Breaking The Chain Of Trust

Alex Bazhaniuk, Jesse Michael, and Mickey Shkatov

Agenda

Introduction

Architectural overview

Previous work

Architectural weaknesses

Vulnerabilities

Deep dive

Q&A

Whoami

@ABazhaniuk

@jessemichael

@HackingThings

Devices Are Complex With Complex Supply Chain

Huge and Hidden Attack Surface

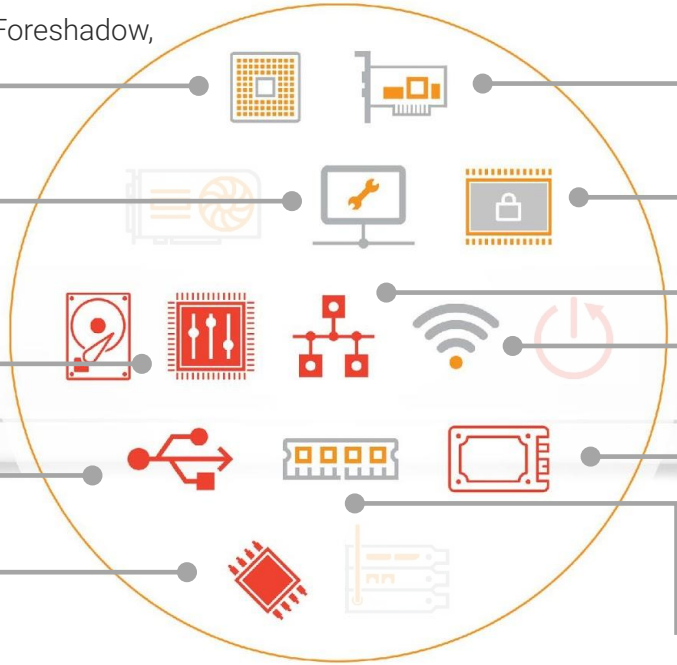
CPU: Spectre, Meltdown, Portsmash, Foreshadow, MDS, ZombieLoad, PlunderVolt

