Netanel Ben Simon & Meir Bloya

Boot Camp: A deep dive into windows boot security

About

1. Meir Bloya (https://il.linkedin.com/in/meir-blau)
2. Netanel Ben Simon (https://il.linkedin.com/in/netanel-ben-simon-419681226)
3. EPFS-IL Group - Edge and Platform Security Fundamentals - Israel
4. Pre OS - A specialized group within EPSF-IL, focus on low-level security research, for example: Boot, Firmware etc.

Boot Attacks: The Hidden Commonality of Cyber Elite

Equation Group

CIA

APT28

APT41

]HackingTeam[

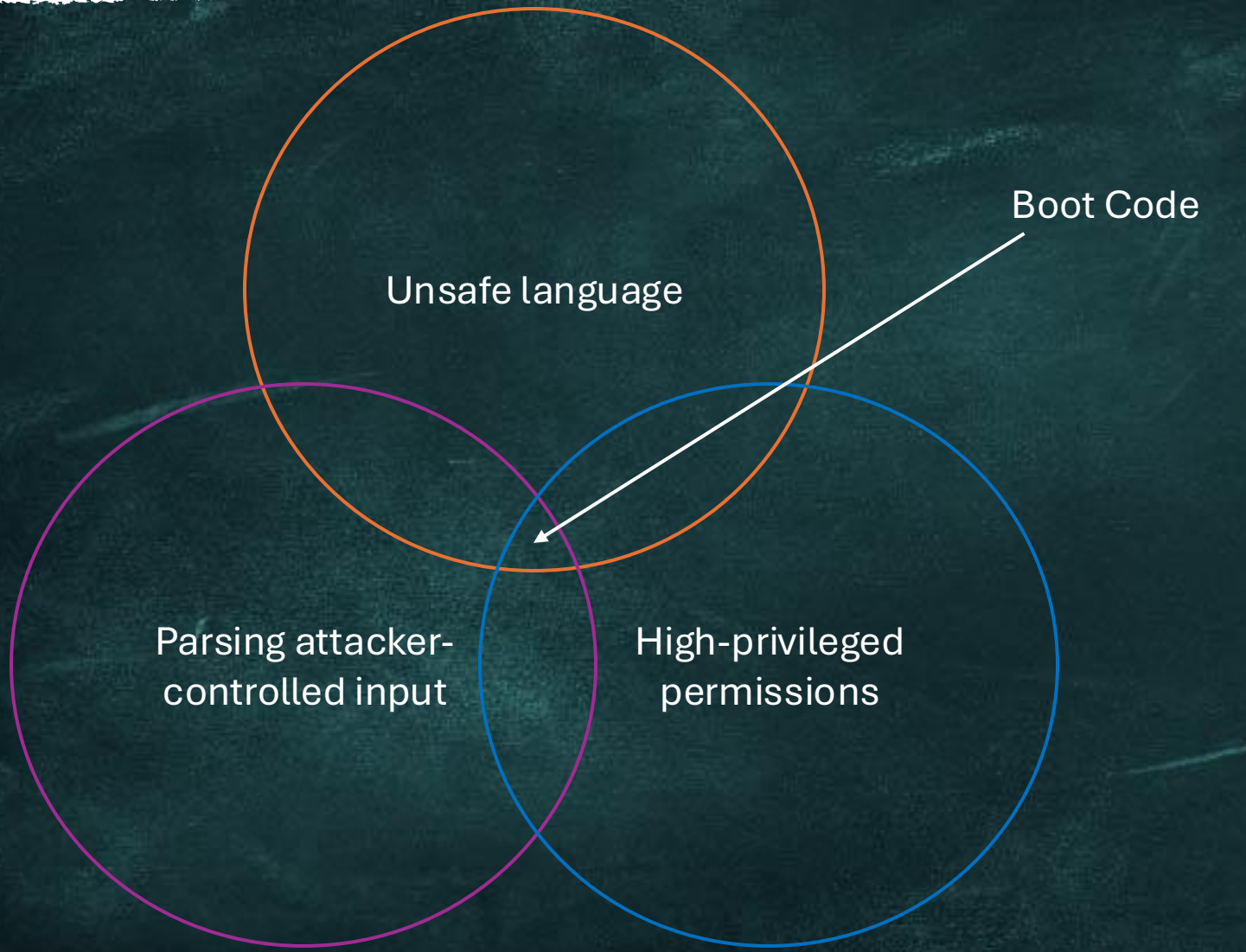
GAMMAGROUP

The Important Of Windows Secure Boot

Windows security

Windows Boot Security

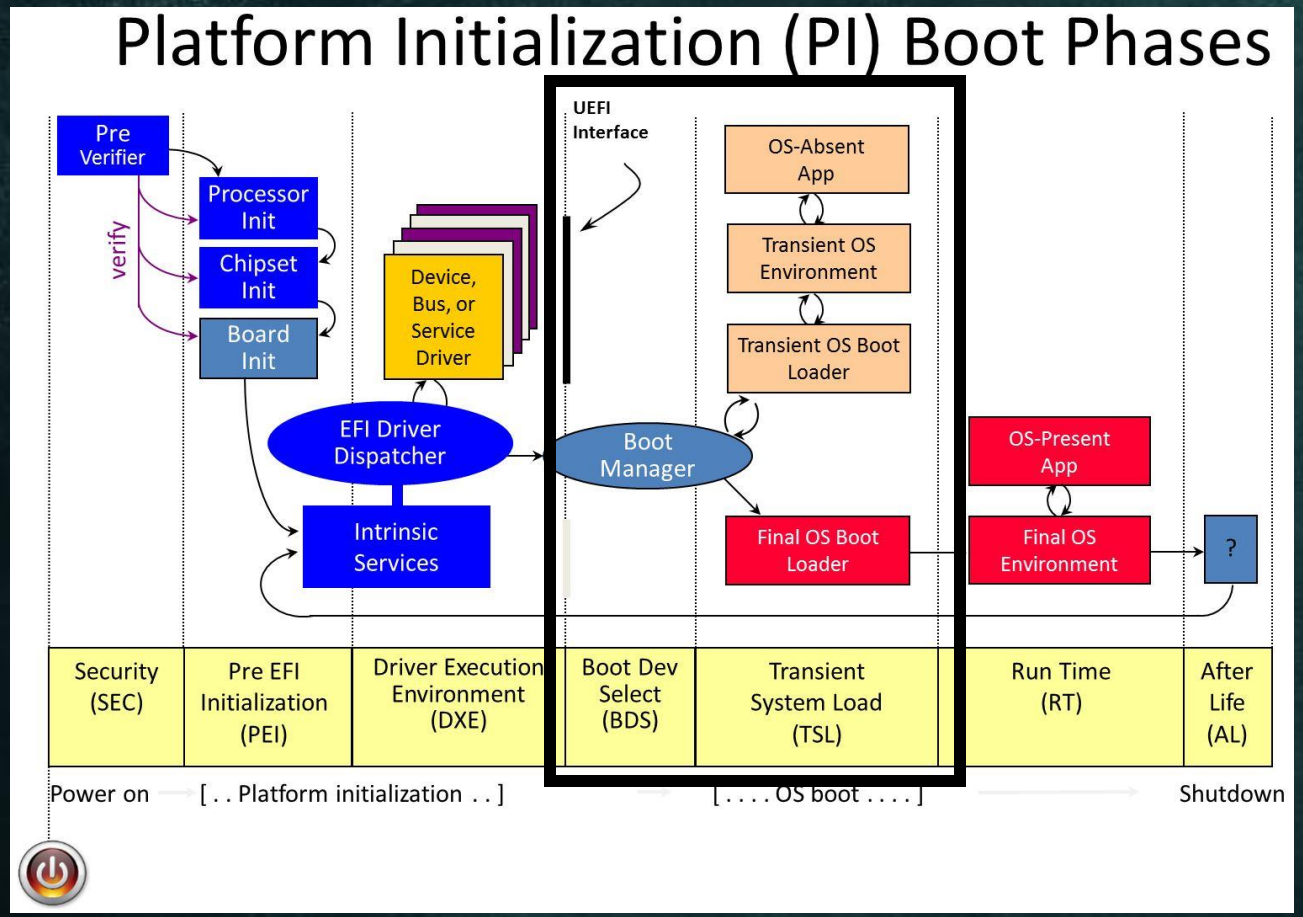
Rule Of 2


Unsafe language
Boot Code
Parsing attacker-controlled input
High-privileged permissions

