Implant, bootkit and boot protection

November 20, 2024
Sébastien BRILLET DGA

Summary

1. Boot phases overview

2. Threats targeting the boot phase
   - ➢ implant and bootkit

3. Secure boot : protection mechanism

4. Focus on Blacklotus malware (and others)

5. Best Practices

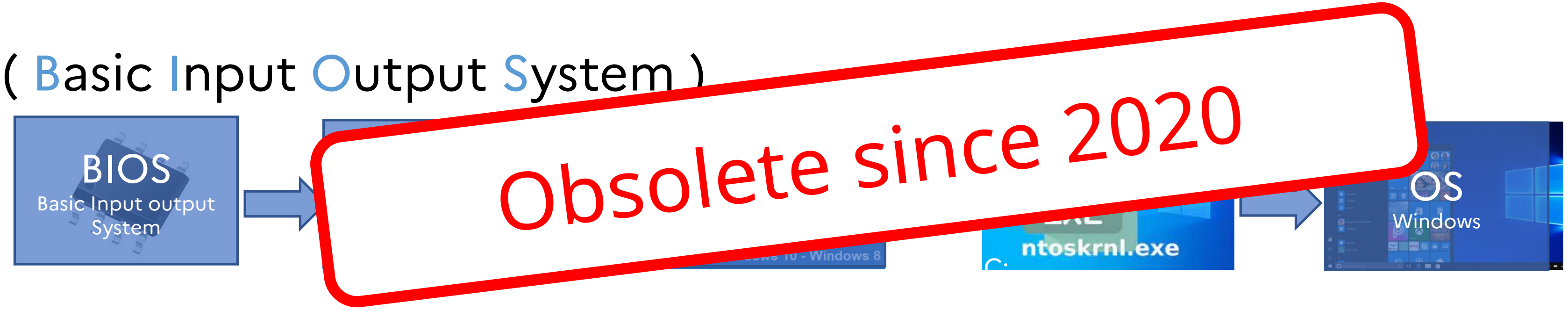
OVERVIEW OF THE BOOT PHASES

System boot firmware BIOS/UEFI

Firmware :

- low-level software
- controls hardware or peripherals
- boots before the Operating System

(Basic Input Output System)



BIOS
Basic Input output System

ws 10 - Windows 8

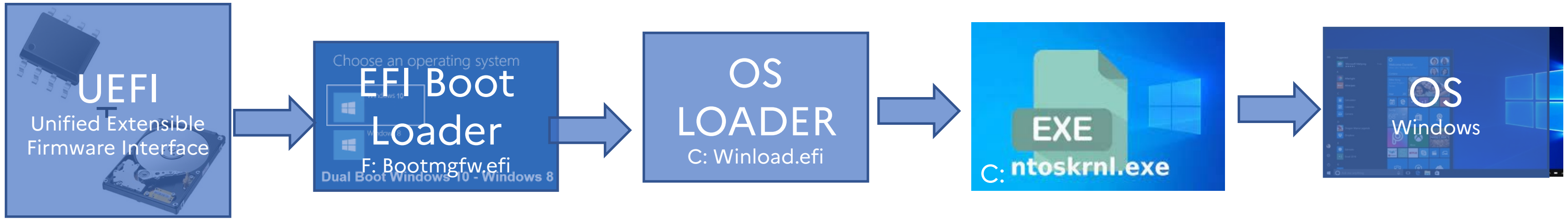
EXE
C: ntoskrnl.exe

OS
Windows

Obsolete since 2020

We will focus on **UEFI**

(Unified Extensible Firmware Interface)



UEFI
Unified Extensible Firmware Interface

Choose an operating system
EFI Boot Loader
F: Bootmgfw.efi
Dual Boot Windows 10 - Windows 8

OS LOADER
C: Winload.efi

EXE
C: ntoskrnl.exe

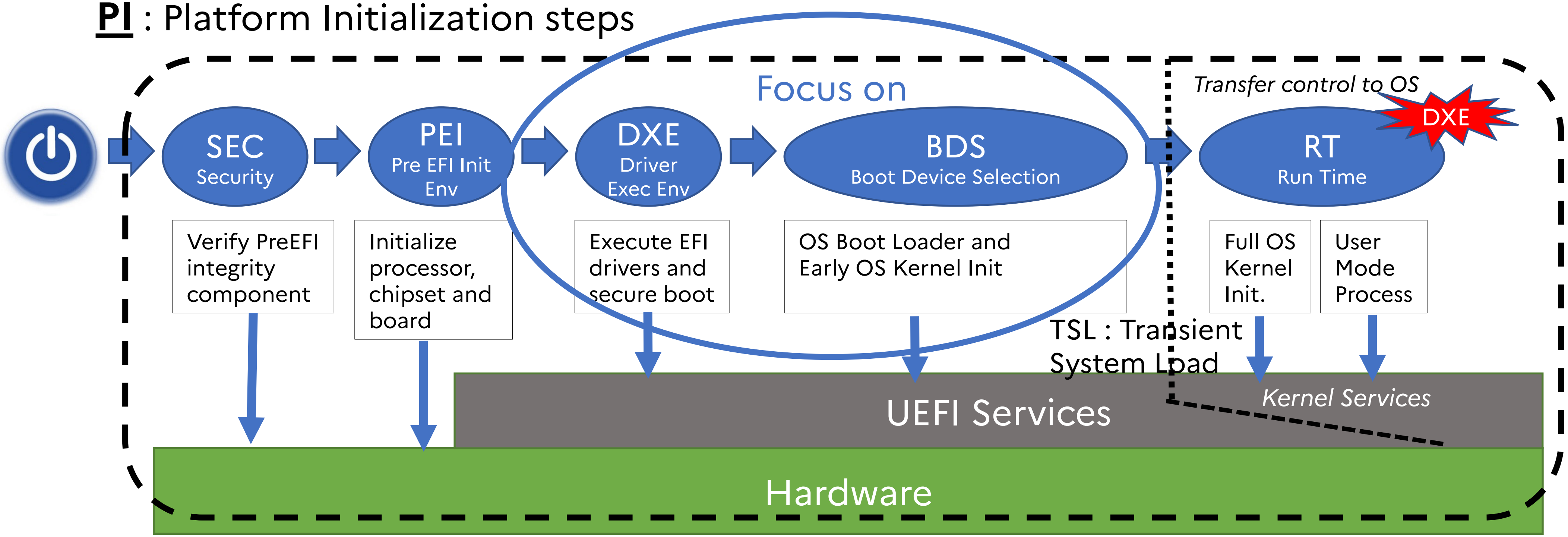
OS
Windows

UEFI (Unified Extensible Firmware Interface) and PI (Platform Initialization)

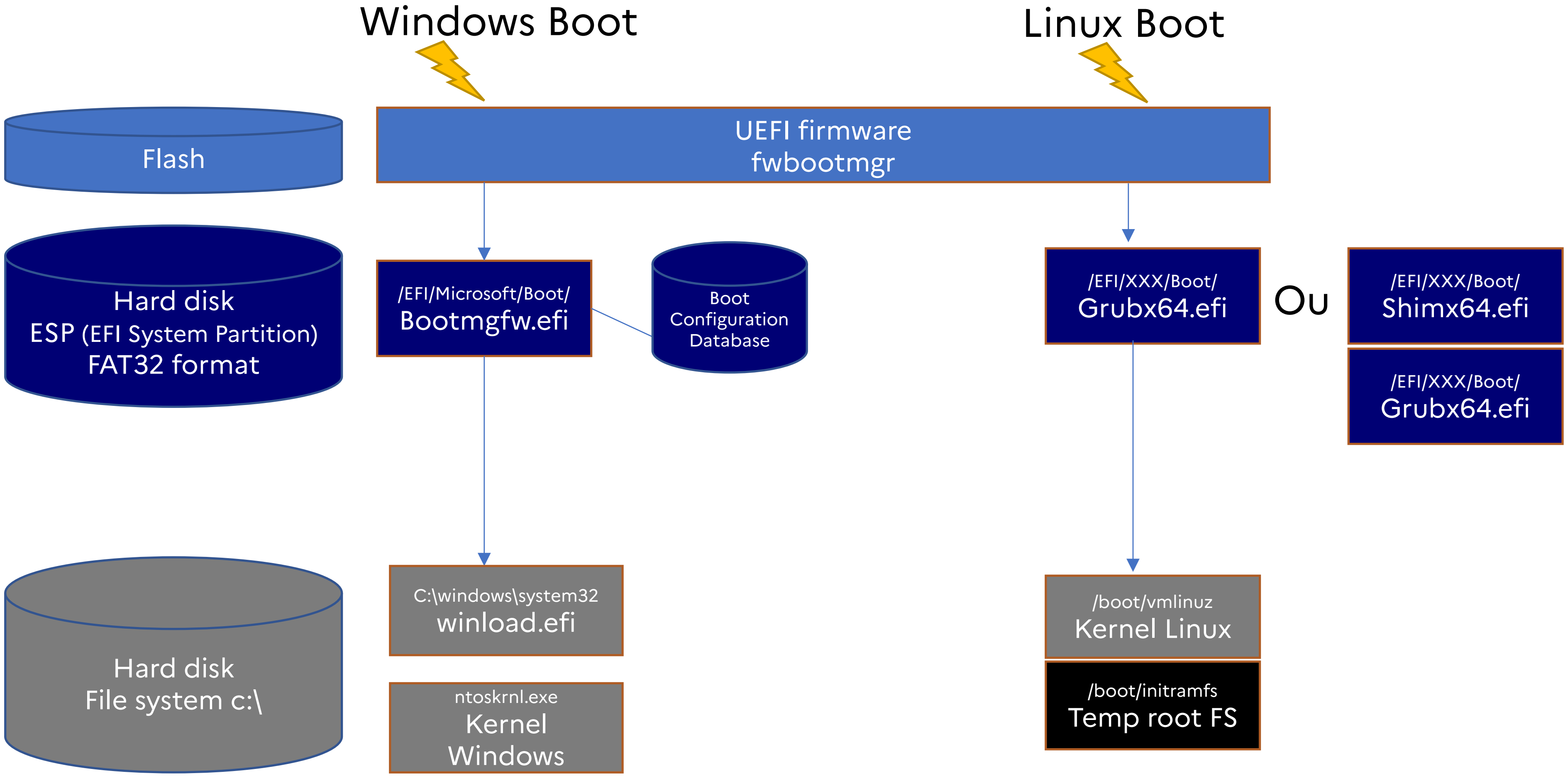
UEFI : Interface between firmware and OS while a computer is being booted.