CSME: CVE-2017-5689, CVE-2018-3657, CVE 2017-5712

BMC: Cloudborne, CVE-2019-6260, iDRACula, CVE-2017-12542

USB: Bad USB, USBAnywhere

UEFI/SMM: Speedracer, S3 Bootscript Thunderstrike, Thinkpwn



PCI/Thunderbolt: DMA Attacks

TPM: ROCA, CVE-2018-6622, AMDFlaws, TPM-Fail

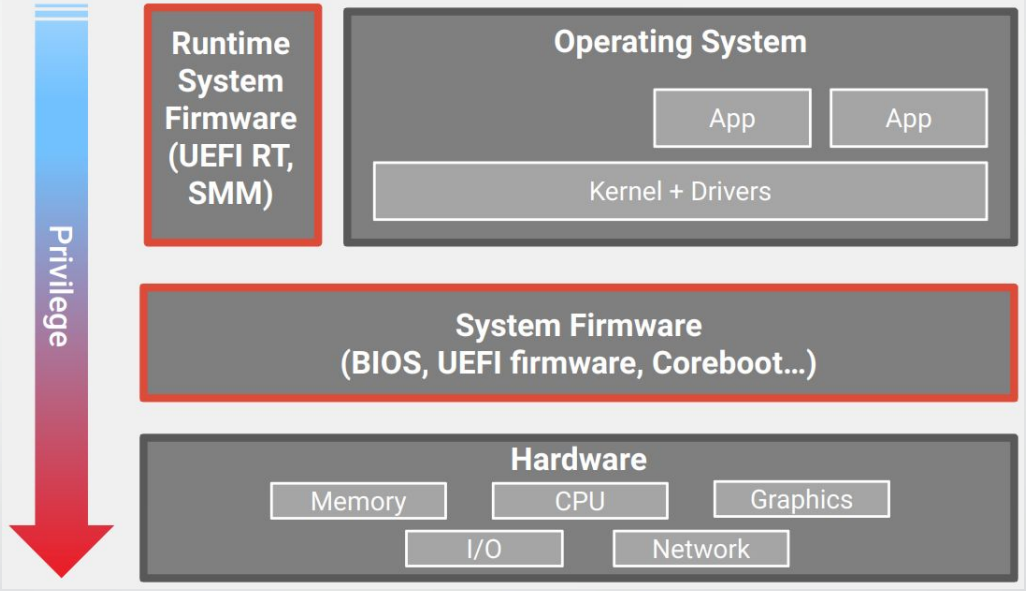
Network: Throwhammer, NetCAT

Wi-Fi: Broadpwn, CVE-2019-6496

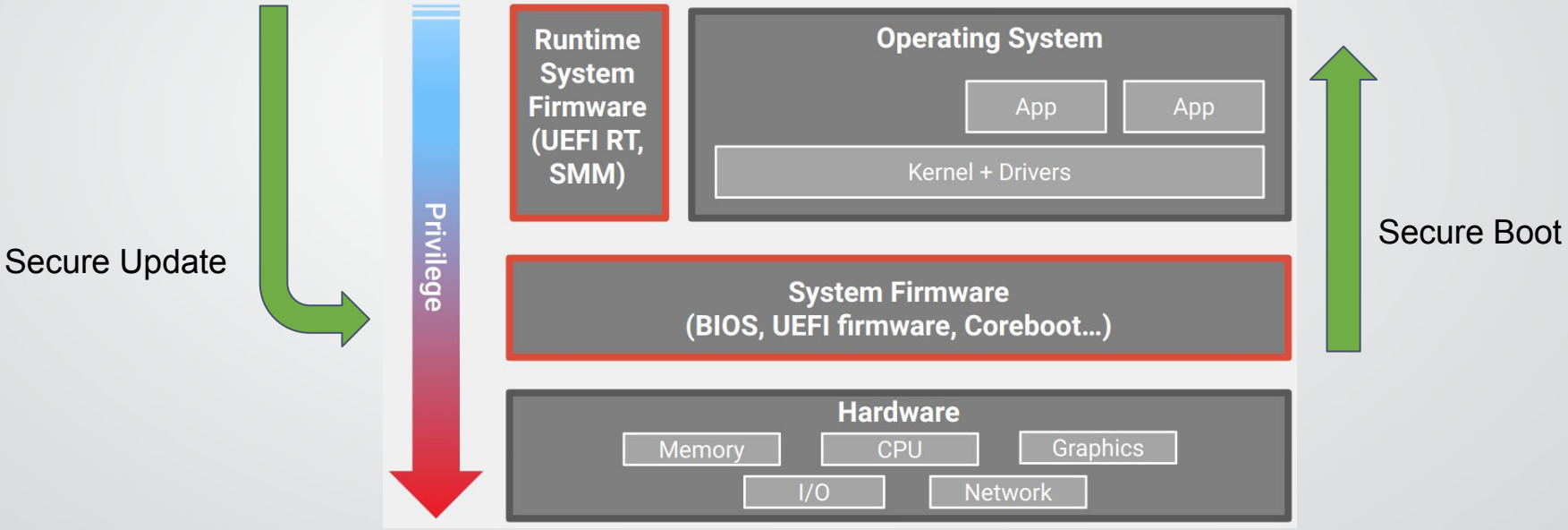
SSD: CVE-2019-10705, CVE-2019-11686, CVE-2018-12037 ...