Research Methodology

1. Manual Code Review
2. Fuzzing
3. Dynamic Debug

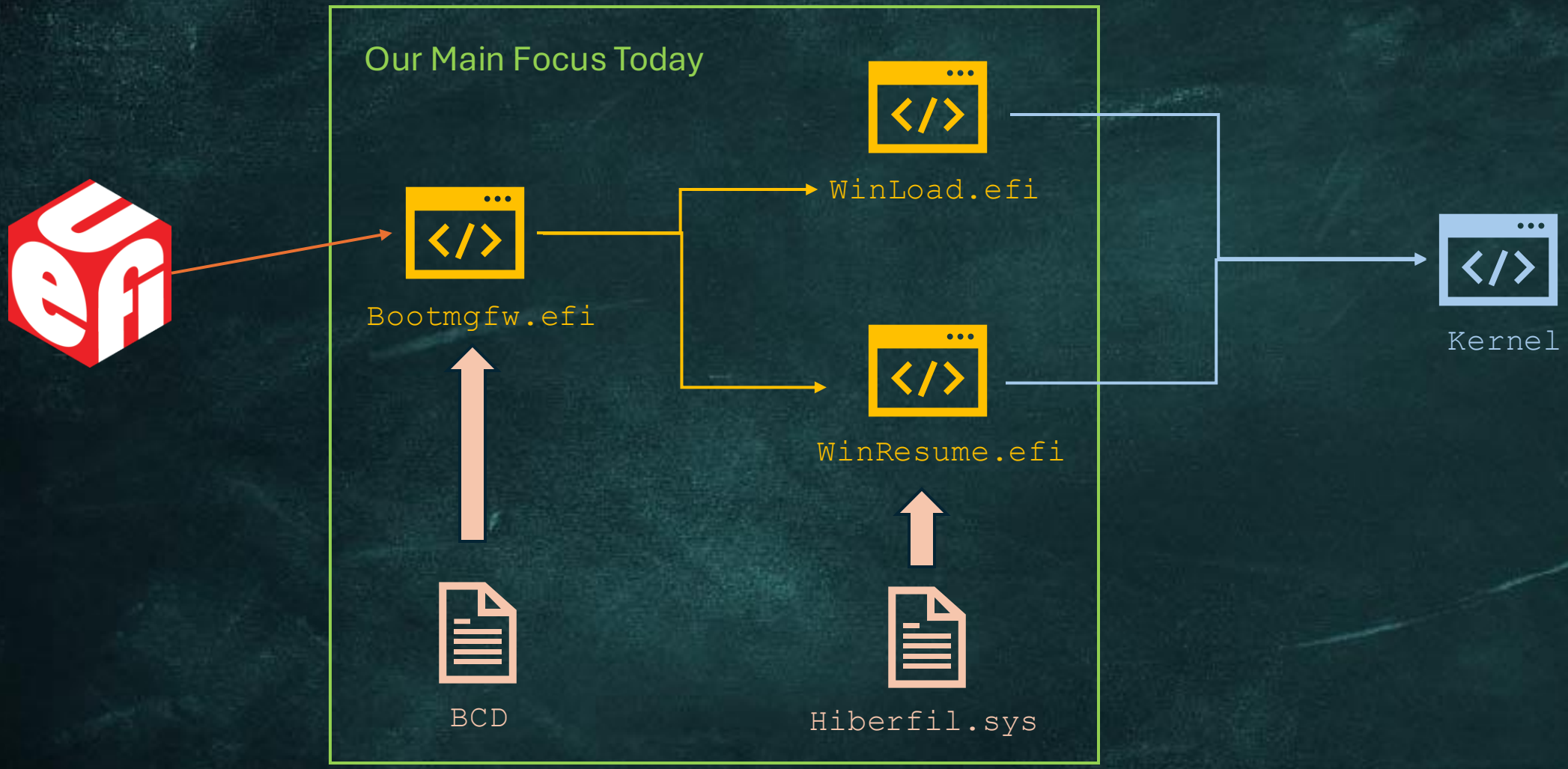
Boot: Behind the Scenes



1. BitLocker
2. Secure Boot
3. Measured Boot
4. VBS

https://tianocore-training.github.io/Lesson-0/

Boot overview – Simplified Boot Process



Attack surfaces

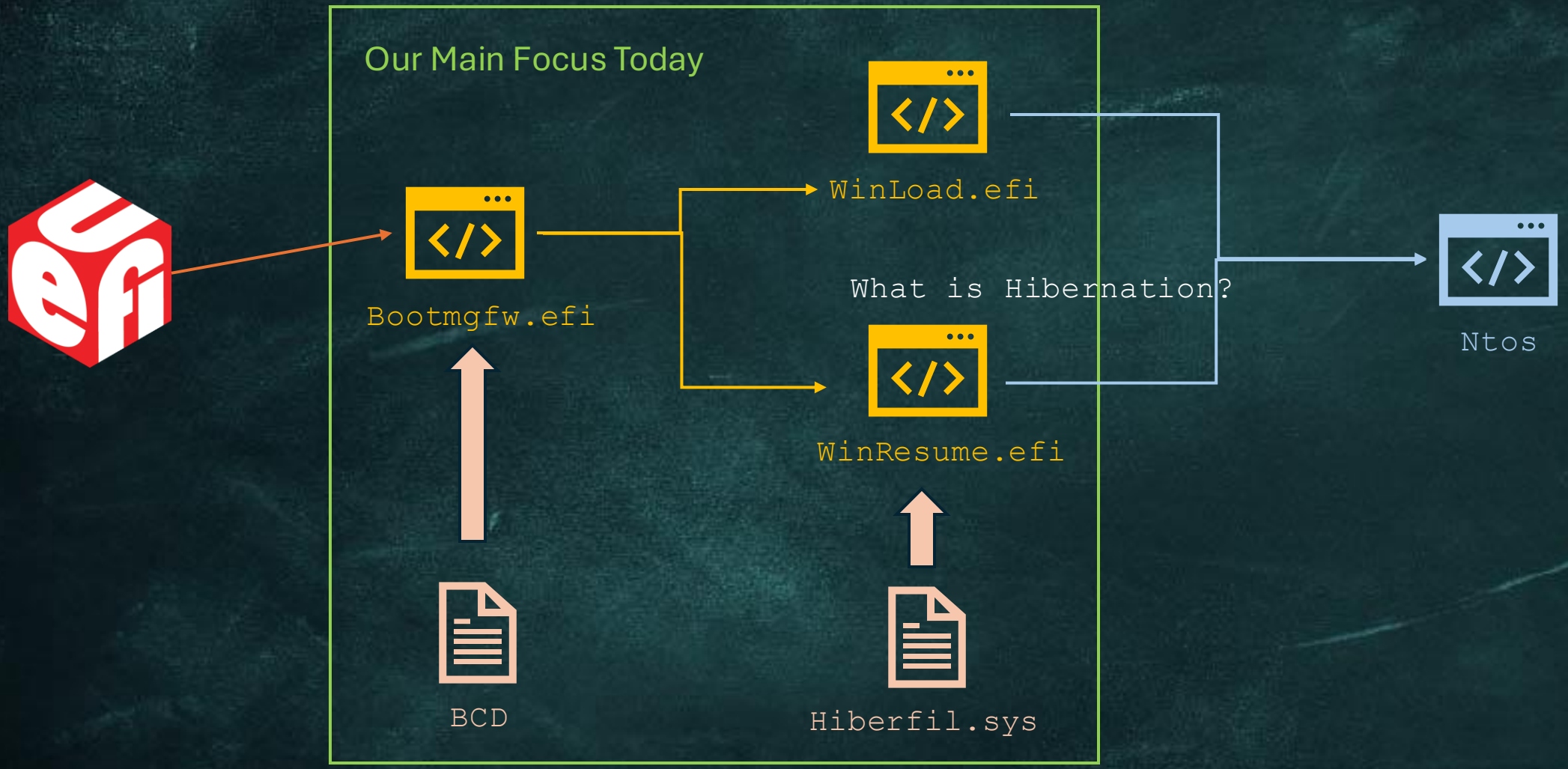