- Standard from **2005**, last specification => 2,10,A **(2024-08)**
- Specific framework available to **develop** UEFI applications and drivers

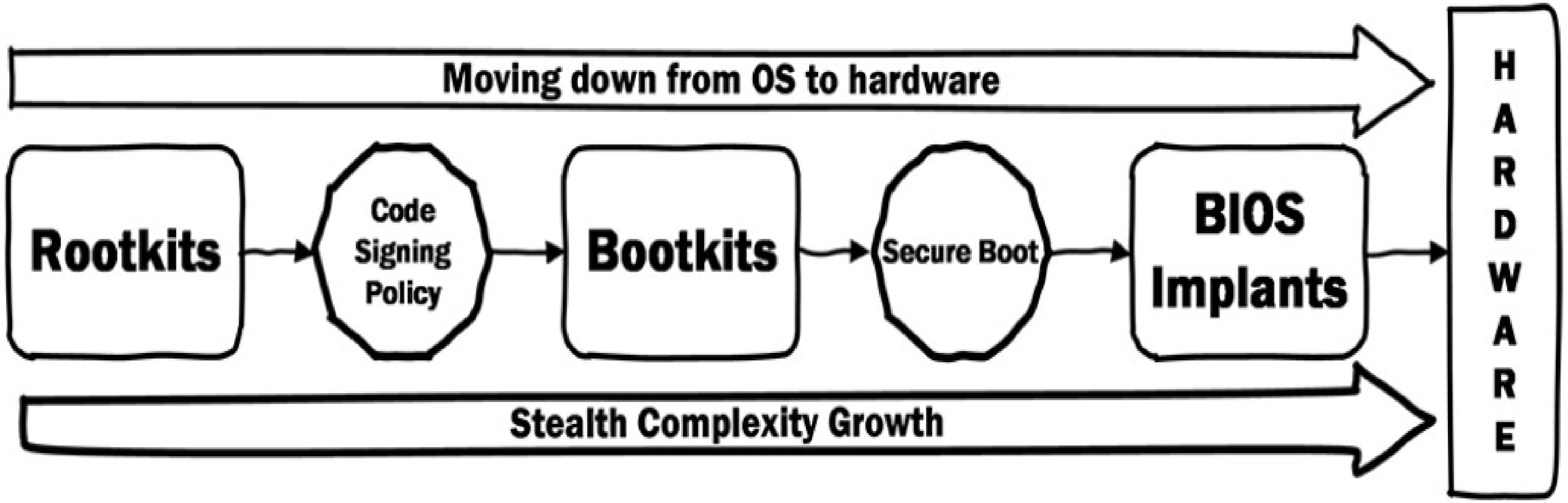
PI : Platform Initialization steps



Simplified view of a PC's UEFI boot process



THREATS TARGETING THE BOOT PHASE


Moving down from OS to hardware

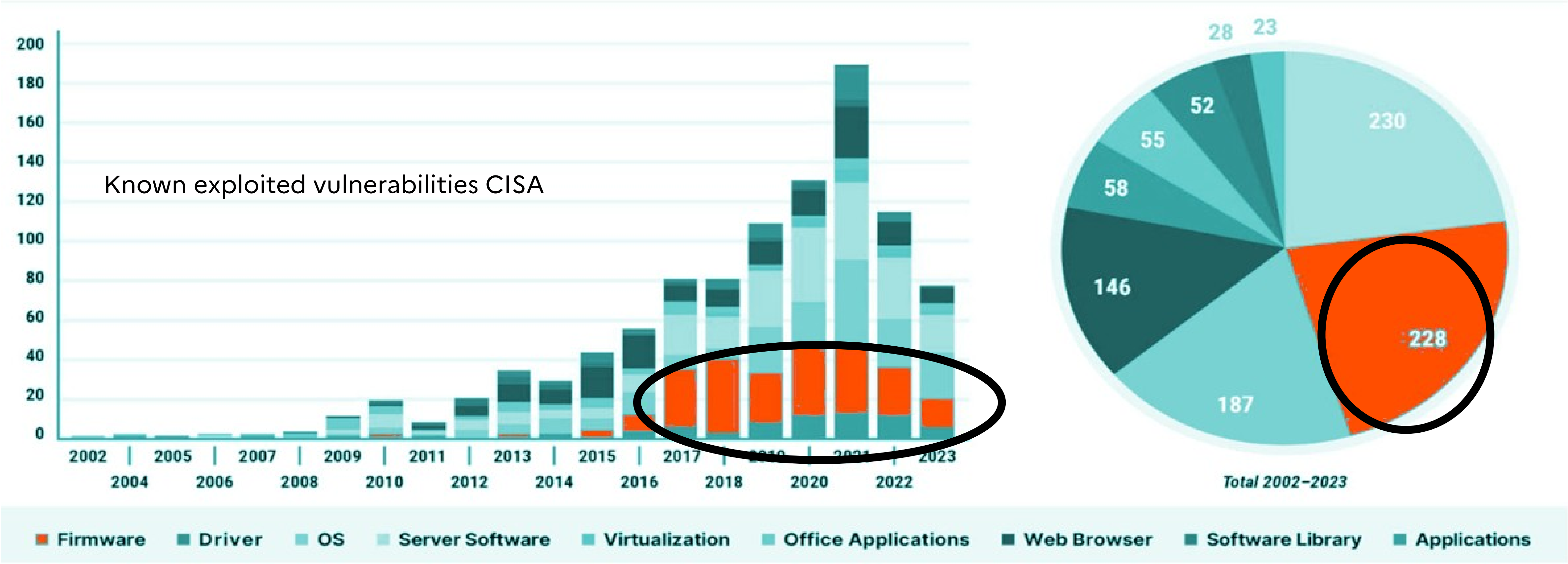
Rootkits → Code Signing Policy → Bootkits → Secure Boot → BIOS Implants → HARDWARE

Stealth Complexity Growth

https://github.com/eclypsium/Publications/blob/master/2023/BruCON/Ghosts%20in%20the%20Machine%20-%20BruCON%200x0F.pdf

Firmware vulnerability

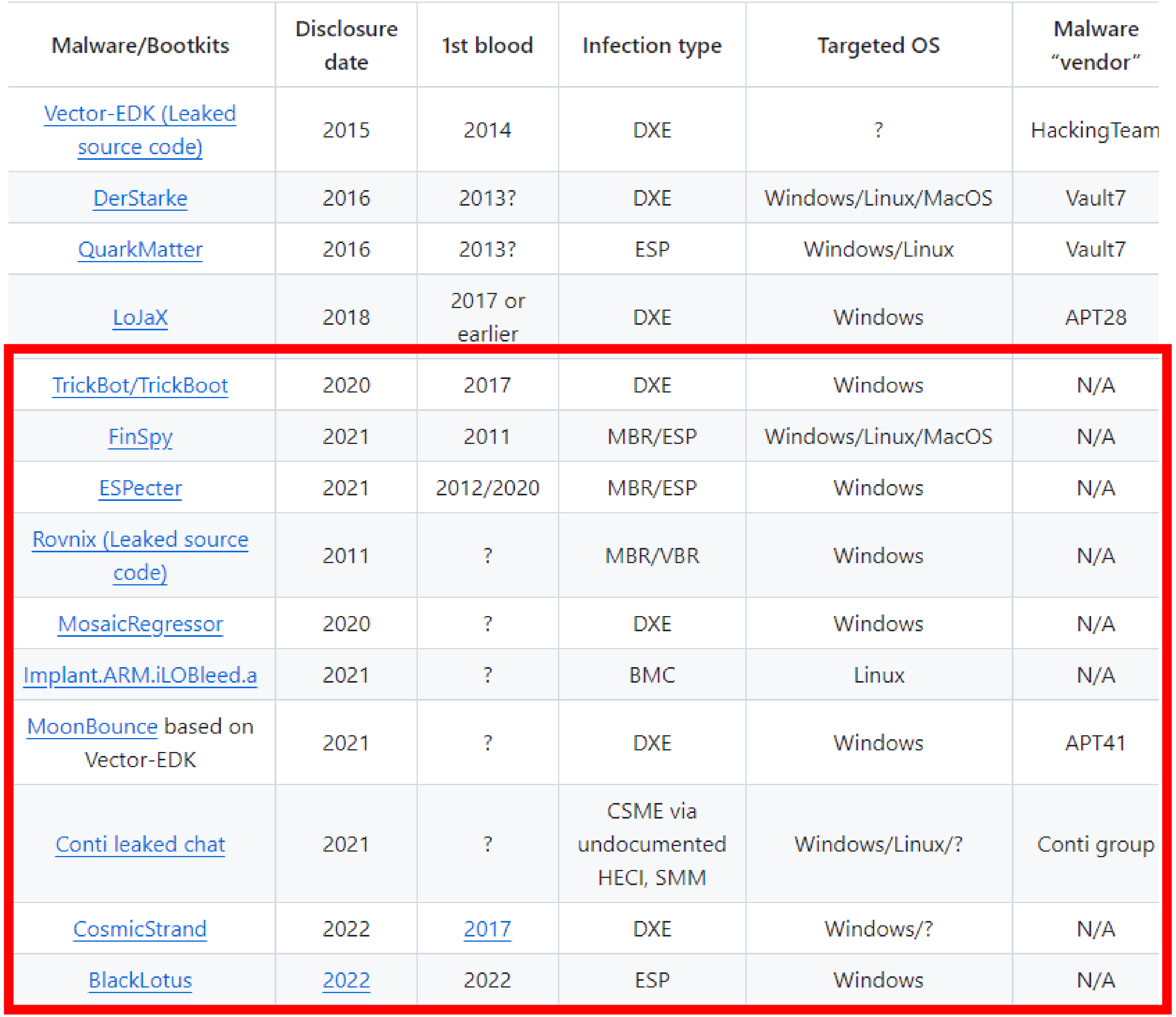
Since 2017, acceleration of discovered vulnerabilities (228 since 2002)


Known exploited vulnerabilities CISA

Firmware Threats

Since 2020, increase in discovered bootkits/implants.

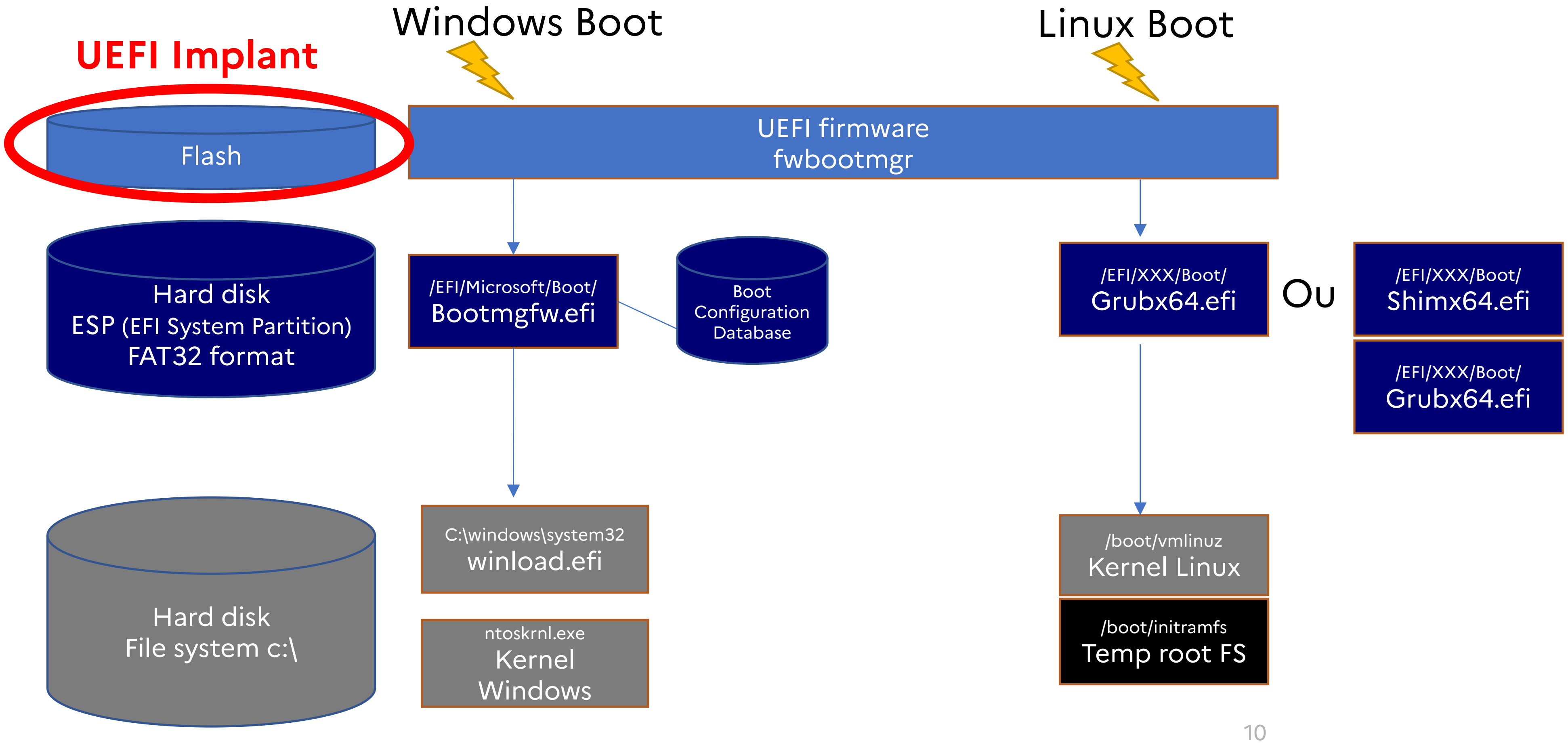
https://github.com/hardenedvault/bootkit-samples



Malware/Bootkits	Disclosure date	1st blood	Infection type	Targeted OS	Malware "vendor"
Vector-EDK (Leaked source code)	2015	2014	DXE	?	HackingTeam
DerStarke	2016	2013?	DXE	Windows/Linux/MacOS	Vault7
QuarkMatter	2016	2013?	ESP	Windows/Linux	Vault7
LoJaX	2018	2017 or earlier	DXE	Windows	APT28
TrickBot/TrickBoot	2020	2017	DXE	Windows	N/A
FinSpy	2021	2011	MBR/ESP	Windows/Linux/MacOS	N/A
ESPecter	2021	2012/2020	MBR/ESP	Windows	N/A
Rovnix (Leaked source code)	2011	?	MBR/VBR	Windows	N/A
MosaicRegressor	2020	?	DXE	Windows	N/A
Implant.ARM.iLOBleed.a	2021	?	BMC	Linux	N/A
MoonBounce based on Vector-EDK	2021	?	DXE	Windows	APT41
Conti leaked chat	2021	?	CSME via undocumented HECI, SMM	Windows/Linux/?	Conti group
CosmicStrand	2022	2017	DXE	Windows/?	N/A
BlackLotus	2022	2022	ESP	Windows	N/A

UEFI Implant

UEFI implants modify UEFI firmware stored in flash.