DRAM: Rowhammer, RAMBleed

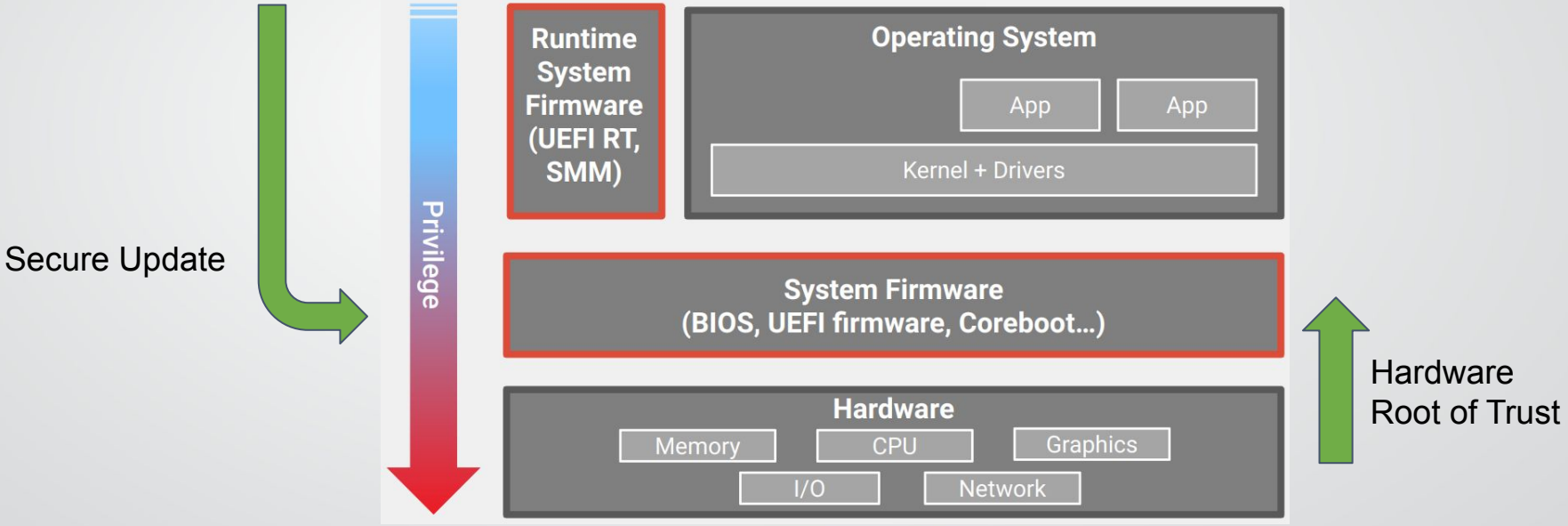
Overview: Architectural Privilege



Overview: SecureBoot & Secure Update



Overview: Hardware Root of Trust

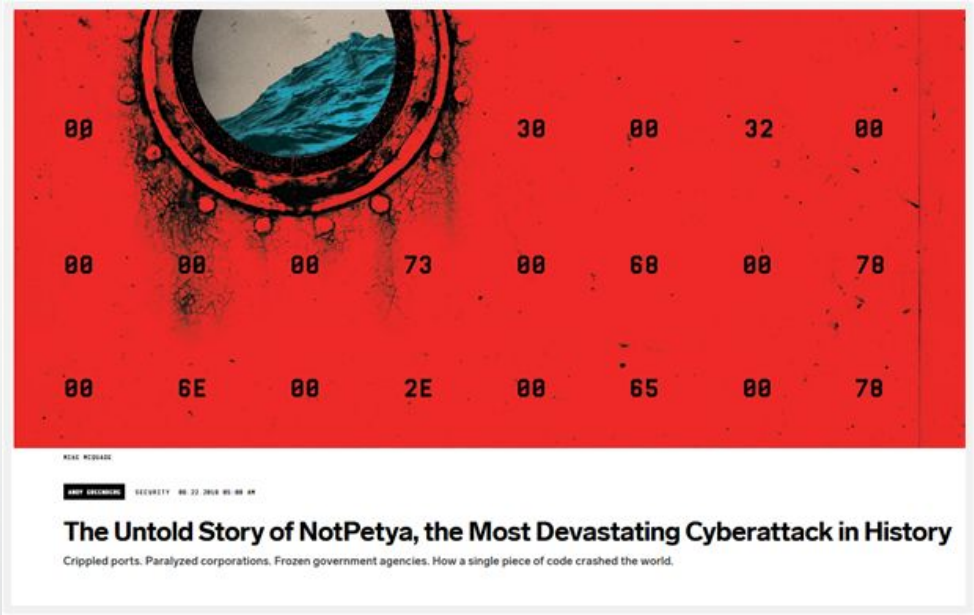


Overview: History of Bootloader Attacks

Threat groups use malware tampering with OS bootloaders

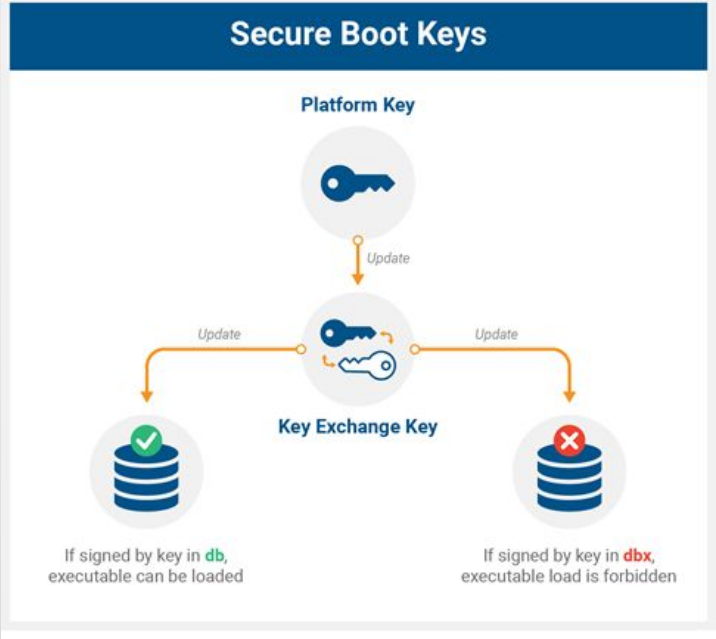
Legacy: Petya/NotPetya, APT41 Rockboot, LockBit, FIN1 Nemesis, MBR-ONI, Rovnix, etc.

UEFI: new EFILock ransomware using malicious EFI bootloaders



Overview: UEFI Secure Boot

Platform key hierarchy



In platform SPI:

- Persistent **Platform Key**
- Persistent **Key Exchange Key**
- **DB** and **DBX** Persistent Databases

In EFI System Partition:

