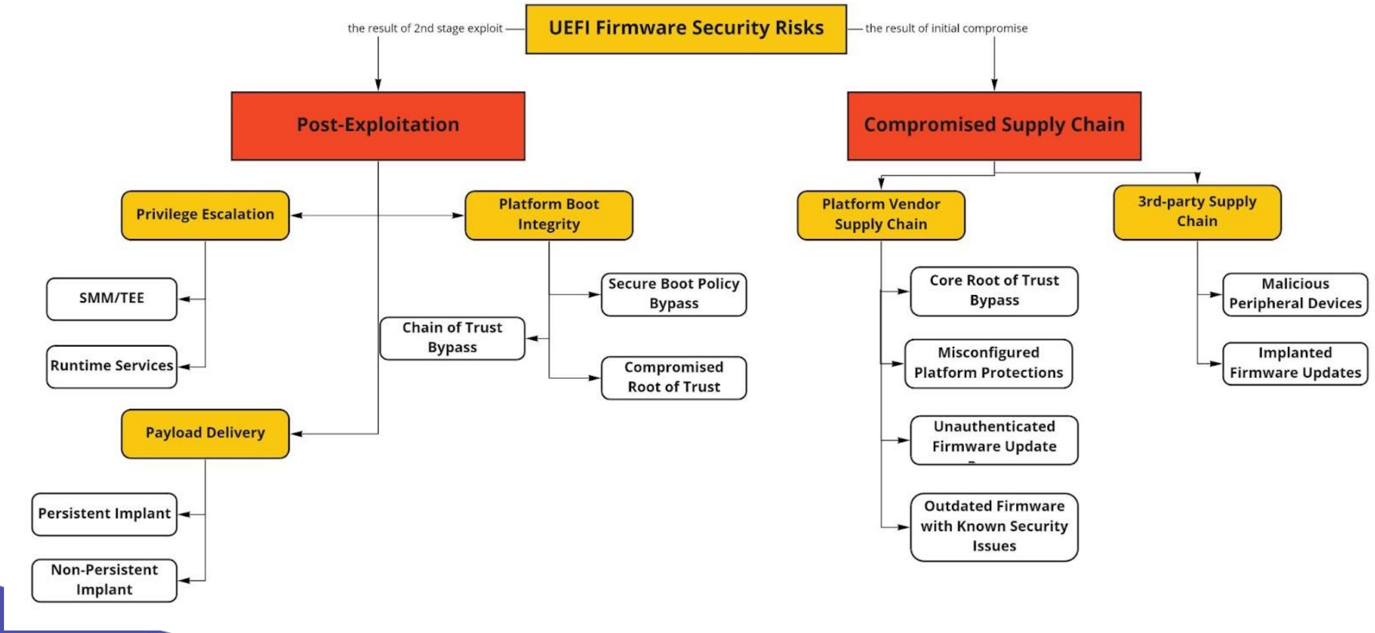
Local (L)	A vulnerability exploitable with only *local access* requires the attacker to have either physical access to the vulnerable system or a local (shell) account. Examples of locally exploitable vulnerabilities are peripheral attacks such as Firewire/USB DMA attacks, and local privilege escalations (e.g., sudo).
Adjacent Network (A)	A vulnerability exploitable with *adjacent network access* requires the attacker to have access to either the broadcast or collision domain of the vulnerable software. Examples of local networks include local IP subnet, Bluetooth, IEEE 802.11, and local Ethernet segment.
Network (N)	A vulnerability exploitable with *network access* means the vulnerable software is bound to the network stack and the attacker does not require local network access or local access. Such a vulnerability is often termed "remotely exploitable". An example of a network attack is an RPC buffer overflow.