Flash

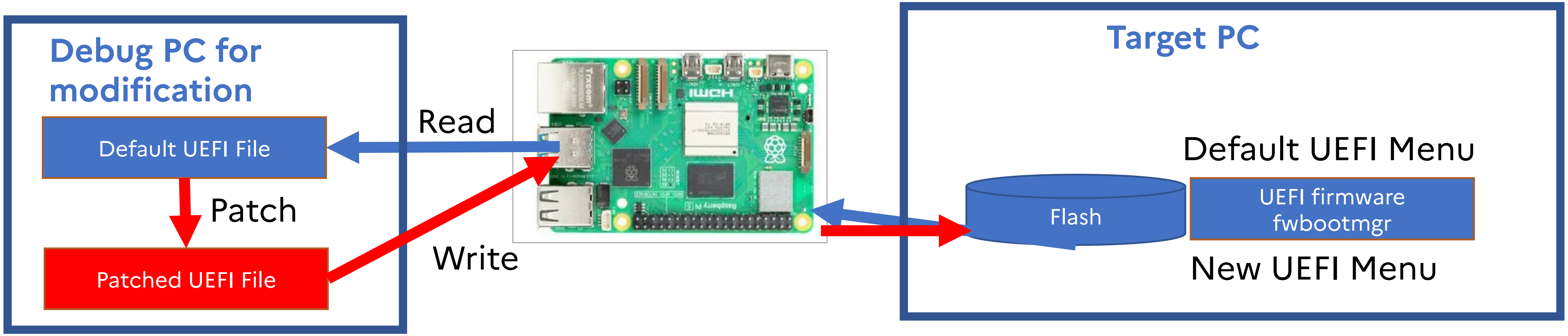
UEFI firmware
fwbootmgr

UEFI Implant characteristics

- UEFI implants are **OS independent**.
- UEFI implants have very good **persistence**:
  - ➢ They reside in flash memory.
  - ➢ They withstand hard drive formatting or OS changes.
  - ➢ They persist on PCs booting from CD-ROMs
- Main deployment vectors :
  - ➢ Exploiting a **vulnerability** in the UEFI firmware.
  - ➢ Having the ability to modify the **firmware of a device**, for example, PCI.
  - ➢ Conducting a **supply chain attack** on UEFI firmware updates.
  - ➢ Gaining **physical access** to the machine to access the SPI interface of the flash.

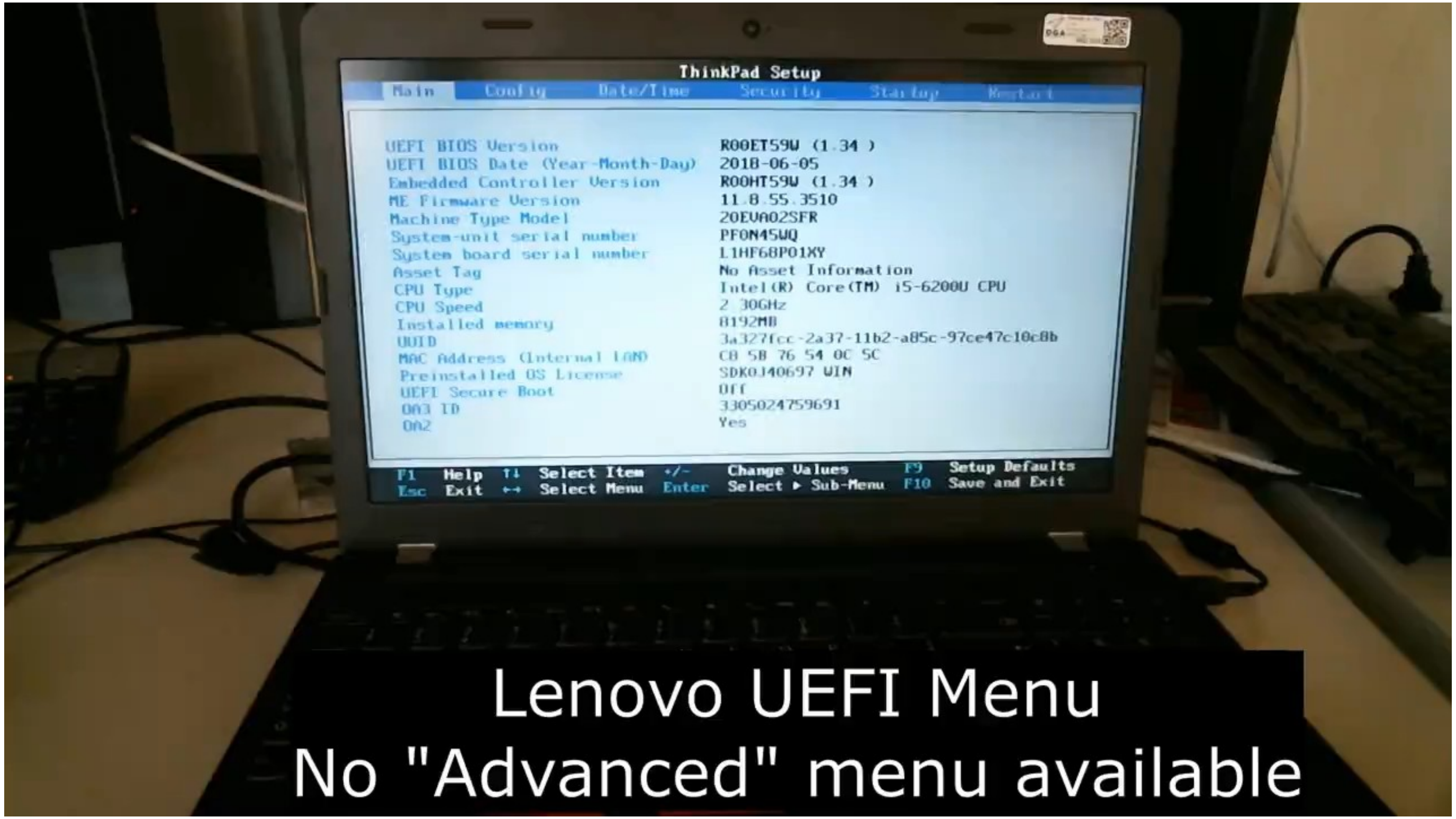
Non offensive deployment example

- **Mock implant** : Activate advanced option in the UEFI menu (processor debug mode)
- **Solution :** Modify the flash contents to activate this option in the UEFI menu



Please note that this is a proof of concept. Using this kind of method to implement a feature at the UEFI level is not advisable

Non offensive deployment example



```
                              ThinkPad Setup
 Main       Config      Date/Time      Security      Startup      Restart

 UEFI BIOS Version                        R00ET59W (1.34 )
 UEFI BIOS Date (Year-Month-Day)          2018-06-05
 Embedded Controller Version              R00HT59W (1.34 )
 ME Firmware Version                      11.8.55.3510
 Machine Type Model                       20EVA02SFR
 System-unit serial number                PF0N45WQ
 System board serial number               L1HF68P01XY
 Asset Tag                                No Asset Information
 CPU Type                                 Intel(R) Core(TM) i5-6200U CPU
 CPU Speed                                2.30GHz
 Installed memory                         8192MB
 UUID                                     3a327fcc-2a37-11b2-a85c-97ce47c10c8b
 MAC Address (Internal LAN)               C8 5B 76 54 0C 5C
 Preinstalled OS License                  SDK0J40697 WIN
 UEFI Secure Boot                         Off
 OA3 ID                                   3305024759691
 OA2                                      Yes

 F1   Help   ↑↓  Select Item   +/-    Change Values         F9   Setup Defaults
 Esc  Exit   ↔   Select Menu   Enter  Select ▸ Sub-Menu     F10  Save and Exit
```

Lenovo UEFI Menu
No "Advanced" menu available

https://www.welivesecurity.com/2018/09/27/lojax-first-uefi-rootkit-found-wild-courtesy-sednit-group/

Real example : UEFI Implant - Lojax

- September 2018, found by ESET
- ≈ APT28 (Fancy Bear –RU)

- Deployment use software **vulnerability** in Computrace Lojak (Absolutelojack)
  - Allows locating a stolen PC
  - Pre-installed software in the firmware of many laptops which launches Windows Agent

- Deployment :
  - ❑ Accesses flash from the OS thanks to the Lojax agent
  - ❑ Bypass flash write protections to patch firmware in flash
- Payload :
  - ❑ Replace legitimate autocheck.exe
  - ❑ Install a windows service and contact C&C

Some protections

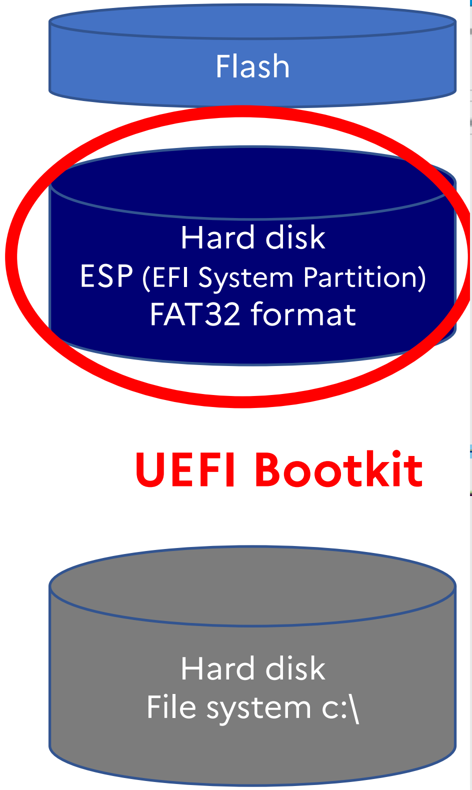
- **Check security protection**
  - Write protection on the flash