- Microsoft-signed Bootloader

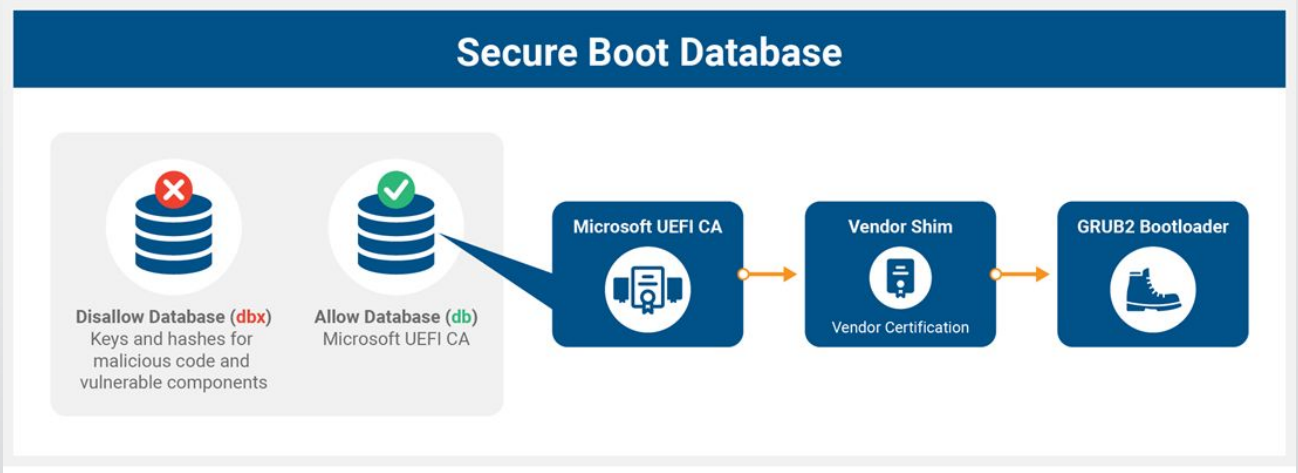
- or -

- Microsoft-signed SHIM
- Vendor-signed GRUB2 Bootloader

Overview: UEFI Secure Boot

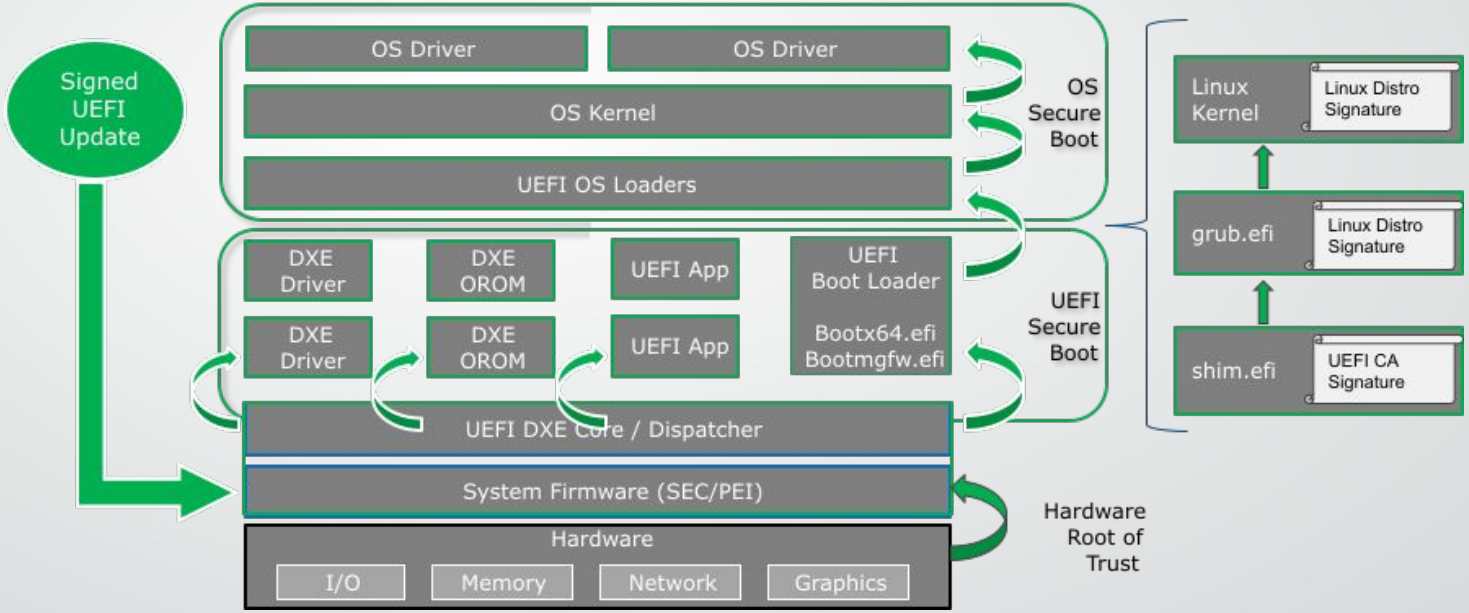
Secure Boot in Linux

- Using Microsoft CA's to securely boot Linux
- Shim signed by MS Third Party UEFI CA
- Bootloader signed by Linux distro