1. Admin to Boot
2. Physical access
3. Over the LAN

Admin To Boot - Why

Security Bypass

Stealth

Admin To Boot - How



Hibernation File Anatomy

```
C:\> dir /A
 ...
 Directory of C:\

...
03/02/2025  09:42 PM    27,363,241,984 hiberfil.sys
...
```

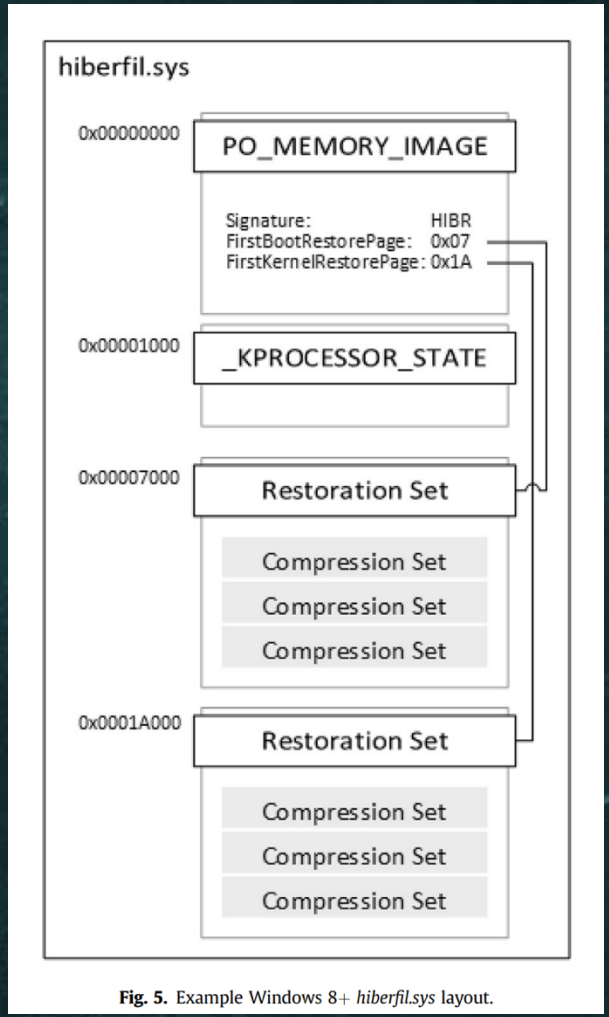
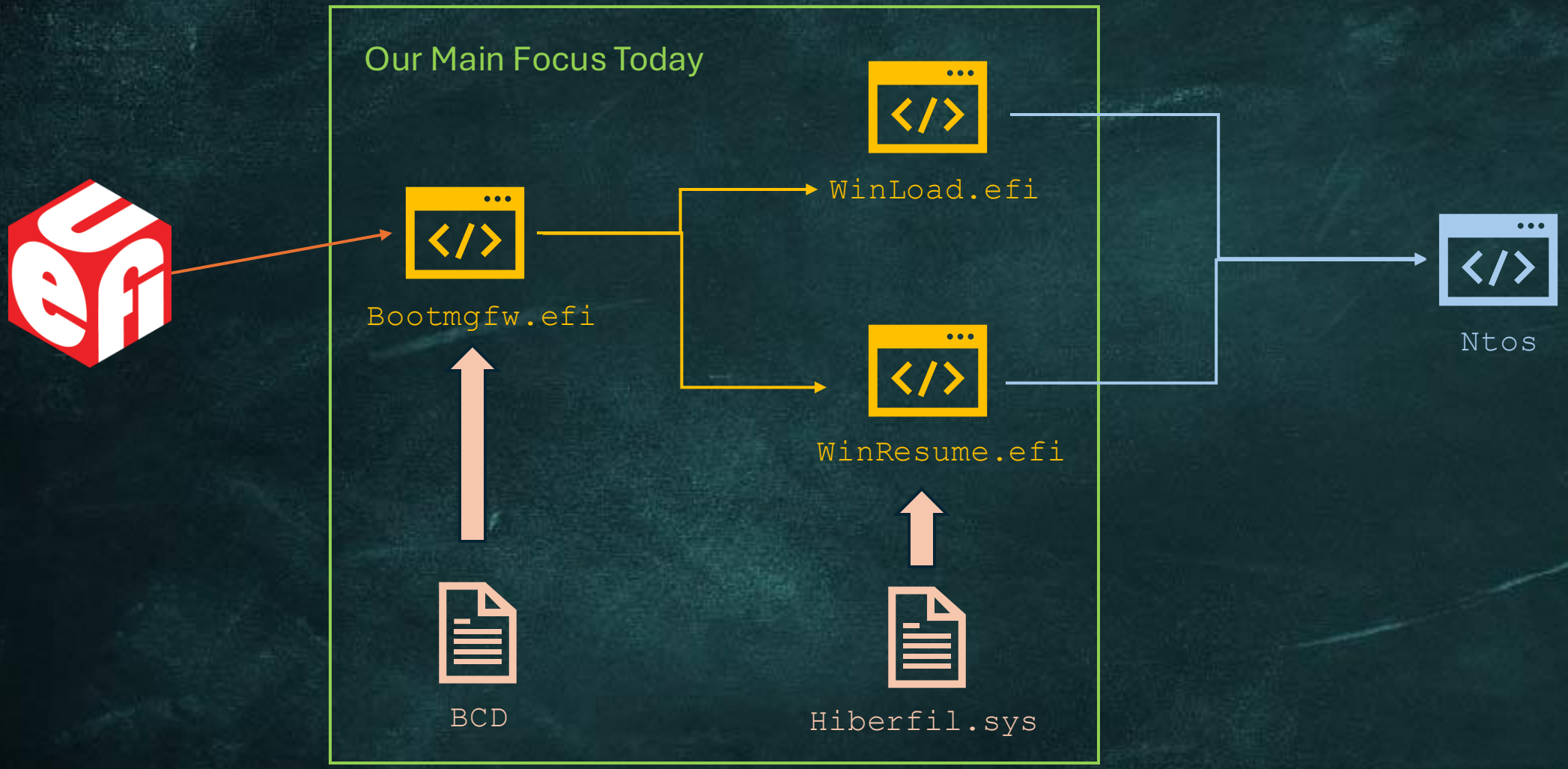