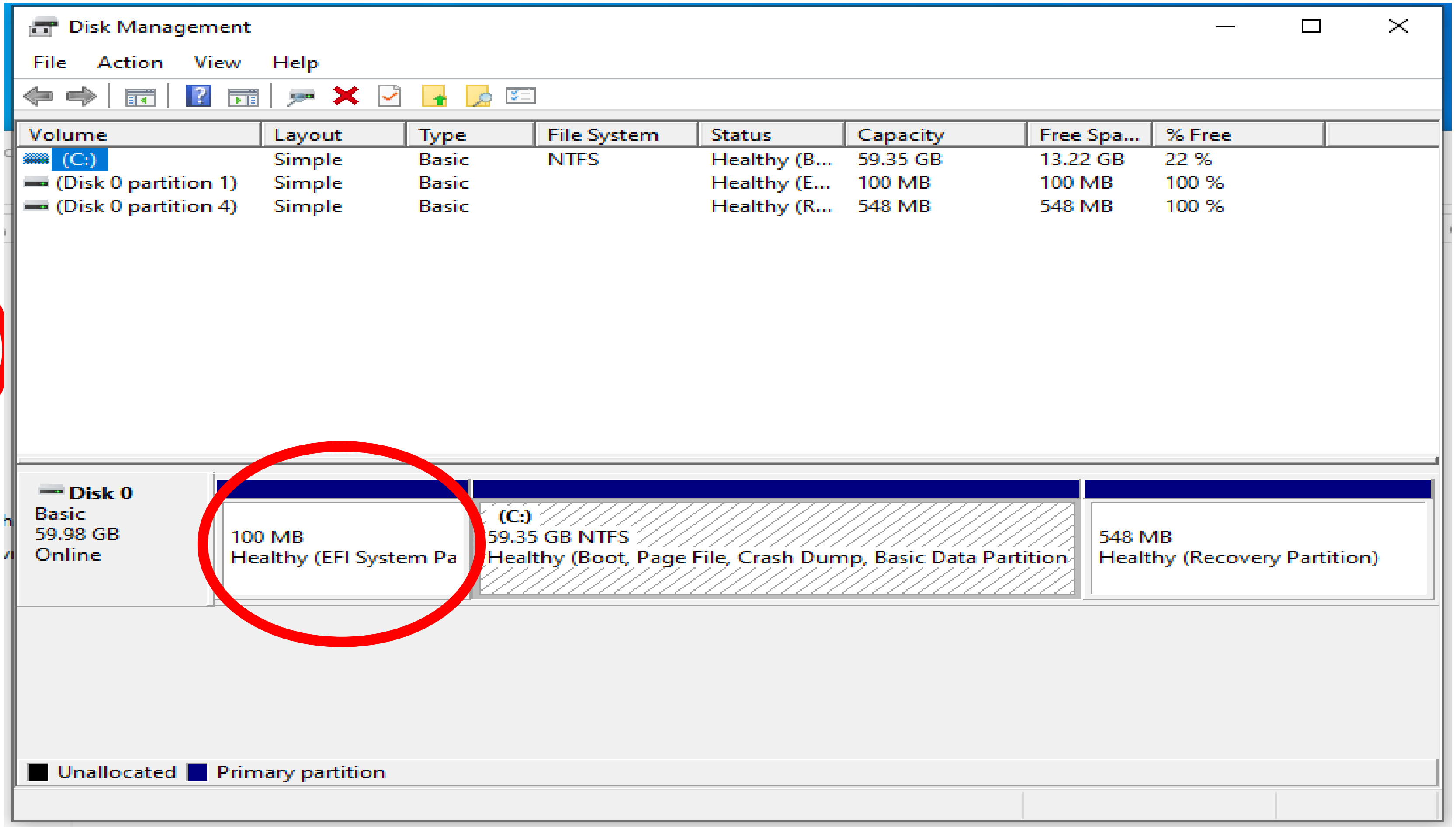
- **Use recent processors with security features**
  - Flash integrity verification (for example : Intel boot guard pour Intel ou PSP pour AMD)

- **Update the Firmware** : Firmware should be updated regularly and treated as importantly as operating system and application updates.

UEFI bootkits

UEFI bootkits will install themselves on the EFI partition of the hard drive.





Volume	Layout	Type	File System	Status	Capacity	Free Spa...	% Free
(C:)	Simple	Basic	NTFS	Healthy (B...	59.35 GB	13.22 GB	22 %
(Disk 0 partition 1)	Simple	Basic		Healthy (E...	100 MB	100 MB	100 %
(Disk 0 partition 4)	Simple	Basic		Healthy (R...	548 MB	548 MB	100 %

Disk 0
Basic
59.98 GB
Online

100 MB
Healthy (EFI System Pa

(C:)
59.35 GB NTFS
Healthy (Boot, Page File, Crash Dump, Basic Data Partition

548 MB
Healthy (Recovery Partition)

■ Unallocated ■ Primary partition

UEFI bootkits characteristics

- UEFI bootkits are **OS independent**.
- No need to have access to flash.
- **Bootkits** can be generic:
  - ❑ They execute with elevated privileges before the OS and allow for the deployment of payloads in kernel mode and/or user mode.
  - ❑ They can be installed remotely.
- However, they **do not survive the installation of a new OS**

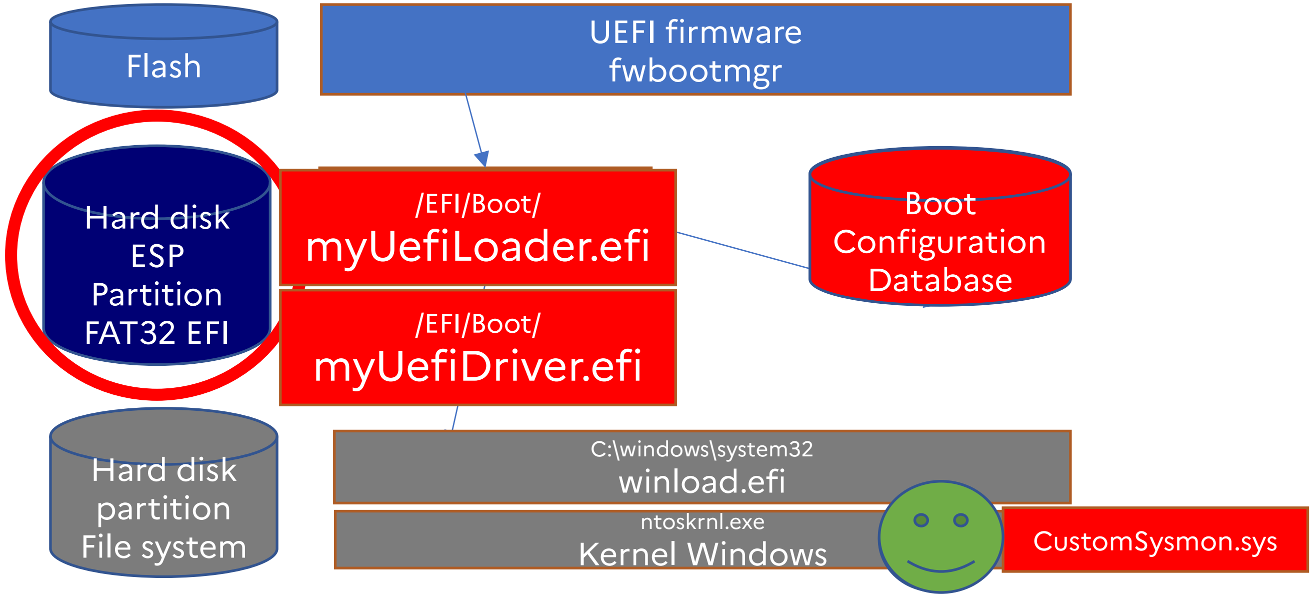
Non offensive deployment example

- **Mock bootkit** : Install and test our own drivers on Windows in "normal" mode (bypass Microsoft signature verification).

- **Solution :** Use a UEFI driver that allows disabling signature verification on windows.

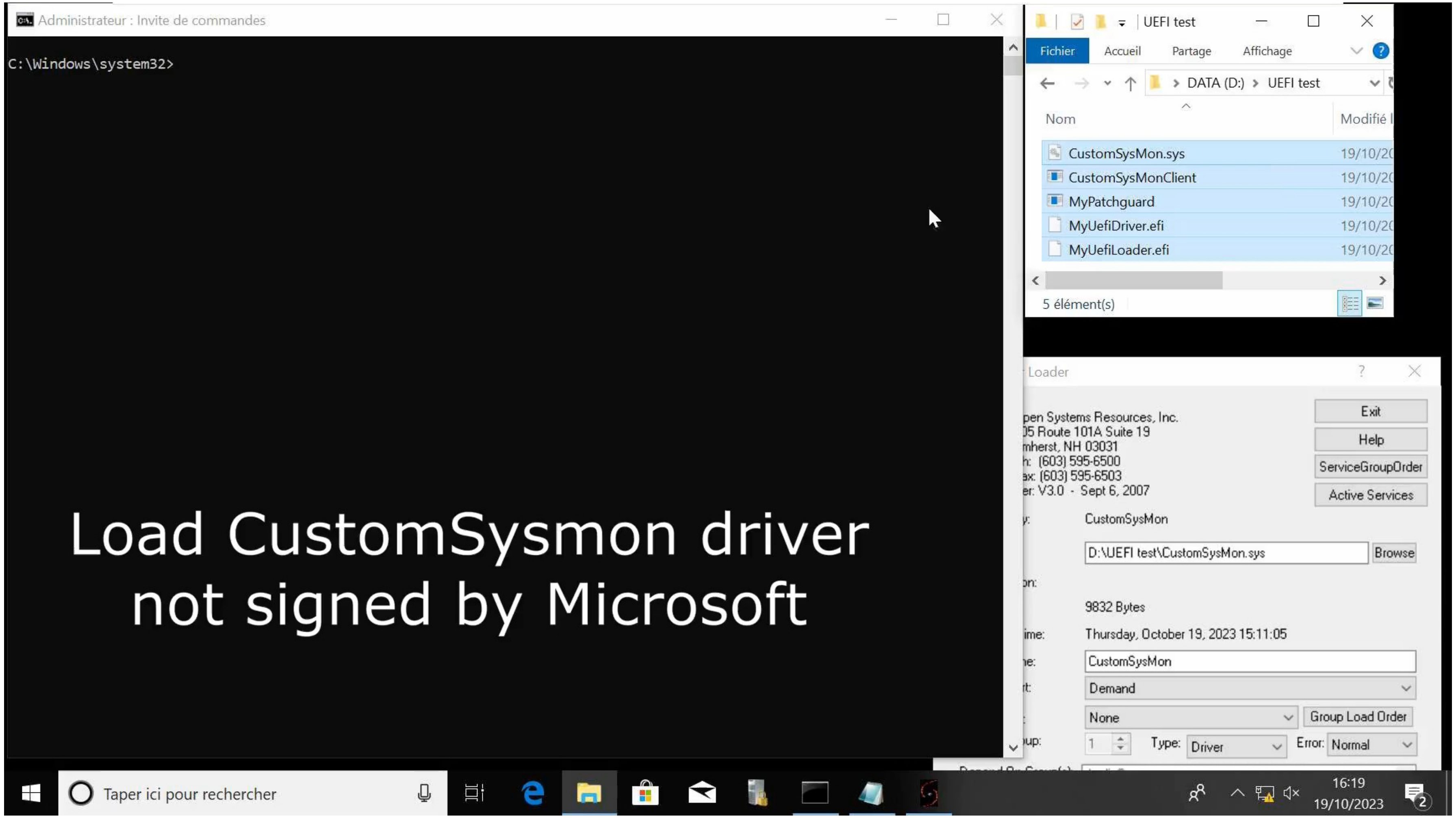
Modification of the ESP partition to load a UEFI driver to disable the OS driver signature verification

Modify boot configuration database to load the UEFI application myUEFILoader.efi



Please note that this is a proof of concept. Using this kind of method to implement a feature at the UEFI level is not advisable https://github.com/Mattiwatti/EfiGuard

Non offensive deployment example



Administrateur : Invite de commandes

```
C:\Windows\system32>
```

Load CustomSysmon driver
not signed by Microsoft

Nom	Modifié l
CustomSysMon.sys	19/10/20
CustomSysMonClient	19/10/20
MyPatchguard	19/10/20
MyUefiDriver.efi	19/10/20
MyUefiLoader.efi	19/10/20

5 élément(s)

Loader

pen Systems Resources, Inc.
05 Route 101A Suite 19
mherst, NH 03031
h: (603) 595-6500
ax: (603) 595-6503
er: V3.0 - Sept 6, 2007

y: CustomSysMon

D:\UEFI test\CustomSysMon.sys

on:

9832 Bytes

ime: Thursday, October 19, 2023 15:11:05

ne: CustomSysMon

rt: Demand

None

up: 1 Type: Driver Error: Normal

Diving Into Glupteba's UEFI Bootkit (paloaltonetworks.com)

https://www.welivesecurity.com/2021/10/05/uefi-threats-moving-esp-introducing-especter-bootkit/