Overview: UEFI Secure Boot

Cryptographic signature verification



Overview: Hardware Root Of Trust

- TPM
  - Cryptographic processor to provide secure storage of “measurements”
  - Relies on other components to make enforcement decisions
  - Allows “sealing” secrets to specific measurements
- Intel BootGuard
  - Feature in modern Intel CPUs & chipsets
  - Verifies initial BIOS Boot Block before starting execution at reset vector
  - Can be configured to perform verified boot, measured boot, or both
- HP SureStart
  - Embedded Controller verifies BIOS Boot Block before powering on main CPU
- Google Titan
  - Custom chip initially built to protect Google Cloud
  - Part of Verified Boot process
  - Titan M and Titan C derivatives used in Google phones and chromebooks
- Apple T2
  - Custom SoC that handles platform operations
  - Provides secure enclave
  - Locks down the boot process

Previous work

Known Secure boot bypasses

- BCD Golden key
- Kaspersky Bootloader
- Incorrect protection of SPI contents or boot policy

Threats

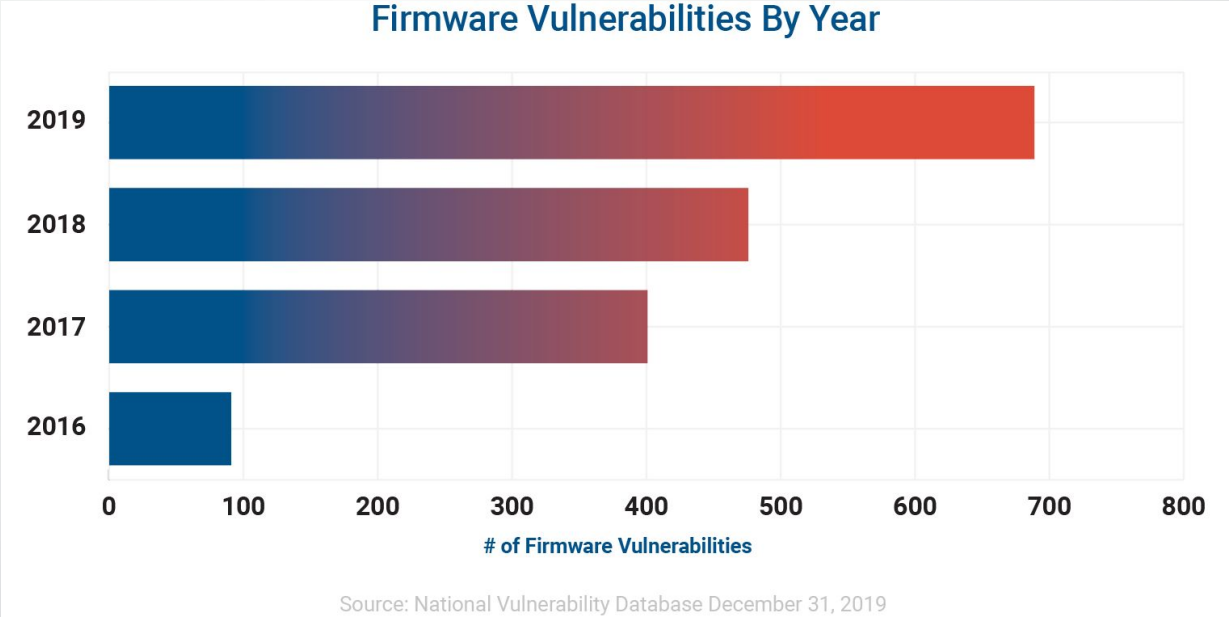
- Vulnerabilities in configuration and implementation
- Vulnerabilities in the bootloader
  - Vulnerable signed bootloader breaking root of trust
  - Unsigned component during boot flow, like ramfs

Previous work

Vulnerabilities in hardware root of trusts:

- TPM vulnerabilities
- Eclypsium DMA attacks against HP Sure Start
- Intel BootGuard bypasses
- Vulnerabilities in Titan and T2 chips.