Fig. 5. Example Windows 8+ *hiberfil.sys* layout.

Source:

https://www.cct.lsu.edu/~golden/Papers/sylvehiber.pdf

Resume Process



Hibernation Bug - CVE-2024-37976

```
3     ...
4     // [NBS] TranslatedPage under 100% attacker control
5     HardwareAddress.QuadPart = (ULONGLONG)ImageHeader->TranslatedPage << PAGE_OFFSET;
6
7     // [NBS] Get the virtual address of our TranslatedPage HW address
8     Result = OsMapHardwareAddress(&RecoveryHeaderPtr,
9                                   HardwareAddress,
10                                  MEMORY_PAGE_SIZE,
11                                  MEMORY_FLAG_NONE);
12
13    ...
14
15    // [NBS] BOF - RecoveryHeaderPtr is 100% under attacker controlol
16    OsCopyMemory(RecoveryHeaderPtr, ImageHeader, sizeof(MEMORY_SNAPSHOT_HEADER));
17    ...
```

Attacker controlled

The controlled address used as src, we have Write *What *Where

Exploit - challgenges

Can not be edited

Hiberfil.sys

```
C:\>echo "rewrite?" >  hiberfil.sys
The process cannot access the file because it is being used by another process.
```

```
C:\>bcdedit /set {91c5aed0-f20b-11ef-8892-d8db38b9011f} filepath  \my\malicious\hiberfil.sys
```

Exploit - challgenges

~~Can not be edited~~

The kernel rewrite our custom file

Resume from hiberfil demand physical pushed button

Exploit - challgenges

~~Can not be edited~~

~~The kernel rewrite our custom file~~

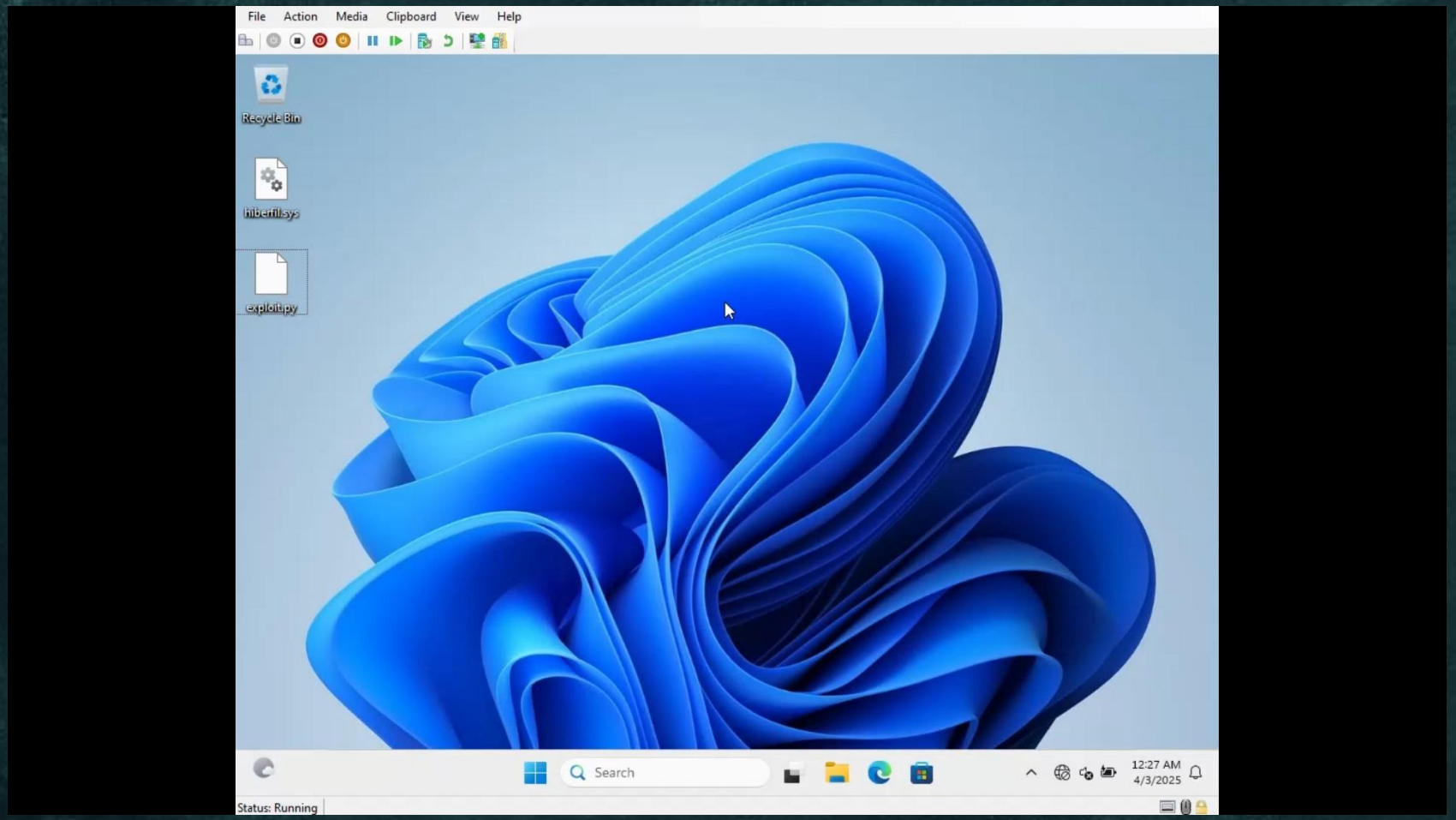
~~Resume from hiberfil demand physical pushed button~~