Real Example : Bootkit UEFI - ESPecter

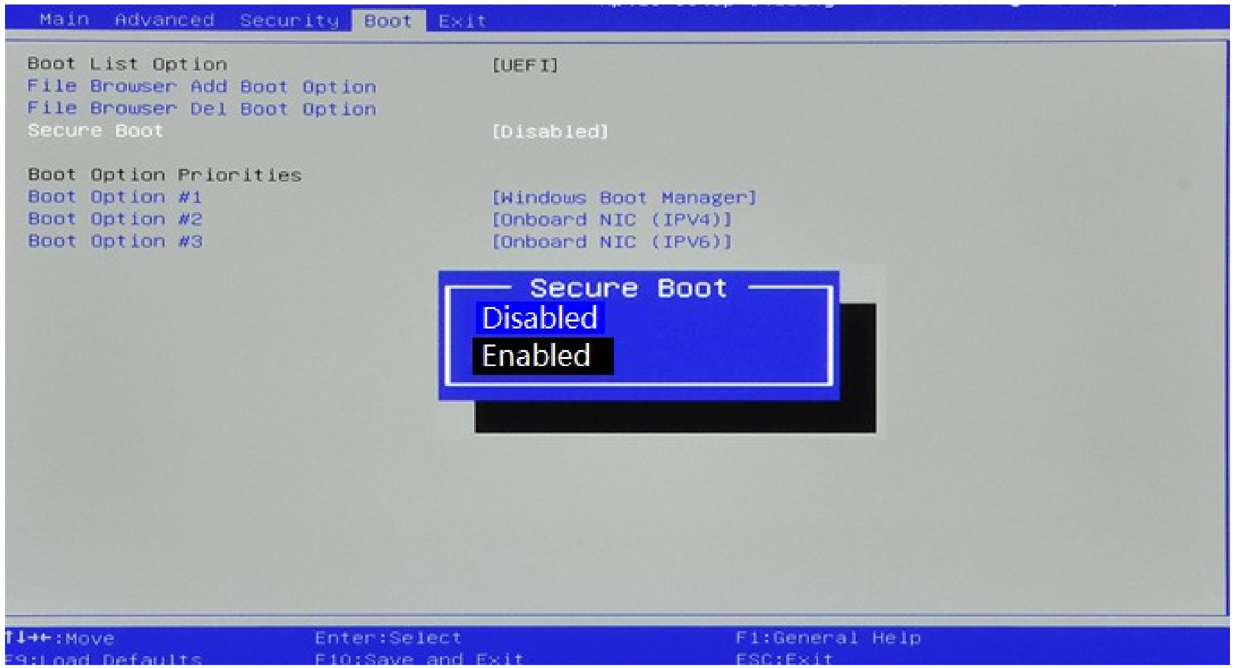
- October 2021, found by ESET
- No attribution

- Focus on Windows machine.
  - Active since 2012: initially targets the BIOS.
  - Updated in 2020 to target UEFI.

- Deployment :
  - ❑ Modify bootmgfw.efi file
  - ❑ Bypasses Windows driver signature verification and loads its own driver.
- Payload :
  - ❑ Install keylogger and injects itself into a system process
  - ❑ Contacts its C&C

2024 : Glupteba (modular malware) has added a UEFI bootkit to its attack arsenal.

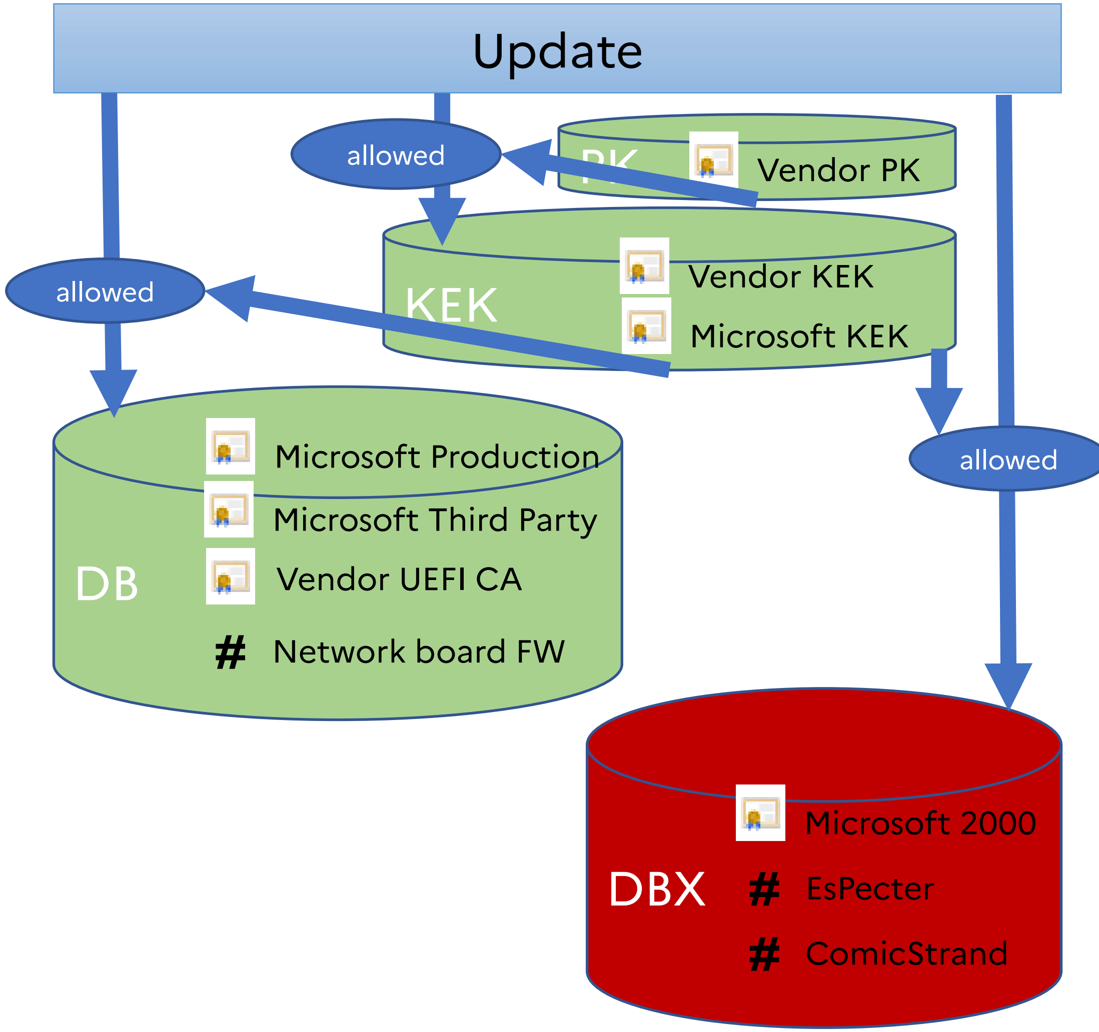
SECURE BOOT



```
Boot List Option                      [UEFI]
File Browser Add Boot Option
File Browser Del Boot Option
Secure Boot                           [Disabled]

Boot Option Priorities
Boot Option #1                        [Windows Boot Manager]
Boot Option #2                        [Onboard NIC (IPV4)]
Boot Option #3                        [Onboard NIC (IPV6)]

                         Secure Boot
                      Disabled
                      Enabled

↑↓→←:Move           Enter:Select            F1:General Help
F9:Load Defaults    F10:Save and Exit       ESC:Exit
```

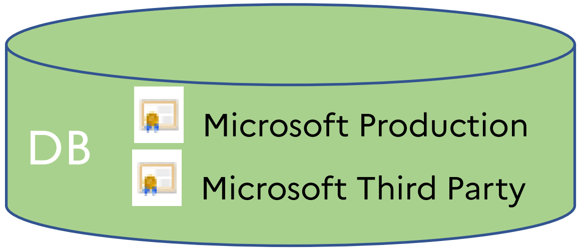
Secure Boot key

- **Platform Key Database (PK) :**
  - Contains master key certificate (Vendor)
  - Protect KEK from uncontrolled modification
- **Key Exchange Keys Database (KEKs) :**
  - Contains vendors and Microsoft certificates
  - Protect DB and DBX
- **Allow list Database (DB) :**
  - Contains public key certificates or hashes.
  - Binaries that can be validated by a certificate or hash will be allowed to execute at boot time
- **Deny list Database (DBX) :**
  - Contains public key certificates or hashes.
  - Any binary hash that matches a DBX hash or has a signature verified by a DBX certificate will be prevented from executing at boot time.
  - **The DBX has ultimate veto power at boot time.**

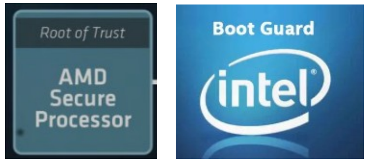


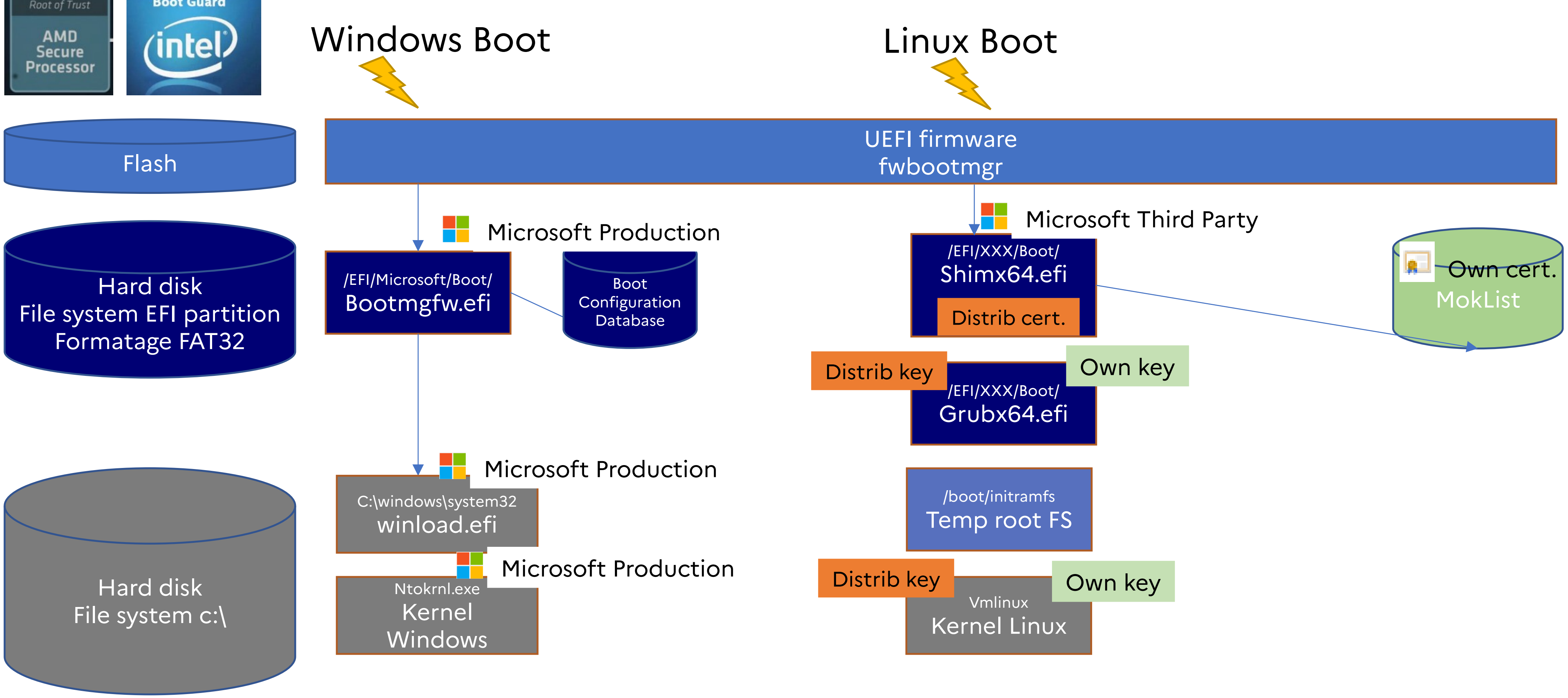
2024 – PKFail : firmware **supply-chain issue**.
The problem arises from the Secure Boot **Platform Key (PK)**, which is **untrusted** because it is generated by Independent BIOS Vendors and shared among different vendors.

Protect PC with UEFI secure boot.



DB
Microsoft Production
Microsoft Third Party





Windows Boot

Linux Boot

Flash

UEFI firmware
fwbootmgr

Hard disk
File system EFI partition
Formatage FAT32

Microsoft Production

/EFI/Microsoft/Boot/
Bootmgfw.efi

Boot
Configuration
Database

Microsoft Production

C:\windows\system32
winload.efi

Microsoft Production

Ntokrnl.exe
Kernel
Windows

Hard disk
File system c:\

Microsoft Third Party

/EFI/XXX/Boot/
Shimx64.efi
Distrib cert.