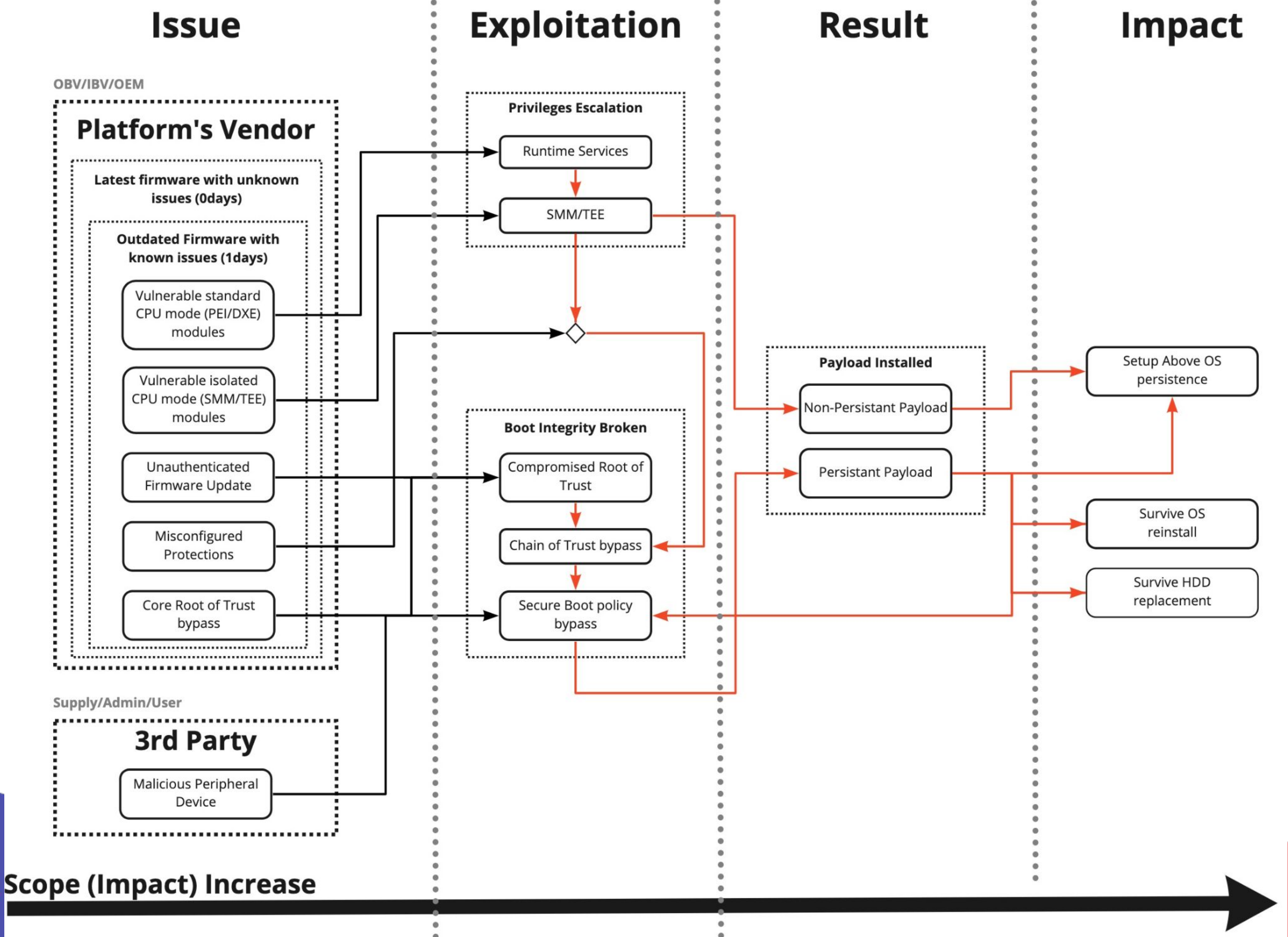
Firmware Vulnerabilities Classification: How It Started



https://medium.com/firmware-threat-hunting/uefi-vulnerabilities-classification-4897596e60af

Firmware Vulnerabilities Classification: Based on Impact


UEFI Firmware Security Risks
the result of 2nd stage exploit
the result of initial compromise
Post-Exploitation
Compromised Supply Chain
Privilege Escalation
Platform Boot Integrity
Platform Vendor Supply Chain
3rd-party Supply Chain
SMM/TEE
Runtime Services
Chain of Trust Bypass
Secure Boot Policy Bypass
Compromised Root of Trust
Payload Delivery
Persistent Implant
Non-Persistent Implant
Core Root of Trust Bypass
Misconfigured Platform Protections
Unauthenticated Firmware Update
Outdated Firmware with Known Security Issues
Malicious Peripheral Devices
Implanted Firmware Updates


Issue
Exploitation
Result
Impact
OBV/IBV/OEM
Platform's Vendor
Latest firmware with unknown issues (0days)
Outdated Firmware with known issues (1days)
Vulnerable standard CPU mode (PEI/DXE) modules
Vulnerable isolated CPU mode (SMM/TEE) modules
Unauthenticated Firmware Update
Misconfigured Protections
Core Root of Trust bypass
Supply/Admin/User
3rd Party
Malicious Peripheral Device
Privileges Escalation
Runtime Services
SMM/TEE
Boot Integrity Broken
Compromised Root of Trust
Chain of Trust bypass
Secure Boot policy bypass
Payload Installed
Non-Persistant Payload
Persistant Payload
Setup Above OS persistence
Survive OS reinstall
Survive HDD replacement
Scope (Impact) Increase

Multiply Affected Devices

Escalate Privileges

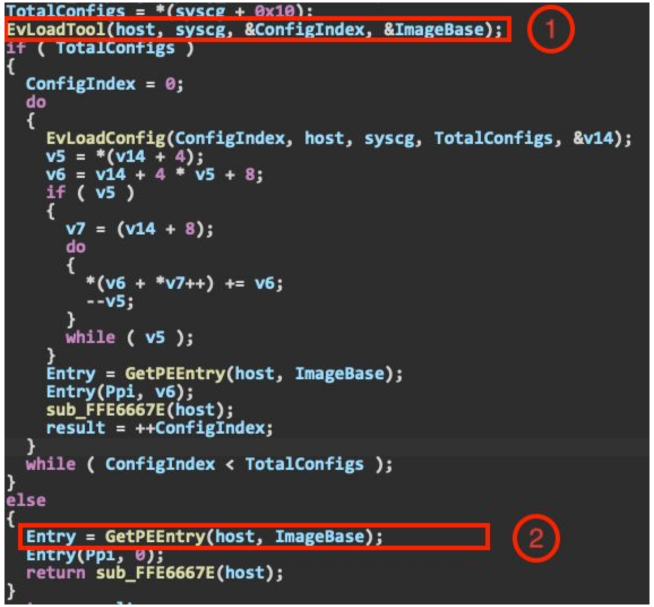
Payload workflow

◇AND
○OR

Disclosing and getting fixed the vulnerability is one side of the problem.