Hibernate from restart!

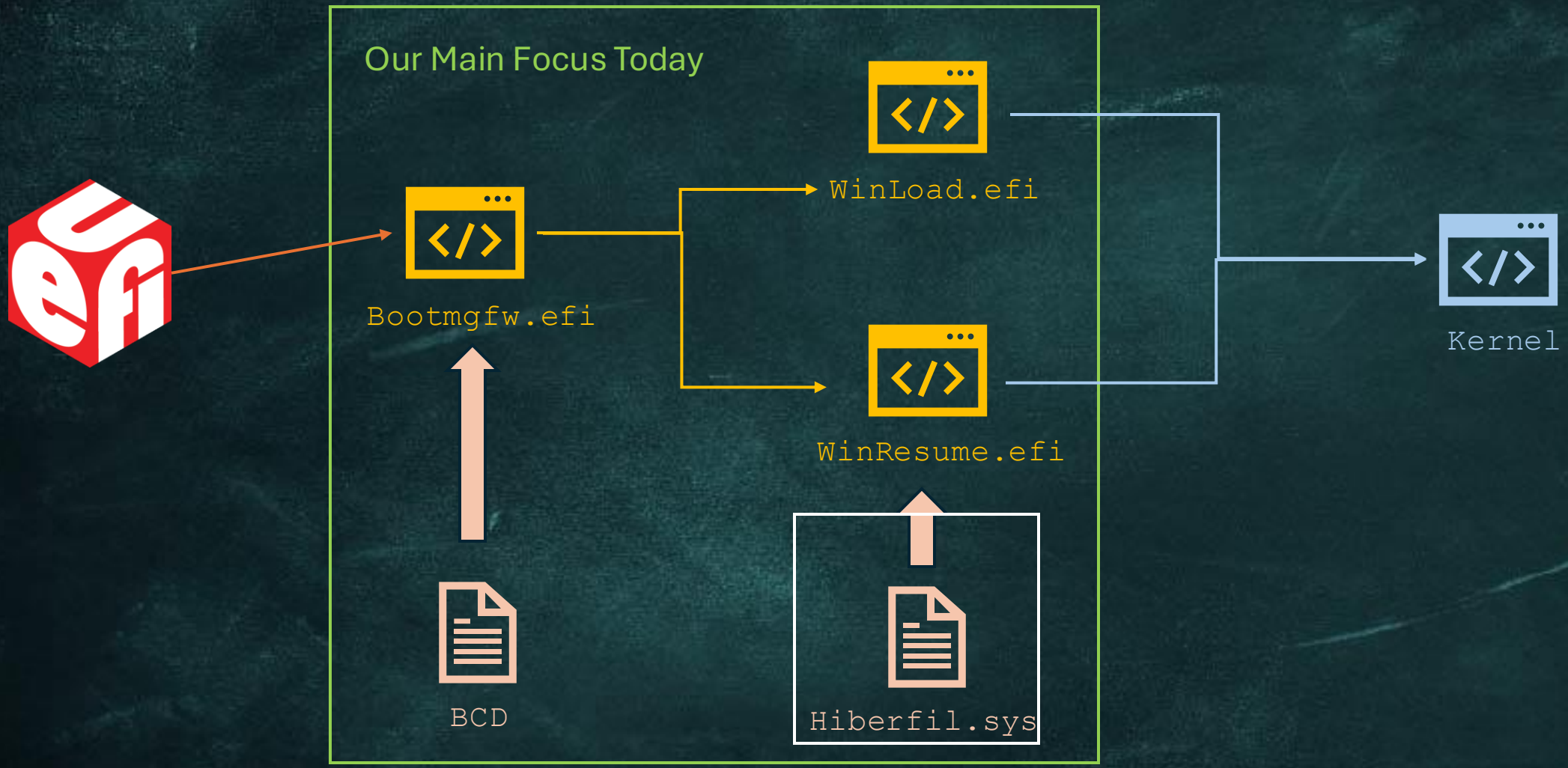
Replace Winload boot app with Winresume in the BCD

Point hiberfil to our controlled file

Demo


File
Action
Media
Clipboard
View
Help
Recycle Bin
hiberfil.sys
exploit.py
Search
12:27 AM
4/3/2025
Status: Running