Own cert.
MokList

Distrib key

Own key

/EFI/XXX/Boot/
Grubx64.efi

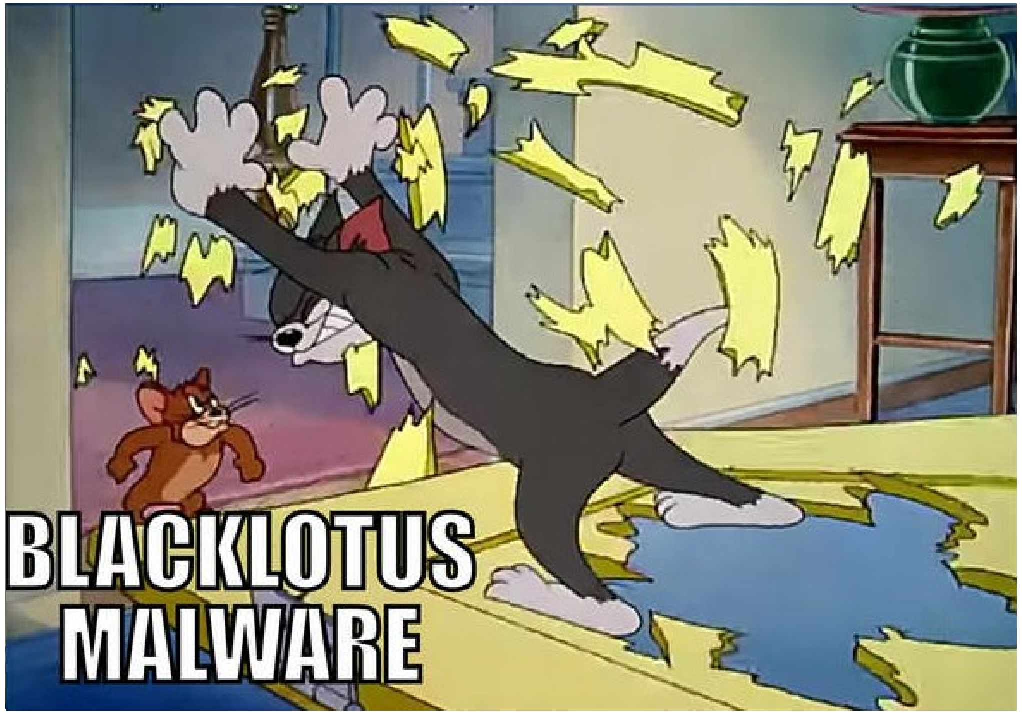
/boot/initramfs
Temp root FS

Distrib key

Own key

Vmlinux
Kernel Linux

BLACKLOTUS



https://www.welivesecurity.com/2023/03/01/blacklotus-uefi-bootkit-myth-confirmed/

What is Blacklotus malware

Blacklotus is a bootkit that is sold on hacking forums for **$5,000** since at least **October 2022.**

- ❑ Works on the latest Windows 11 systems
- ❑ Works with Secure Boot UEFI enabled
- ❑ BlackLotus does not install if the compromised host uses one of the following locales :
    - ❑ Roumain (Moldavie), ro-MD
    - ❑ Russe (Moldavie), ru-MD
    - ❑ Russe (Russie), ru-RU
    - ❑ Ukrainien (Ukraine) , uk-UA
    - ❑ Biélorusse (Biélorussie), be-BY
    - ❑ Arménien (Arménie), hy-AM
    - ❑ Kazakh (Kazakhstan), kk-KZ

Blacklotus infection steps

1. Download and execute an installer
2. Bypass UAC (User Account Control)
3. Disable Bitlocker and Windows Defender.
4. Install files on ESP partition (old legitimate files with baton drop CVE)

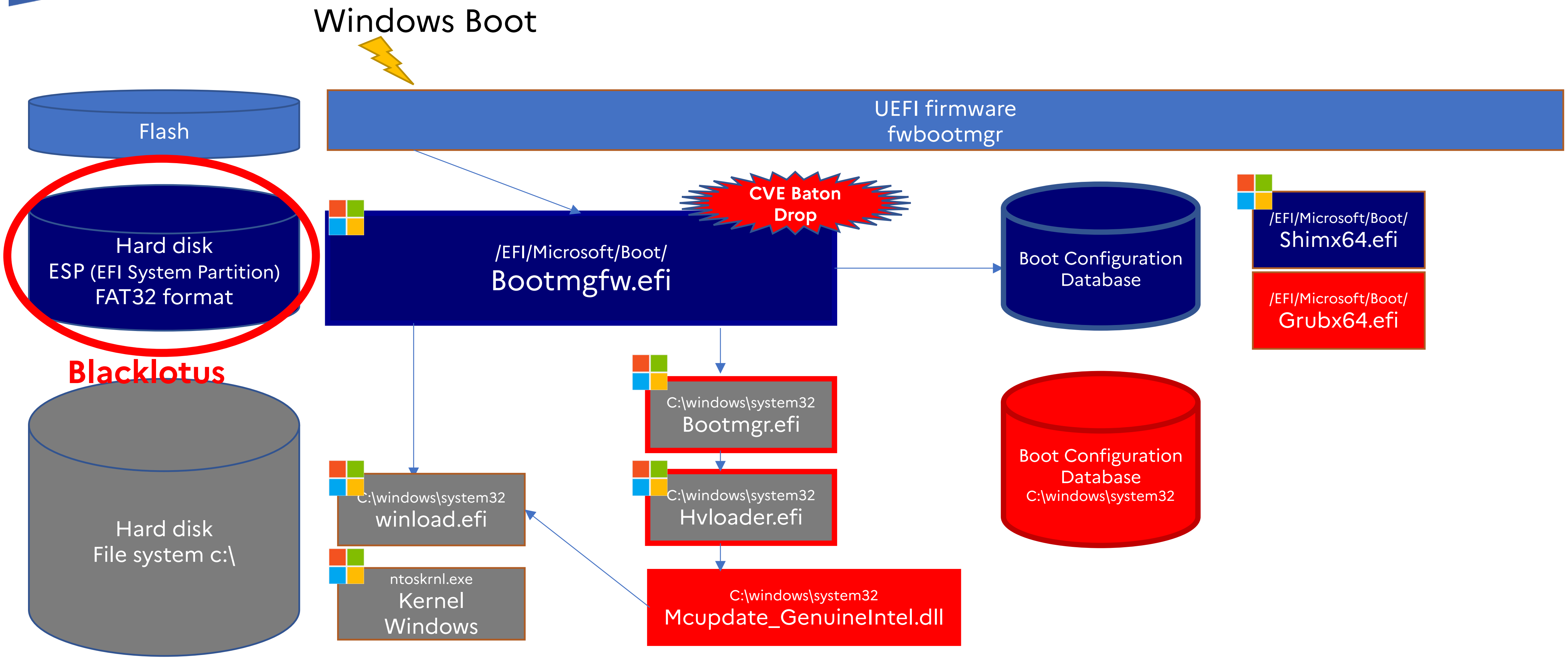
---------- Reboot ------------

5. Use **CVE-2022-21894 Baton Drop** vulnerability on windows bootmgfw.efi file
6. Add MOK List
7. Replace file bootmgfw.efi with simx64.efi

---------- Reboot ------------

8. Grubx64.efi will hook the winload.efi (OS windows loader) to install an infected driver
9. The driver injects some code in Winlogon process which contacts the C&C

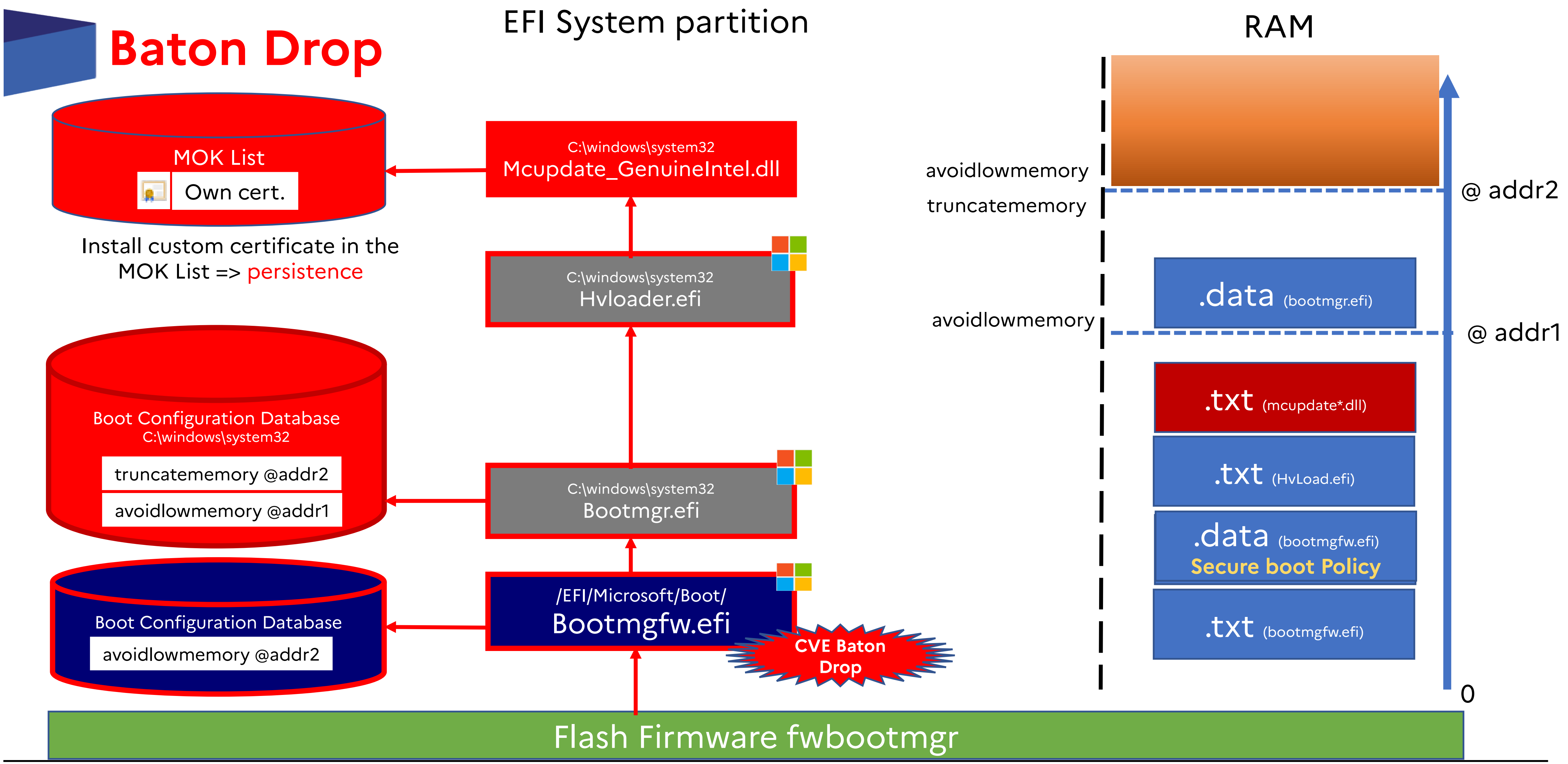
Blacklotus UEFI infection



Restarting

https://github.com/Wack0/CVE-2022-21894

Baton Drop



EFI System partition

RAM

MOK List

Own cert.