7.5x Growth in Firmware Vulnerabilities



Architecture weaknesses

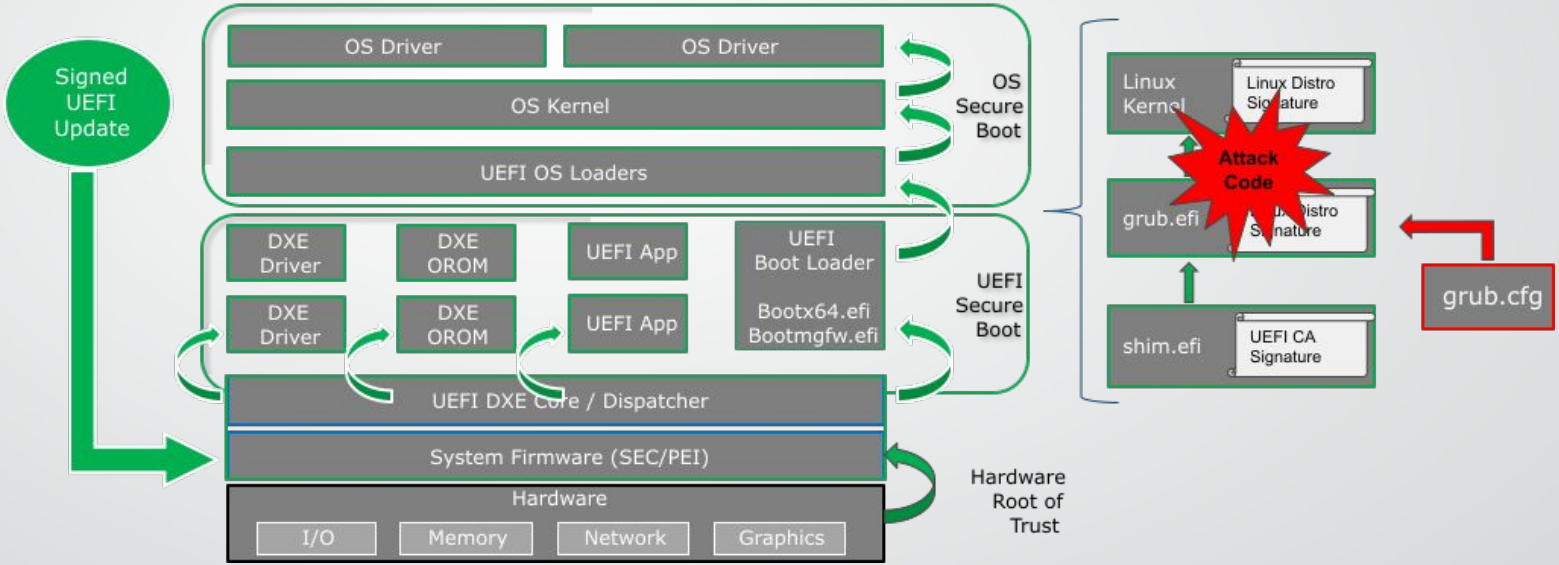
- Exploiting vulnerable signed bootloaders:
  - Signed old GRUB or shim (example: legacy Ubuntu GRUB)
  - Signed 3rd party signed EFI application or boot loaders (example: Kaspersky Rescue Disk EFI application).
  - Issues with blocking these binaries via DBX

Architecture weaknesses

- Evil maid attacks against Linux boot loader
  - Any of the bypasses in the late stage of boot will not be detected by hardware root of trust.
    - Bypass HP Sure Start, Intel BootGuard
  - Can booting into an internal UEFI shell be a Secure Boot bypass?

The “BootHole” Vulnerability

CVE-2020-10713: Buffer overflow in token parsing when reading grub.cfg



Deep dive: A closer look at CVE-2020-10713

Code generated by flex

```
#define YY_DO_BEFORE_ACTION \
    yyg->yytext_ptr = yy_bp; \
    yyleng = (int) (yy_cp - yy_bp); \
    yyg->yy_hold_char = *yy_cp; \
    *yy_cp = '\0'; \
    if ( yyleng >= YYLMAX ) \
        YY_FATAL_ERROR( "token too large, exceeds YYLMAX" ); \
    yy_flex_strncpy( yytext, yyg->yytext_ptr, yyleng + 1 , yyscanner); \
    yyg->yy_c_buf_p = yy_cp;
```

Macro definition in grub2 package

```
#define YY_FATAL_ERROR(msg)                         \
  do {                                              \
      grub_printf (_("fatal error: %s\n"), _(msg));     \
  } while (0)
```

Deep dive: A closer look at CVE-2020-10713

After the overflow…

```
struct grub_lexer_param *lexer = parser->lexerstate;

if (!lexer->record || !lexer->recording)
  return;

len = grub_strlen (str);
if (lexer->recordpos + len + 1 > lexer->recordlen)
  {
    ... code to realloc lexer->recording pointer with size of 2 * lexer->recordlen ...
  }
grub_strcpy (lexer->recording + lexer->recordpos, str);
lexer->recordpos += len;
```

Deep Dive: Additional Vulnerabilities

Additional Grub2 vulnerabilities found during mitigation process

- CVE-2020-14308: Integer overflow in grub_malloc
- CVE-2020-14309: Integer overflow in grub_squash_read_symlink
- CVE-2020-14310: Integer overflow in read_section_from_string
- CVE-2020-14311: Integer overflow in grub_ext2_read_link
- CVE-2020-15705: Would load unsigned kernels when grub2 used without shim
- CVE-2020-15706: Use-after-free in script handling
- CVE-2020-15707: Integer overflow in initrd size handling

Deep Dive: Additional Vulnerabilities (round 2)