Another it's deliver this patch at scale to many systems in the field.

Intel BSSA DFT case study

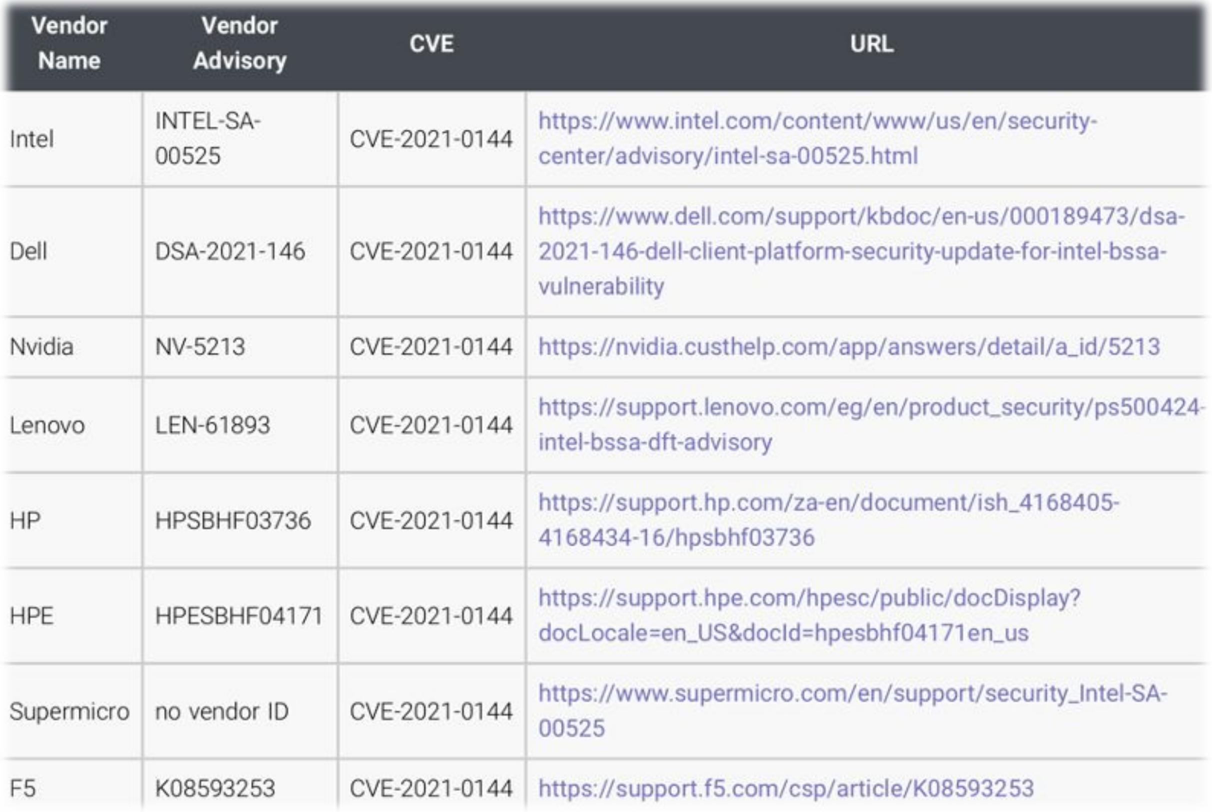


```
TotalConfigs = *(syscg + 0x10);
EvLoadTool(host, syscg, &ConfigIndex, &ImageBase);
if ( TotalConfigs )
{
  ConfigIndex = 0;
  do
  {
    EvLoadConfig(ConfigIndex, host, syscg, TotalConfigs, &v14);
    v5 = *(v14 + 4);
    v6 = v14 + 4 * v5 + 8;
    if ( v5 )
    {
      v7 = (v14 + 8);
      do
      {
        *(v6 + *v7++) += v6;
        --v5;
      }
      while ( v5 );
    }
    Entry = GetPEEntry(host, ImageBase);
    Entry(Ppi, v6);
    sub_FFE6667E(host);
    result = ++ConfigIndex;
  }
  while ( ConfigIndex < TotalConfigs );
}
else
{
  Entry = GetPEEntry(host, ImageBase);
  Entry(Ppi, 0);
  return sub_FFE6667E(host);
}
```

- These hidden features of Intel BSSA were designed to run arbitrary unsigned code blobs stored in EFI variables.
- To make matters worse, Intel BSSA DFT is a part of the reference code.
- Intel BSSA DFT was intended to be used for debugging or testing purposes only.