Install custom certificate in the MOK List => persistence

C:\windows\system32
Mcupdate_GenuineIntel.dll

C:\windows\system32
Hvloader.efi

Boot Configuration Database
C:\windows\system32

truncatememory @addr2

avoidlowmemory @addr1

C:\windows\system32
Bootmgr.efi

Boot Configuration Database

avoidlowmemory @addr2

/EFI/Microsoft/Boot/
Bootmgfw.efi

CVE Baton Drop

avoidlowmemory

truncatememory

@ addr2

.data (bootmgr.efi)

avoidlowmemory

@ addr1

.txt (mcupdate*.dll)

.txt (HvLoad.efi)

.data (bootmgfw.efi)
Secure boot Policy

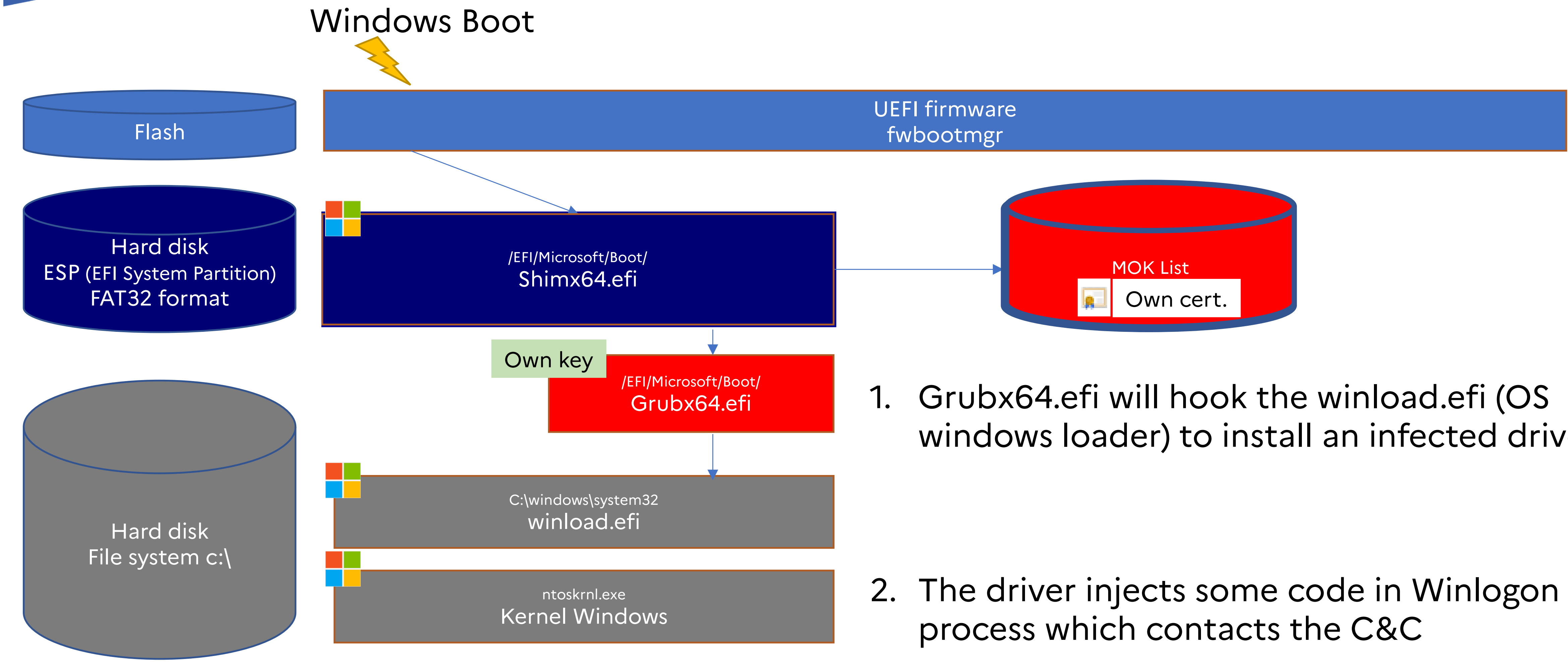
.txt (bootmgfw.efi)

0

Flash Firmware fwbootmgr

Restarting

Blacklotus UEFI persistence



1. Grubx64.efi will hook the winload.efi (OS windows loader) to install an infected driver

2. The driver injects some code in Winlogon process which contacts the C&C

Protection problem against Blacklotus => downgrade attack

- □ Baton Drop CVE fixed in **January 2022**, but vulnerable firmware **still usable in early 2024**

- □ Signatures of vulnerable EFI files revoked on **May 9, 2023** (DBX list updated) => https://uefi.org/revocationlistfile

- □ But the blocking of these signatures is not enabled by default.
  - ⇒ Still needed for **"legacy compatibility"**.
  - ⇒ Installing an older version of Windows is impossible because the firmware signature is blocked (problematic for restore points and backups).

- □ **July 11, 2023 :** Manual activation of signature blocking possible (hash in the DBX)

- □ **April 9, 2024 :** Add Microsoft Production PCA 2023 to the DB. (in order to revoke PCA2011)

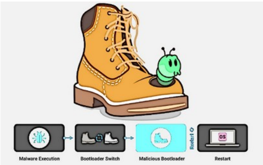
- □ **July 9, 2024 :** Windows security update includes mitigations but are not enable by default (Verify SVN)

- □ **+ 6 months ?? :** Blocking becomes mandatory (revoke PCA2011 in DBX) https://eclypsium.com/research/theres-a-hole-in-the-boot/

https://binarly.io/posts/The_Far_Reaching_Consequences_of_LogoFAIL/

This is not limited to Windows and Intel

BootHole on Grub Linux :



July 2020 : Buffer overflow vulnerability in the analysis of the GRUB2 configuration file (grub.cfg)

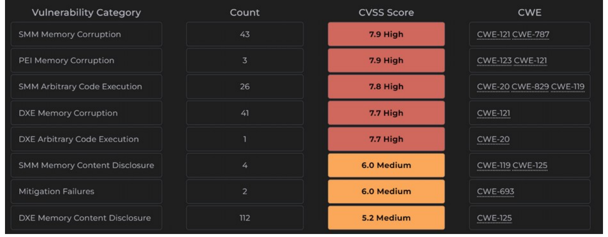
- Modification of the configuration file by an attacker with admin rights
- Works with secure boot
  no cryptographic verification of the config file
- All Linux distributions (Debian, Ubuntu, Mint, Red Hat, etc.) are affected

LogoFail on UEFI parser:

December 2023 Blackhat London: Vulnerabilities in the UEFI firmware image parser

- Modification of the splash image by an attacker with admin rights
- Works with secure boot
  no cryptographic verification of the config file
- Operates on x86/ARM and on PCs/servers from Lenovo, Dell, HP,…
  Demo presented at Blackhat using Lenovo ThinkCentre M70s Gen

https://binarly.io/posts/The_Untold_Story_of_the_BlackLotus_UEFI_Bootkit/index.html

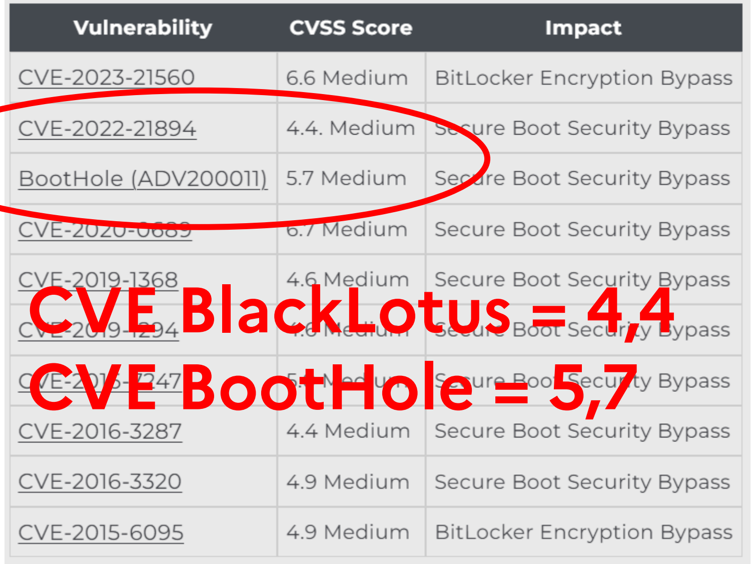


Vulnerability Category	Count	CVSS Score	CWE
SMM Memory Corruption	43	7.9 High	CWE-121 CWE-787
PEI Memory Corruption	3	7.9 High	CWE-123 CWE-121
SMM Arbitrary Code Execution	26	7.8 High	CWE-20 CWE-829 CWE-119
DXE Memory Corruption	41	7.7 High	CWE-121
DXE Arbitrary Code Execution	1	7.7 High	CWE-20
SMM Memory Content Disclosure	4	6.0 Medium	CWE-119 CWE-125
Mitigation Failures	2	6.0 Medium	CWE-693
DXE Memory Content Disclosure	112	5.2 Medium	CWE-125

Low consideration of the threat

- **Low CVSS scores, because**
  - ⇒ CVE tends to be quite complex without remote access.
  - ⇒ Need admin rights

However, the system impact is significant

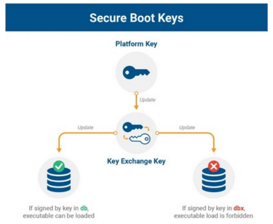


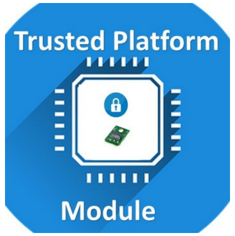
Vulnerability	CVSS Score	Impact
CVE-2023-21560	6.6 Medium	BitLocker Encryption Bypass
CVE-2022-21894	4.4. Medium	Secure Boot Security Bypass
BootHole (ADV200011)	5.7 Medium	Secure Boot Security Bypass
CVE-2020-0689	6.7 Medium	Secure Boot Security Bypass
CVE-2019-1368	4.6 Medium	Secure Boot Security Bypass
CVE-2019-1294	4.6 Medium	Secure Boot Security Bypass
CVE-2016-7247	5.[illegible] Medium	Secure Boot Security Bypass
CVE-2016-3287	4.4 Medium	Secure Boot Security Bypass
CVE-2016-3320	4.9 Medium	Secure Boot Security Bypass
CVE-2015-6095	4.9 Medium	BitLocker Encryption Bypass

CVE BlackLotus = 4,4
CVE BootHole = 5,7

- **Limited use of detection or protection functions on firmware**
  - ⇒ However, there are options: Secure boot, TPM, measured boot, DBX update, …

- **Few firmware updates**
  - ⇒ Despite numerous vulnerabilities identified (228 referenced by binarly.io)


Secure Boot Keys
Platform Key
Update
Update
Update
Key Exchange Key
If signed by key in db, executable can be loaded
If signed by key in dbx, executable load is forbidden

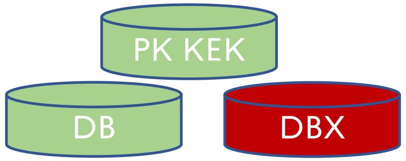


Best practices

Update firmware

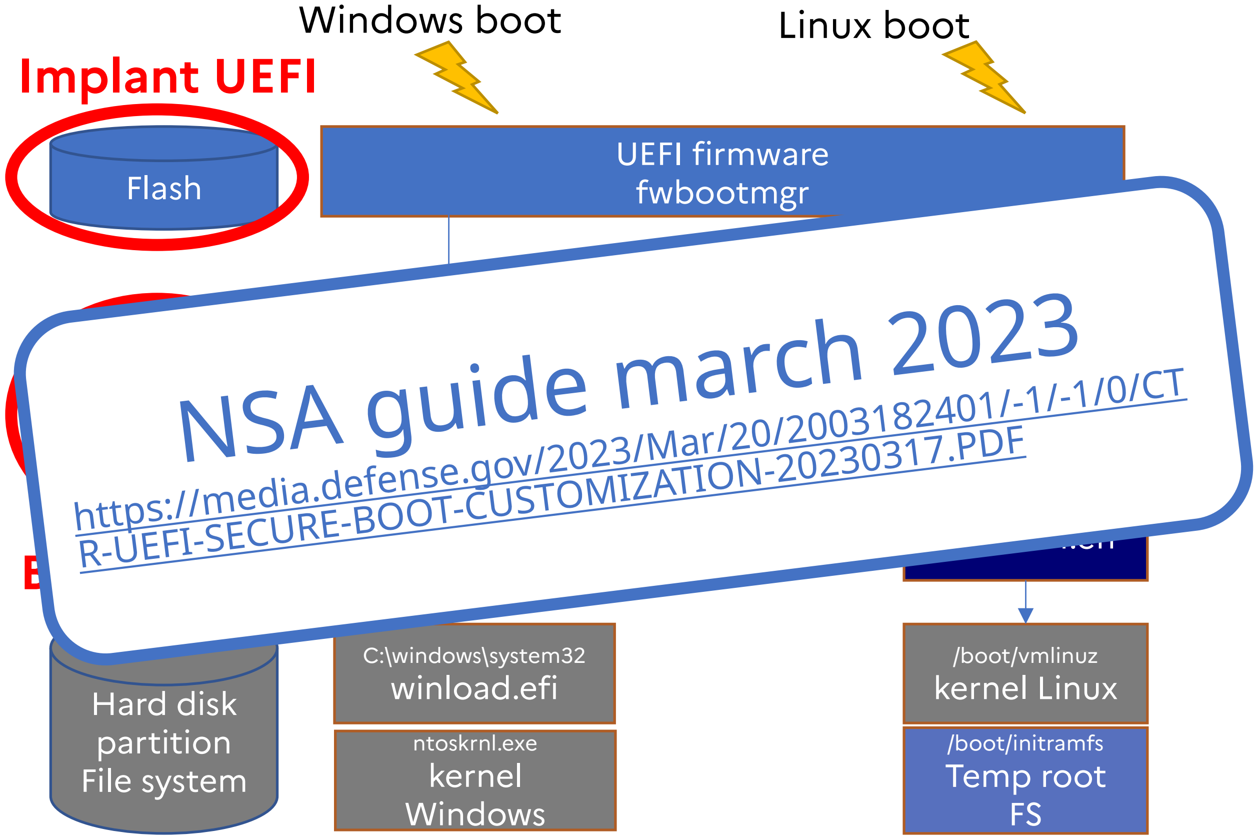
Use recent processors and their protection mechanisms