Even more Grub2 vulnerabilities found and fixed 7 months later

- CVE-2020-14372: acpi command can be used to load crafted ACPI tables when Secure Boot is enabled
- CVE-2020-25632: Use-after-free in rmmod command
- CVE-2020-25647: Out-of-bound write in grub_usb_device_initialize()
- CVE-2020-27749: Stack buffer overflow in grub_parser_split_cmdline
- CVE-2020-27779: cutmem command can be used to manipulate memory map when Secure Boot is enabled
- CVE-2021-3418: GRUB 2.05 reintroduced CVE-2020-15705
- CVE-2021-20225: Heap out-of-bounds write in short form option parser
- CVE-2021-20233: Heap out-of-bound write due to mis-calculation of space required for quoting

Deep Dive: Additional Vulnerabilities

Many other potential vulnerabilities found and fixed, not just those assigned CVEs

- Integer overflow before allocation found throughout the codebase
- Potential to dereferencing past end of array in decompression code
- Multiple instances of use-after-free
- Multiple instances of double-free
- Memory leaks

New security features added to try to harden GRUB against future vulns

- Secure Boot Advanced Targeting (SBAT)
- Initial Stack Protector implementation

Systems affected by the vulnerability

Majority of Windows & Linux devices which have Secure Boot (2013+)

- Servers, Workstations
- End-user desktops, laptops
- Network appliances (Cisco, NetApp etc)
- Security appliances
- Special purpose devices: ATMs, POSs, industrial workstations, etc.
- Linux-based OT / IoT devices
- Cloud instances

Mitigation

- Updates to GRUB2 to address the vulnerabilities
- Linux distributions and other vendors using GRUB2 have published updates for their installers, bootloaders, and shims
- These new shims have been signed by the Microsoft 3rd Party UEFI CA
- Administrators of affected devices will need to update installed versions of operating systems in the field as well as installer images, including disaster recovery media.
- Eventually the UEFI revocation list (dbx) needs to be updated in the firmware of each affected system to prevent running this vulnerable code during boot.

What to look out for … revocation process

- UEFI-related updates have had a history of making devices unusable.
- If the revocation list (dbx) is updated before a given Linux bootloader and shim are updated, then the operating system will not load.
- Edge cases: dual-boot or deprovisioned machines
- The bootloader and OS need to be updated before the revocation is applied to the system.

What to look out for … disaster recovery

- Enterprise disaster recovery processes can run into issues where approved recovery media no longer boots on a system if dbx updates have been applied.
- In addition when a device swap is needed due to failing hardware, new systems of the same model may have already had dbx updates applied and will fail when attempting to boot previously-installed operating systems.
- Before dbx updates are pushed out to enterprise fleet systems, recovery and installation media must be updated and verified as well.

Process will be slow …

- Microsoft has released a set of signed dbx updates, which can be applied to systems to block shims that can be used to load the vulnerable versions of GRUB2.
- Due to the risk of bricking systems or otherwise breaking operational or recovery workflows, these dbx updates will initially be made available for interested parties to manually apply to their systems rather than pushing them out through Windows Update and applying them automatically.
- IT professionals should test the revocation updates on their individual systems and identify any issues before making the revocations mandatory.

Recommendations

- Start monitoring the contents of the bootloader partition (EFI system partition) now.
- Continue to install OS updates as usual across desktops, laptops, servers, and appliances. Preventing attackers from gaining administrative level access to your systems is critical.
- Test the revocation list update using the same firmware versions and models that are used in the field.

- To close this vulnerability, you need to deploy the revocation update. Make sure that all bootable media has received OS updates first, roll it out slowly to only a small number of devices at a time, and incorporate lessons learned from testing as part of this process.
- Engage with your third-party vendors to validate they are aware of, and are addressing, this issue.
- Eclypsium has powershell and bash scripts available which can be used to detect bootloaders that are being revoked by this dbxupdate.

Examining Bootloaders

- Stored in EFI System Partition
- Fallback bootloader is always \EFI\Boot\bootx64.efi
- Vendor-specific bootloaders are stored in EFI\<vendor> directories
- Mounted at /boot/efi/ by default in Linux
- Not mounted by default in Windows