Simplified Boot Process



Quick overview - BCD

Disk

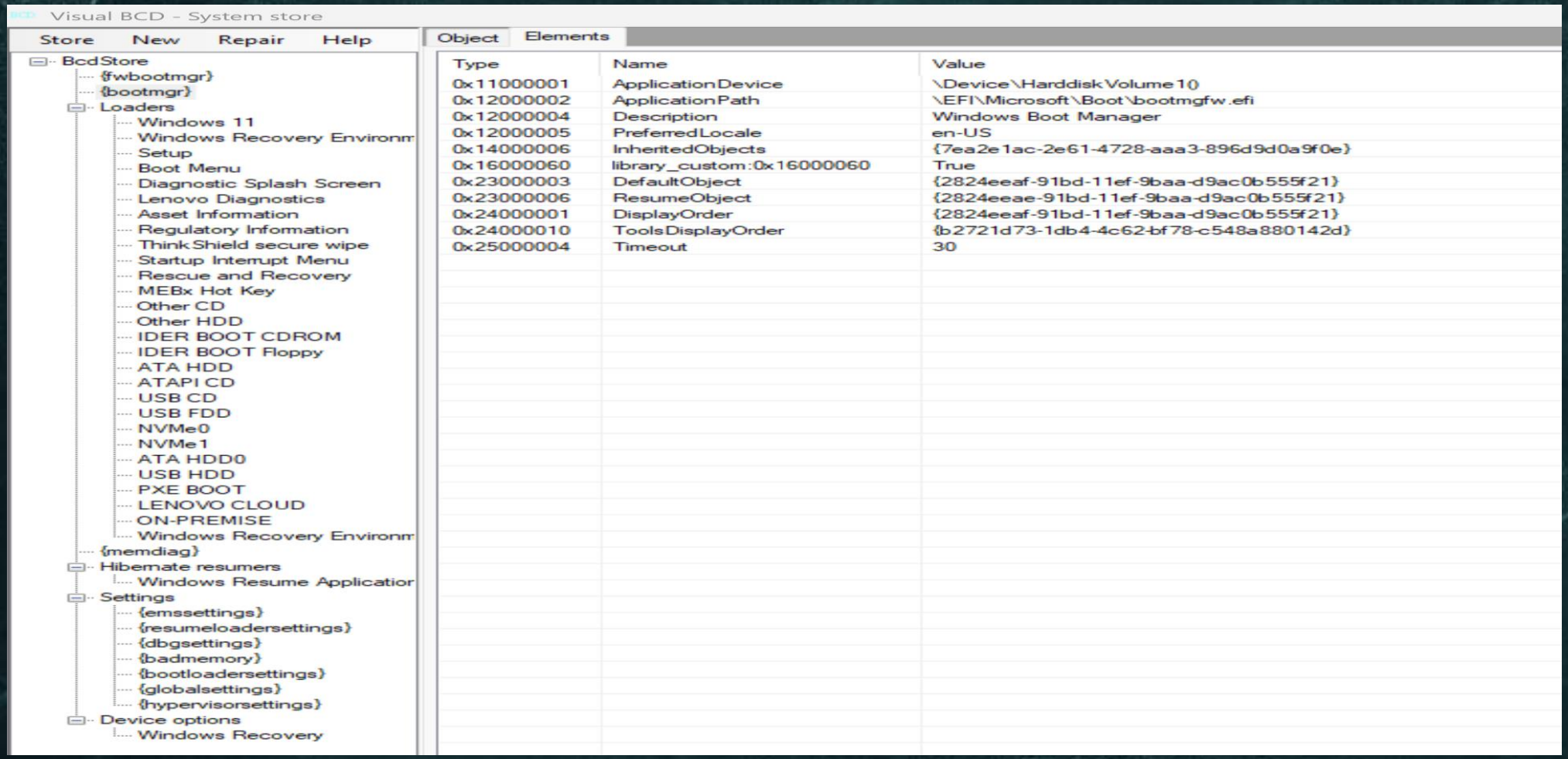
ESP

bootmgfw

BCD

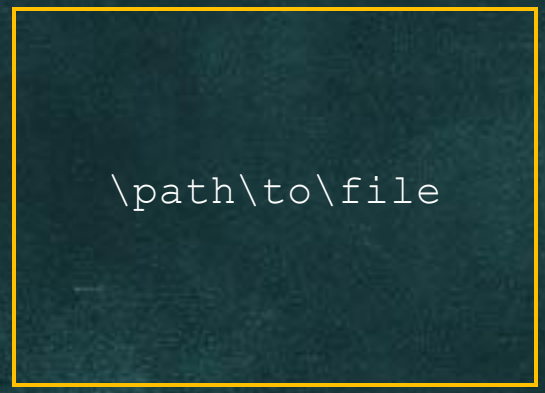
And More...

BCD Overview - Bug



BCD Bug Background - Locate Device

?

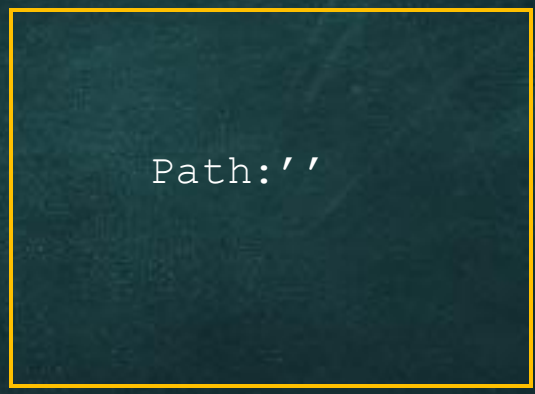



```
Bcdedit /set {current} device vhd=[LOCATE]\my.vhd
```

BCD Bug Background - Locate Device Object Types

String Type

Locate Type



BCD Bug Background - Locate Device Parent


```
Bcdedit /set {current} device vhd=[LOCATE]\my.vhd
```