Enable secure boot
Update revocation list (DBX)
Use of one's own certificates (DB)



UEFI configuration verification tools
EFI partition detection tools
IoC management on EFI partition

Enable TPM
Enable Measured boot (attestation server)



NSA guide march 2023
https://media.defense.gov/2023/Mar/20/2003182401/-1/-1/0/CTR-UEFI-SECURE-BOOT-CUSTOMIZATION-20230317.PDF

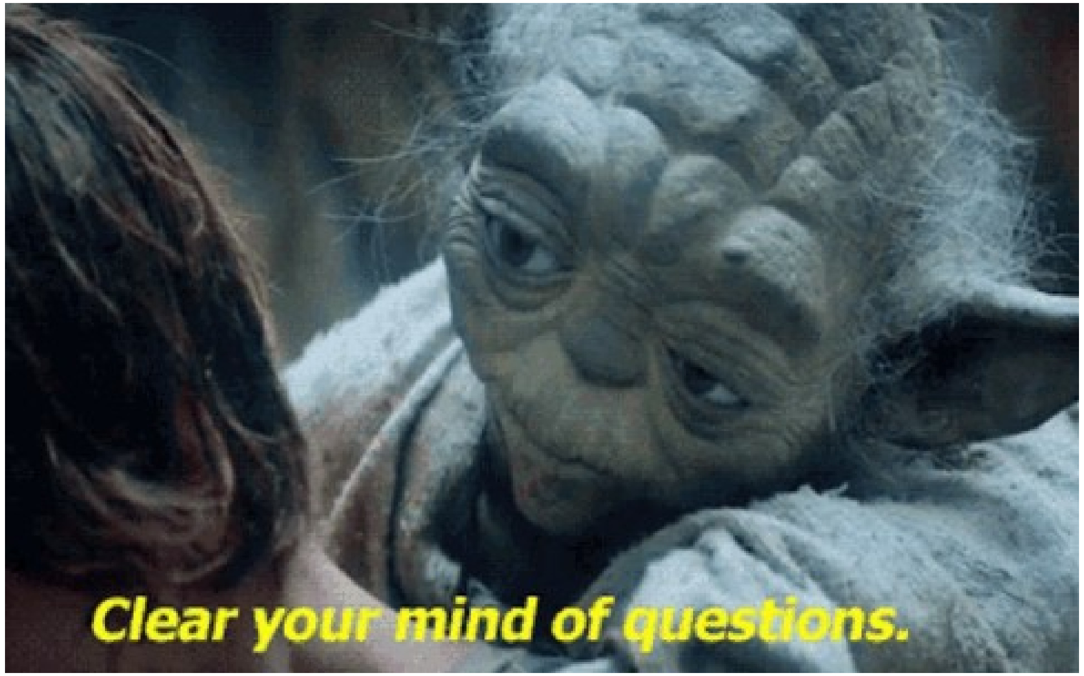
Conclusion

❑ **Threats** to the boot process are very real and must be taken into account in the **protection and detection** of our systems

❑ The protections presented are **imperfect and fallible**, but their implementation will **slow down the attacker**

❑ Some protection mechanisms are **complex and can be expensive** to implement, but let’s try to implement these protections in an **iterative manner**

THANKS, QUESTIONS ?



BONUS

Links

- Github EFI file to bypass HVCI :https://github.com/backengineering/VDM.git
- blog : Voyager – A Hyper-V Hacking Framework // Back Engineering Blog
- bootlicker : GitHub – realoriginal/bootlicker: A generic UEFI bootkit used to achieve initial usermode execution. It works with modifications.
- RUST UEFI :GitHub – rust-osdev/uefi-rs: Rust wrapper for UEFI.
- Visual-UEFI from Ionescu : GitHub – ionescu007/VisualUefi: A project for allowing EDK-II Development with Visual Studio => work with VS2022
- bootmgfw.efi with DMAbackdoor : s6_pcie_microblaze/python/payloads/DmaBackdoorBoot at master · Cr4sh/s6_pcie_microblaze · GitHub
- artemonsecrurity blog : https://artemonsecurity.blogspot.com

Driver classique :

- github VDM : GitHub – backengineering/VDM: Library to manipulate drivers that expose a physical memory read/write primitive., VDM – Vulnerable Driver Manipulation // Back Engineering Blog
- github MSREXEC : GitHub – backengineering/msrexec: Elevate arbitrary MSR writes to kernel execution.(SMEP, SMAP)

Links

Intro :

- https://www.malekal.com/bios-uefi-legacy-csm-gpt-mbr-dossier-complet/
- https://www.ssi.gouv.fr/uploads/IMG/pdf/uefi-pci-bootkits_sstic_article_fr.pdf

SRTM et DRTM (static et Dynamic measurment) :

- https://security.stackexchange.com/questions/39329/how-does-the-tpm-perform-integrity-measurements-on-a-system
- https://web.archive.org/web/20151028130757/http://tiw2013.cse.psu.edu/slides/tiw-2013-martin.pdf

Chipsec : https://github.com/chipsec/chipsec

Binarly : The Untold Story of the BlackLotus UEFI Bootkit | Binarly – AI -Powered Firmware Supply Chain Security Platform

Blacklotus and Lojax

- https://www.welivesecurity.com/2023/03/01/blacklotus-uefi-bootkit-myth-confirmed/
- https://www.welivesecurity.com/2018/09/27/lojax-first-uefi-rootkit-found-wild-courtesy-sednit-group/
- https://github.com/microsoft/secureboot_objects

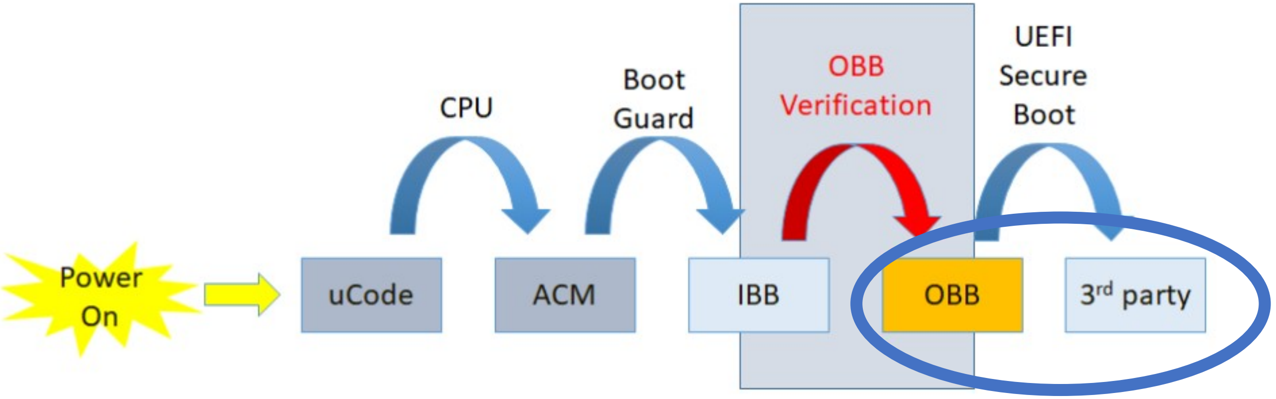
Guides :

- https://www.nsa.gov/portals/75/documents/what-we-do/cybersecurity/professional-resources/csi-boot-security-modes-and-recommendations.pdf
- https://uefi.org/sites/default/files/resources/Insyde%20HPE%20NSA%20and%20UEFI%20Secure%20Boot%20Guidelines_FINAL%20v2.pdf
- https://media.defense.gov/2023/Mar/20/2003182401/-1/-1/0/CTR-UEFI-SECURE-BOOT-CUSTOMIZATION-20230317.PDF

Intel Example

Intel introduced the **Intel® Boot Guard** Authenticated Code Module (**ACM**), which is a module signed by Intel.

- The **ACMs** modules assume responsibility to verify OEM (Original Equipment Manufacturer) platform firmware before the host CPU transfers control to OEM firmware. Because verifying the entire image is time-consuming, the ACM only verifies the initial boot block (**IBB**) code.
- The **IBB** is then responsible for verifying the OEM boot block (**OBB**).



https://github.com/tianocore/tianocore.github.io/wiki/OBB-Verification

MINISTÈRE
DES ARMÉES
ET DES ANCIENS
COMBATTANTS
*Liberté
Égalité
Fraternité*

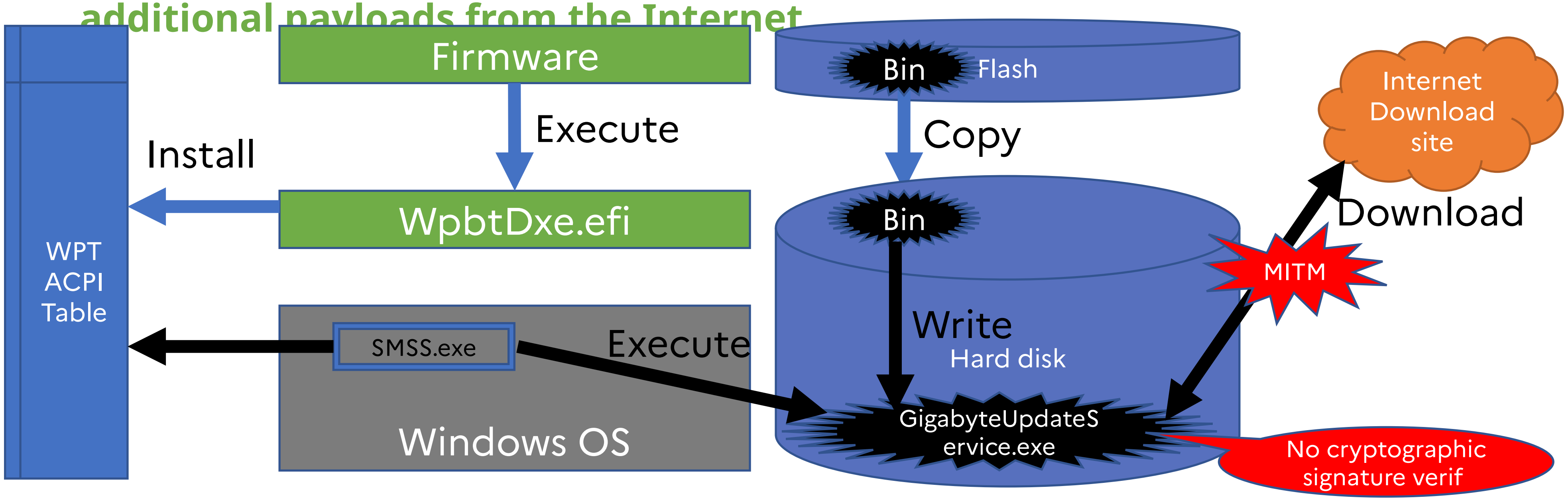
[Supply Chain Risk from Gigabyte App Center Backdoor - Eclypsium | Supply Chain Security for the Modern Enterprise](#)

GIGABYTE™

Other Real example : Gigabyte 2023

May 2023 : Eclypsium detected firmware on Gigabyte systems that **drops an executable Windows binary** => executed during the Windows startup process.

Problem : This executable binary **insecurely downloads and executes additional payloads from the Internet**



WPBT (Windows Platform Binary Table) is an ACPI table in your firmware allowing your computer vendor to run a program every time Windows (8 or later) boots.