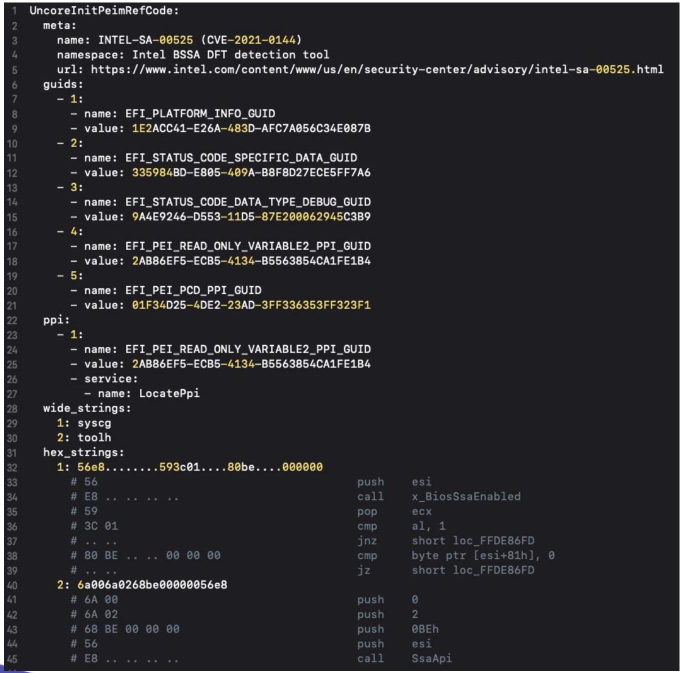
https://www.binarly.io/posts/Attacking_(pre)EFI_Ecosystem

OPEN SOURCE FIRMWARE CONFERENCE

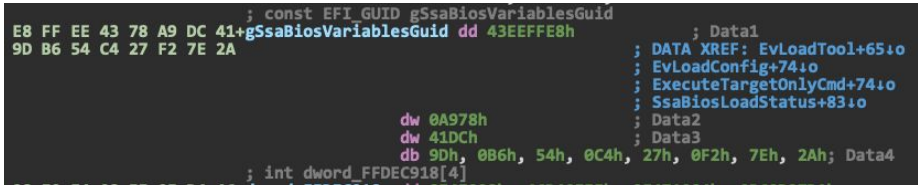


Vendor Name	Vendor Advisory	CVE	URL
Intel	INTEL-SA-00525	CVE-2021-0144	https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00525.html
Dell	DSA-2021-146	CVE-2021-0144	https://www.dell.com/support/kbdoc/en-us/000189473/dsa-2021-146-dell-client-platform-security-update-for-intel-bssa-vulnerability
Nvidia	NV-5213	CVE-2021-0144	https://nvidia.custhelp.com/app/answers/detail/a_id/5213
Lenovo	LEN-61893	CVE-2021-0144	https://support.lenovo.com/eg/en/product_security/ps500424-intel-bssa-dft-advisory
HP	HPSBHF03736	CVE-2021-0144	https://support.hp.com/za-en/document/ish_4168405-4168434-16/hpsbhf03736
HPE	HPESBHF04171	CVE-2021-0144	https://support.hpe.com/hpesc/public/docDisplay?docLocale=en_US&docId=hpesbhf04171en_us
Supermicro	no vendor ID	CVE-2021-0144	https://www.supermicro.com/en/support/security_Intel-SA-00525
F5	K08593253	CVE-2021-0144	https://support.f5.com/csp/article/K08593253