Locate Device Struct

Memory

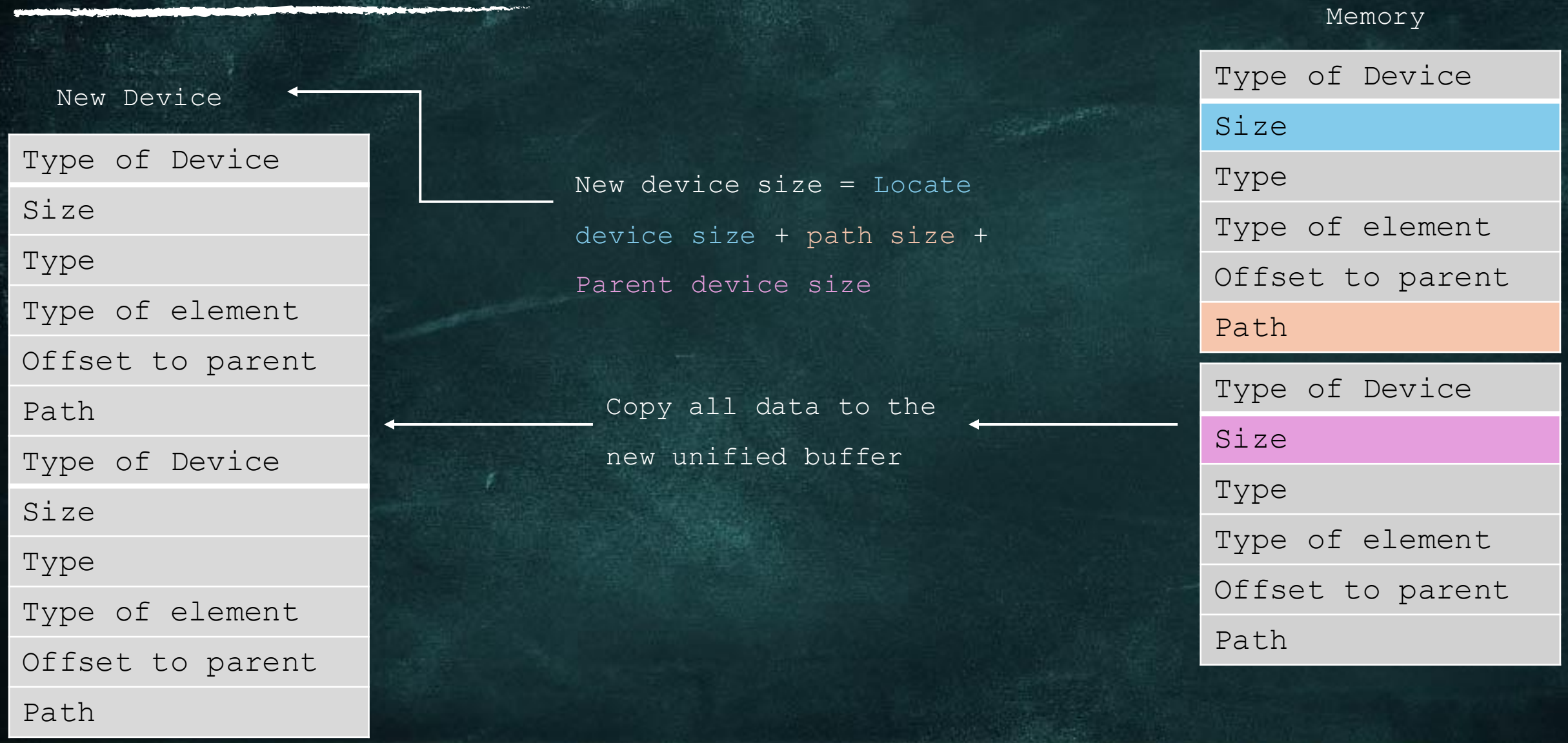
Locate Device

Type of Device
Size
Type
Type of element
Offset to parent
Path

Base address + offset →

Type of Device
Size
Type
Type of element
Offset to parent
Path

Bug Illustration



BCD Bug Example - CVE-2024-28923

```
9       POffset += PathLength;
10      devSize = POffset;
11
12      // [MB] Attacker have control over ldParent, the src of it is on non anitized BCD locate device element
13      if (ldParent != NULL) {
14          // [MB] Integer overflow, ldParent->Size is fully attacker-controlled.
15          // No validation performed on this value.
16          devSize += ldParent->Size;    -> Integer wraparound
17      }
18
19      // [MB] Allocation used the overflowed size
20      unifiedDev = BlMmAllocateHeap(devSize);   -> Using the wraparound data
21      ...
22      // [MB] Buffer overflow accure here, we write to unifiedDev based on ldParent->Size size
23      // This size is fully controlled by attacker  Using attacker-controlled size
24      RtlCopyMemory(Add2Ptr(unifiedDev, POffset), ldParent,ldParent->Size);
```

Attack surfaces

1. Admin to Boot
2. *Physical access*
3. Over the LAN

Physical Access

Disk

User Data

EFI System Partition (ESP)

bootmgfw

BCD

And More…

Attack surfaces

1. Admin to Boot
2. Physical access
3. Over the LAN

Over The LAN

From Manual VR to Automated VR - Fuzzing

Big and active code base

Complex code base (multiple different attack surfaces)