root@wopr: /boot/efi

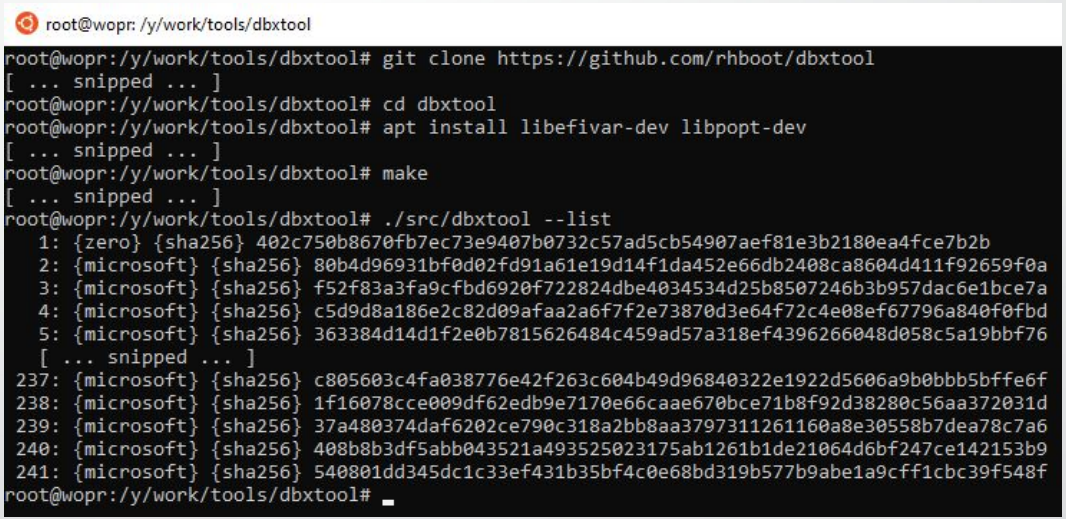
```
root@wopr:/boot/efi# find /boot/efi -iname '*.efi'
/boot/efi/EFI/ubuntu/grubx64.efi
/boot/efi/EFI/ubuntu/shimx64.efi
/boot/efi/EFI/ubuntu/mmx64.efi
/boot/efi/EFI/BOOT/BOOTX64.EFI
/boot/efi/EFI/BOOT/fbx64.efi
/boot/efi/EFI/BOOT/mmx64.efi
root@wopr:/boot/efi#
```

Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32> mountvol X: /S
PS C:\WINDOWS\system32> dir X:\EFI\Boot\


    Directory: X:\EFI\Boot


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         5/11/2021   8:10 PM        1558864 bootx64.efi


PS C:\WINDOWS\system32> dir X:\EFI\Microsoft\Boot\*.efi


    Directory: X:\EFI\Microsoft\Boot


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         5/11/2021   8:10 PM        1558864 bootmgfw.efi
-a----         5/11/2021   8:10 PM        1542992 bootmgr.efi
-a----         5/11/2021   8:10 PM        1349448 memtest.efi


PS C:\WINDOWS\system32>
```

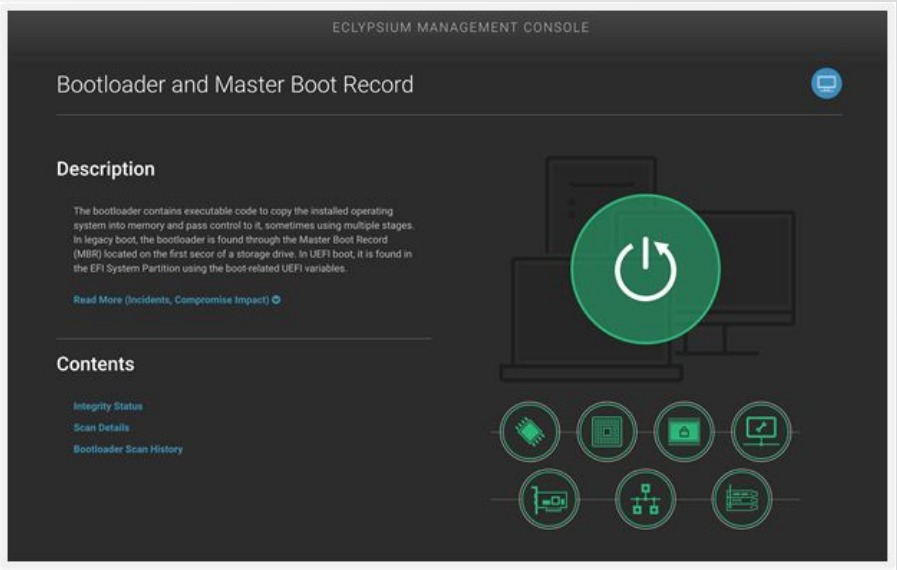
Examining DBX Revocation List

- dbxtool can be used to list and extract contents of dbx UEFI variable on Linux
- Can also operate on dbx file extracted on other system
- Can extract dbx contents on Windows using admin-level powershell
  - Get-SecureBootUefi -Name dbx -OutputFilePath dbx.bin



The right tools can make this process easier

- Gain visibility into bootloaders, EFI system partition, UEFI firmware, revocation database...
- Identify vulnerable bootloaders & missing updates
- Ensure latest stable OS/bootloader and revocation database updates are installed
- Detect malicious bootloaders & exploit code/data
- Across the fleet of devices (Linux, Windows)



Additional Resources

- Research Report:
  - https://eclypsium.com/2020/07/29/theres-a-hole-in-the-boot/
- BootHole open source scripts:
  - https://github.com/eclypsium/BootHole
- List of advisories:
  - https://github.com/eclypsium/BootHole/blob/master/ADVISORIES.md
- "Below the Surface” Threat Report:
  - https://eclypsium.com/firmware-threat-report

Q&A