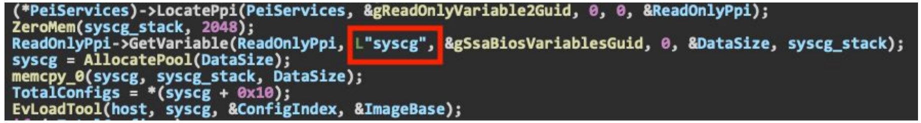
Binarly FwHunt vs Intel BSSA DFT



```
UncoreInitPeimRefCode:
  meta:
    name: INTEL-SA-00525 (CVE-2021-0144)
    namespace: Intel BSSA DFT detection tool
    url: https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00525.html
  guids:
    - 1:
      - name: EFI_PLATFORM_INFO_GUID
      - value: 1E2ACC41-E26A-483D-AFC7A056C34E087B
    - 2:
      - name: EFI_STATUS_CODE_SPECIFIC_DATA_GUID
      - value: 335984BD-E805-409A-B8F8D27ECE5FF7A6
    - 3:
      - name: EFI_STATUS_CODE_DATA_TYPE_DEBUG_GUID
      - value: 9A4E9246-D553-11D5-87E200062945C3B9
    - 4:
      - name: EFI_PEI_READ_ONLY_VARIABLE2_PPI_GUID
      - value: 2AB86EF5-ECB5-4134-B5563854CA1FE1B4
    - 5:
      - name: EFI_PEI_PCD_PPI_GUID
      - value: 01F34D25-4DE2-23AD-3FF336353FF323F1
  ppi:
    - 1:
      - name: EFI_PEI_READ_ONLY_VARIABLE2_PPI_GUID
      - value: 2AB86EF5-ECB5-4134-B5563854CA1FE1B4
      - service:
        - name: LocatePpi
  wide_strings:
    1: syscg
    2: toolh
  hex_strings:
    1: 56e8........593c01....80be....000000
      # 56                                        push    esi
      # E8 .. .. .. ..                            call    x_BiosSsaEnabled
      # 59                                        pop     ecx
      # 3C 01                                     cmp     al, 1
      # .. ..                                     jnz     short loc_FFDE86FD
      # 80 BE .. .. 00 00 00                      cmp     byte ptr [esi+81h], 0
      # .. ..                                     jz      short loc_FFDE86FD
    2: 6a006a0268be00000056e8
      # 6A 00                                     push    0
      # 6A 02                                     push    2
      # 68 BE 00 00 00                            push    0BEh
      # 56                                        push    esi
      # E8 .. .. .. ..                            call    SsaApi
```



```
                              ; const EFI_GUID gSsaBiosVariablesGuid
E8 FF EE 43 78 A9 DC 41+gSsaBiosVariablesGuid dd 43EEFFE8h           ; Data1
9D B6 54 C4 27 F2 7E 2A                                        ; DATA XREF: EvLoadTool+65↓o
                                                               ; EvLoadConfig+74↓o
                                                               ; ExecuteTargetOnlyCmd+74↓o
                                                               ; SsaBiosLoadStatus+83↓o
                                         dw 0A978h             ; Data2
                                         dw 41DCh              ; Data3
                                         db 9Dh, 0B6h, 54h, 0C4h, 27h, 0F2h, 7Eh, 2Ah; Data4
                              ; int dword_FFDEC918[4]
```



```
(*PeiServices)->LocatePpi(PeiServices, &gReadOnlyVariable2Guid, 0, 0, &ReadOnlyPpi);
ZeroMem(syscg_stack, 2048);
ReadOnlyPpi->GetVariable(ReadOnlyPpi, L"syscg", &gSsaBiosVariablesGuid, 0, &DataSize, syscg_stack);
syscg = AllocatePool(DataSize);
memcpy_0(syscg, syscg_stack, DataSize);
TotalConfigs = *(syscg + 0x10);
EvLoadTool(host, syscg, &ConfigIndex, &ImageBase);
```

YARA rules that are NOT tailored to effectively cover UEFI firmware code specifics

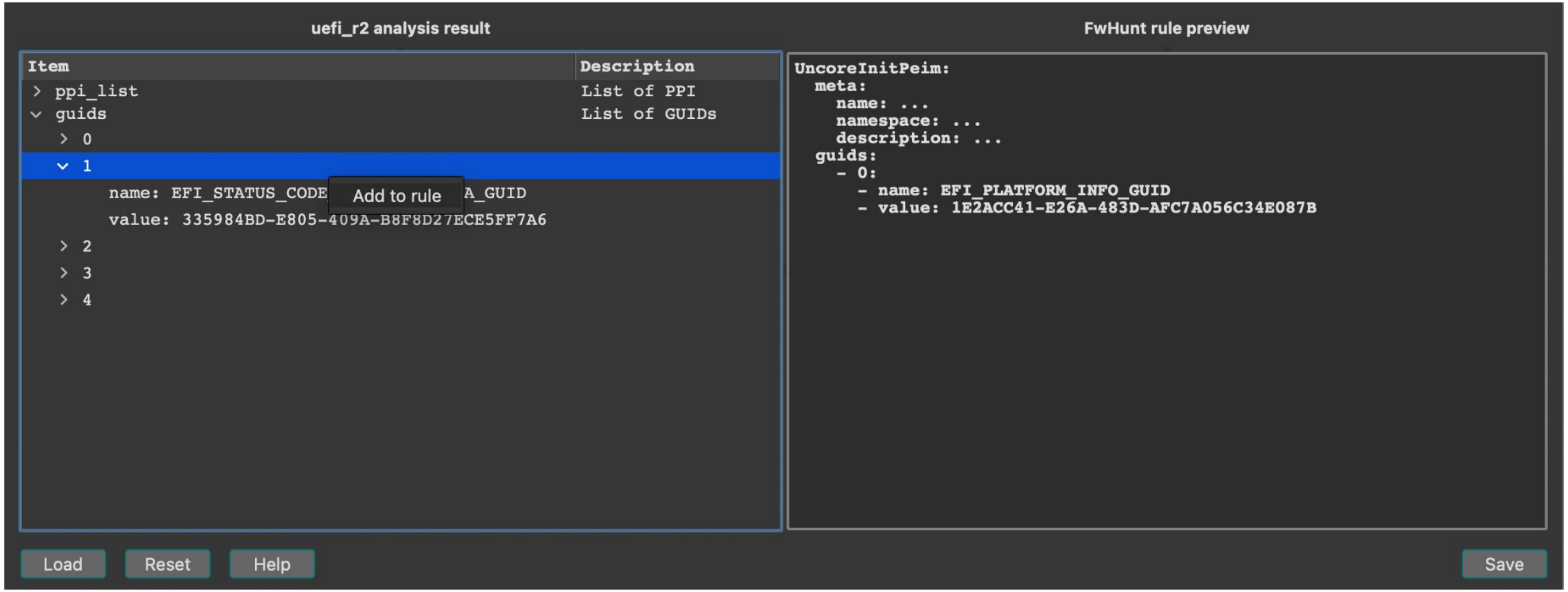
The Firmware Hunt (FwHunt) rule format was designed to scan for known vulnerabilities and verify that an affected OEM vendor has patched the issue in its latest update.

Binarly FwHunt vs Intel BSSA DFT

```
PS C:\tmp\uefi_r2> _
```

https://www.binarly.io/posts/Detecting_Firmware_vulnerabilities_at_scale_Intel_BSSA_DFT_case_study

UEFI Firmware IOC hunt

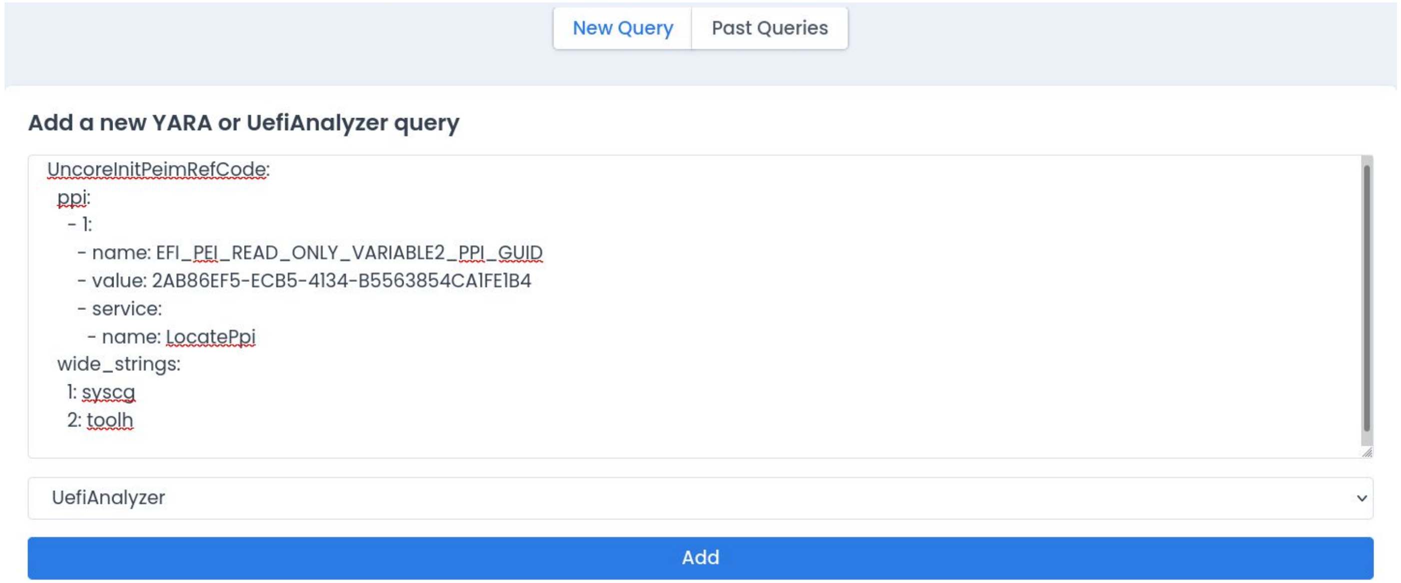


```
Item                                             Description
> ppi_list                                       List of PPI
v guids                                          List of GUIDs
  > 0
  v 1
      name: EFI_STATUS_CODE            A_GUID
      value: 335984BD-E805-409A-B8F8D27ECE5FF7A6
  > 2
  > 3
  > 4
```

```
UncoreInitPeim:
  meta:
    name: ...
    namespace: ...
    description: ...
  guids:
    - 0:
      - name: EFI_PLATFORM_INFO_GUID
      - value: 1E2ACC41-E26A-483D-AFC7A056C34E087B
```

https://github.com/binarly-io/fwhunt-ida
https://www.youtube.com/watch?v=V0-le7z_ojE

LVFS Yara and UEFI R2 scanning

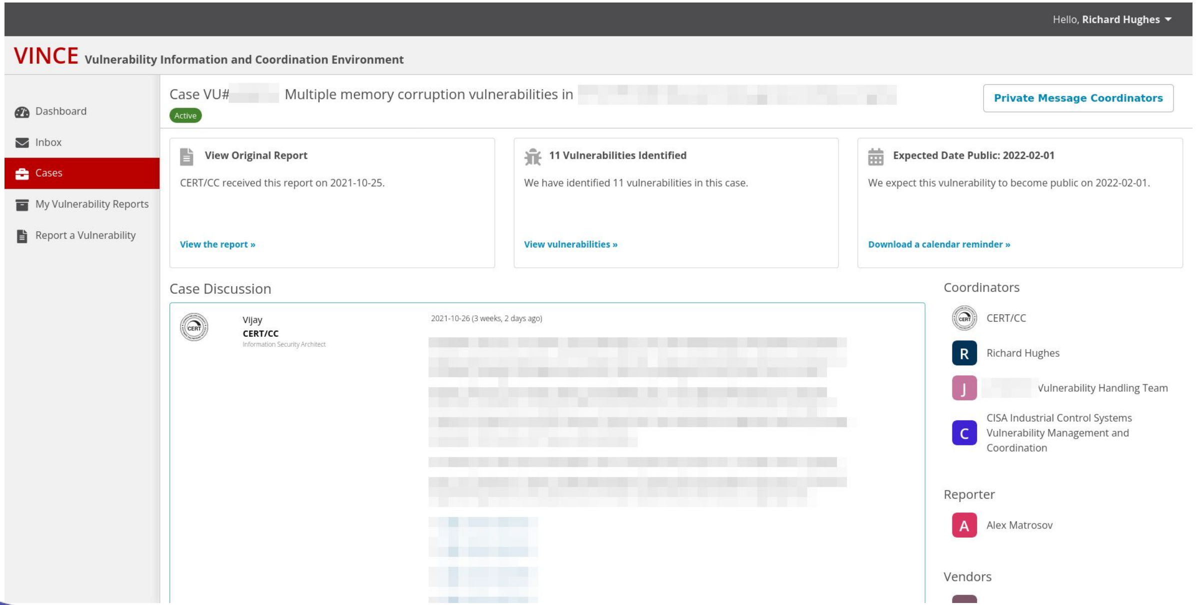


Add a new YARA or UefiAnalyzer query

```
UncoreInitPeimRefCode:
  ppi:
    - 1:
      - name: EFI_PEI_READ_ONLY_VARIABLE2_PPI_GUID
      - value: 2AB86EF5-ECB5-4134-B5563854CA1FE1B4
      - service:
        - name: LocatePpi
  wide_strings:
    1: syscg
    2: toolh
```

UefiAnalyzer

Use LVFS to find affected vendors & devices



Hello, **Richard Hughes**

VINCE **Vulnerability Information and Coordination Environment**

- Dashboard
- Inbox
- Cases
- My Vulnerability Reports
- Report a Vulnerability

Case VU#[redacted] Multiple memory corruption vulnerabilities in [redacted]

Active

Private Message Coordinators

View Original Report

CERT/CC received this report on 2021-10-25.

View the report »

11 Vulnerabilities Identified

We have identified 11 vulnerabilities in this case.

View vulnerabilities »

Expected Date Public: 2022-02-01

We expect this vulnerability to become public on 2022-02-01.

Download a calendar reminder »

Case Discussion

Vijay
CERT/CC
Information Security Architect

2021-10-26 (3 weeks, 2 days ago)

[redacted]

Coordinators

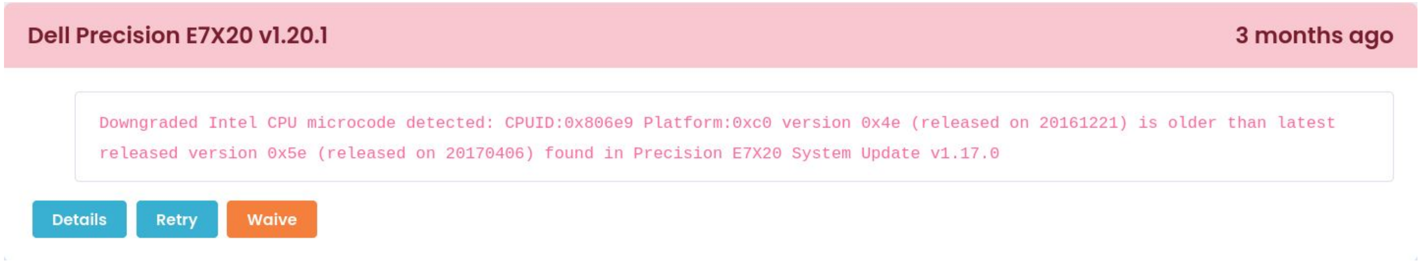
- CERT/CC
- Richard Hughes
- [redacted] Vulnerability Handling Team
- CISA Industrial Control Systems Vulnerability Management and Coordination

Reporter

- Alex Matrosov

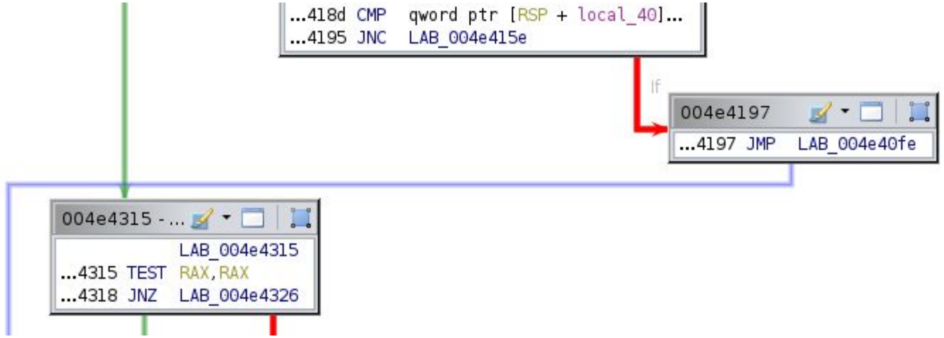
Vendors

Detect µcode downgrade, and **CVEs too?**



Dell Precision E7X20 v1.20.1 3 months ago

```
Downgraded Intel CPU microcode detected: CPUID:0x806e9 Platform:0xc0 version 0x4e (released on 20161221) is older than latest
released version 0x5e (released on 20170406) found in Precision E7X20 System Update v1.17.0
```

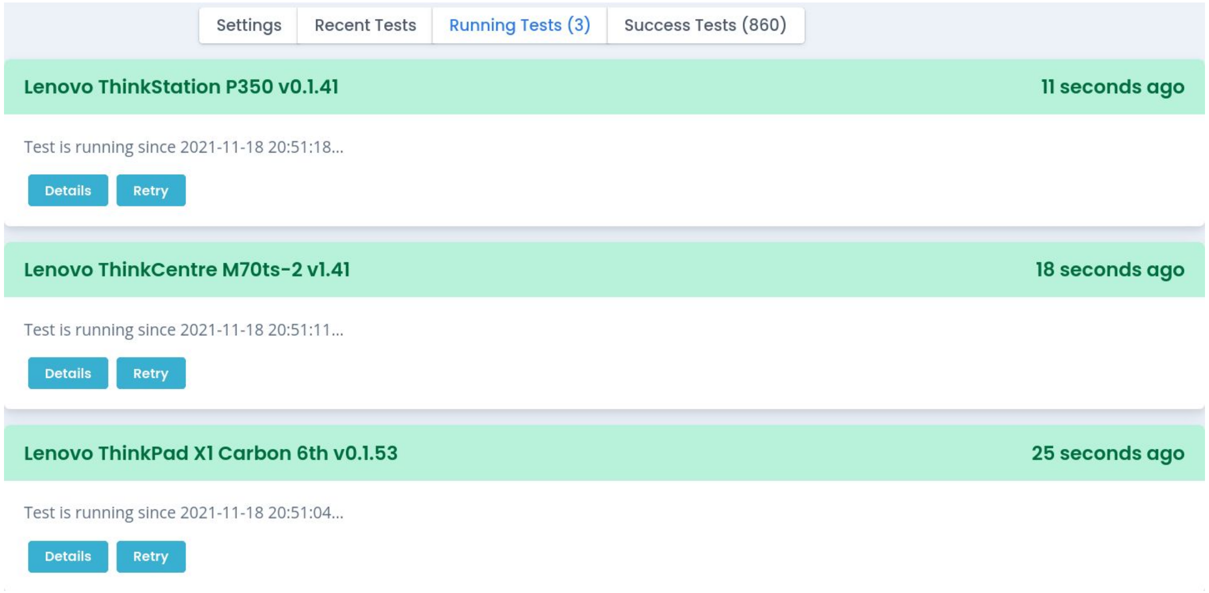


```
...418d CMP   qword ptr [RSP + local_40]...
...4195 JNC   LAB_004e415e

004e4197
...4197 JMP   LAB_004e40fe

004e4315 - ...
              LAB_004e4315
...4315 TEST  RAX,RAX
...4318 JNZ   LAB_004e4326
```

The LVFS run all rules on all firmware



Lenovo ThinkStation P350 v0.1.41 — 11 seconds ago

Test is running since 2021-11-18 20:51:18...

Lenovo ThinkCentre M70ts-2 v1.41 — 18 seconds ago

Test is running since 2021-11-18 20:51:11...

Lenovo ThinkPad X1 Carbon 6th v0.1.53 — 25 seconds ago

Test is running since 2021-11-18 20:51:04...

Open Source Tooling

- **Converged Security Suite**
  https://github.com/9elements/converged-security-suite
- **fwhunt and uefi_r2**
  https://github.com/binarly-io/uefi_r2
- **efiXplorer**
  https://github.com/binarly-io/efiXplorer
- **LVFS (Linux Vendor Firmware Service)**
  https://fwupd.org

How to get involved?

- Firmware Security Workstream
  - Specification and Guidelines
  - Incident Response and Disclosure recommendations
  - Supporting the vendors
    - Open Source Tooling
    - Documentation

Join us now

OPEN SOURCE FIRMWARE — FOUNDATION

opensourcefirmware.foundation

OPEN SOURCE FIRMWARE CONFERENCE

OPEN SOURCE FIRMWARE CONFERENCE

Agenda

- Who are we?
  - Company Intro (immune, binarly)
  - OSFF Workstream (immune)
- Problem Statement
  - Firmware Vulnerabilities Disclosure Timeline - Binarly
  - Explain the exploitation path - Binarly
  - AMI UsbRt and Intel BSSA DFT case studies - Binarly
  - What are vulnerability classifications? - immune
  - Lack of transparency/ independent verification of updates / supply chain - immune
- How do we solve this situation?
  - Why Another Spec? - immune/Binarly
  - Short explanation of the classification (in-depth in the workshop) - Binarly
  - IOC FwHunt - Binarly
  - LVFS to verify what is out there (c.f. lack of transparency) (Richard)
  - Open Source Tooling (Pentesting, Provisioning in the supply chain) - immune
- How to get involved?
  - Concrete Action Points (Alex, Philipp)
  - Link to application form (relevant skills?, time commitment?) (Alex, Philipp)