Fuzzing Boot Complexities

Difficult to harness

Requires special environment

Complex flows and states

Complex structures in different attack surfaces

Fuzzing Boot Complexities

~~Difficult~~ Easy to harness

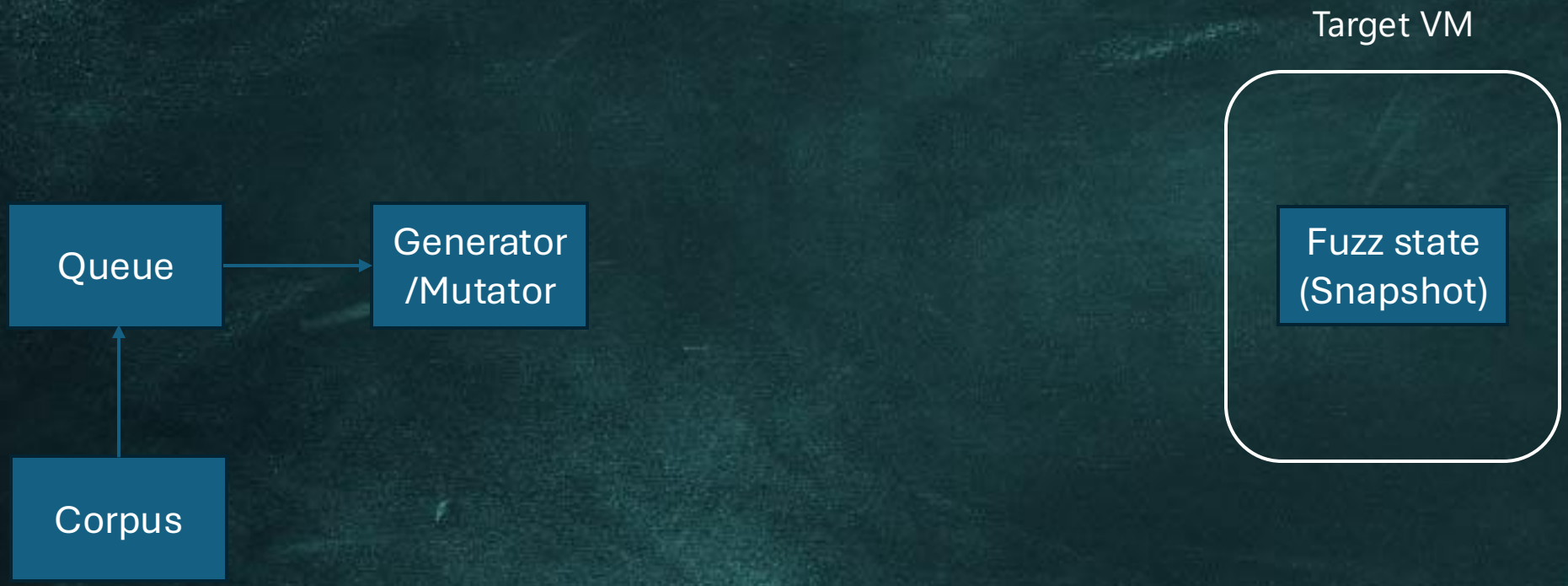
Requires special environment

Complex flows and states

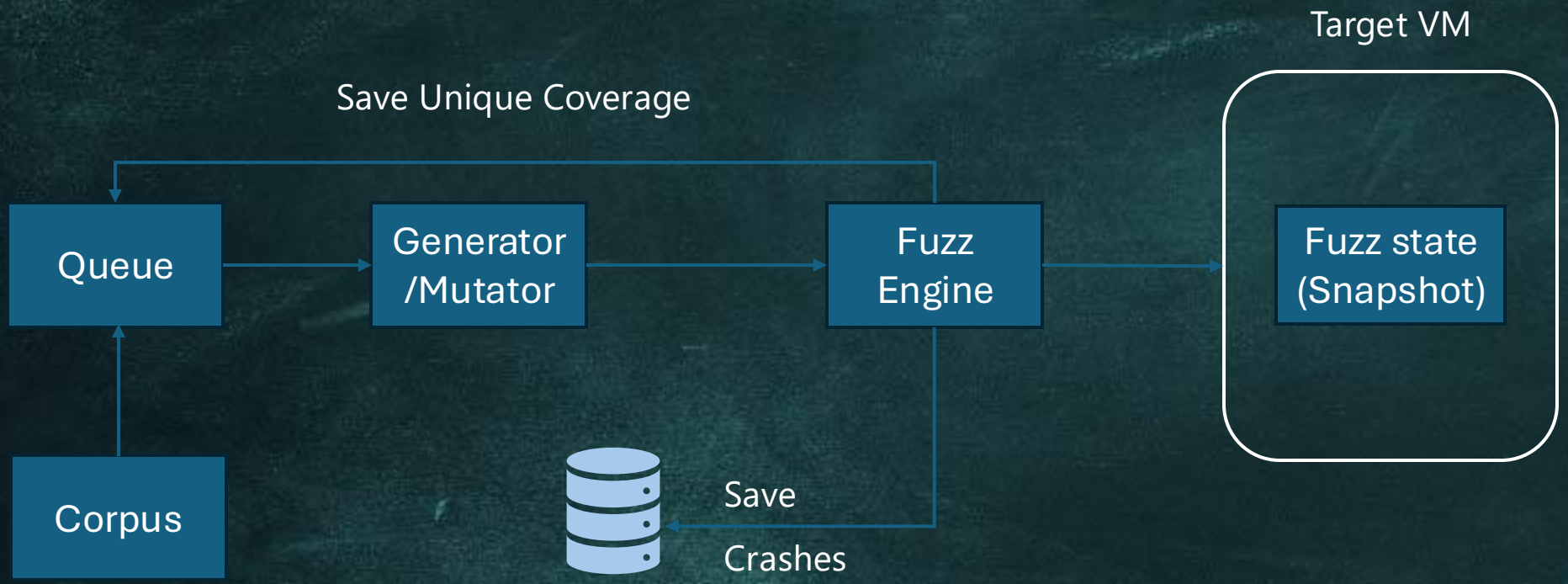
Complex structures in different attack surfaces

Example of Snapshot fuzzer

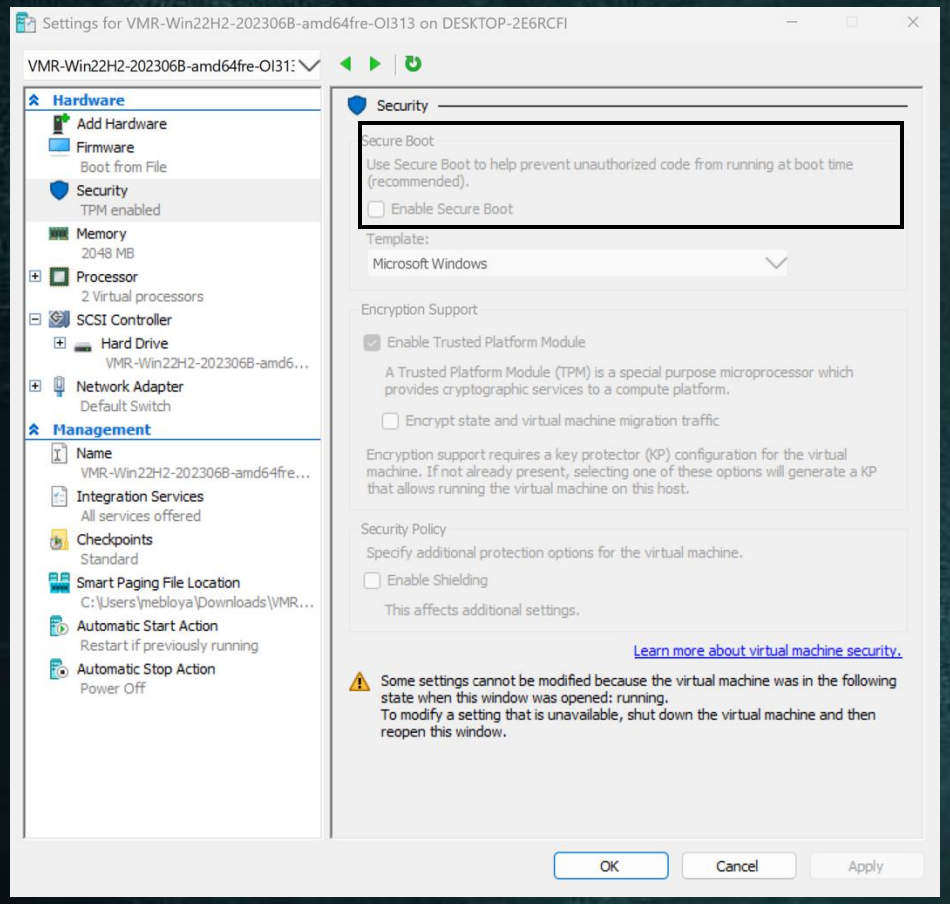
Example of Snapshot fuzzer



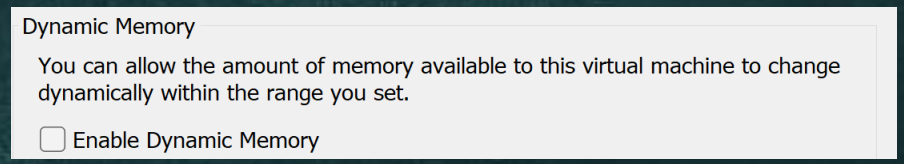
Example of Snapshot fuzzer



Dynamic Debug Setup – Disable Secure Boot On Guest



Disable VM Dynamic memory



Dynamic Debug Setup - Setting up Debugger & Hibernate

```powershell
param($VmName = "VM-Name", $Enable = $true)

# Stop VM, suppress warnings
Get-VM -Name $VmName | Stop-VM -Force -WarningAction SilentlyContinue

# Get the WMI objects
$wmiCS = gwmi -namespace root\virtualization\v2 -query "select * from Msvm_ComputerSystem where ElementName='$VmName'"
$vsSettings = $wmiCS.GetRelated("Msvm_VirtualSystemSettingData","Msvm_SettingsDefineState",$null,$null,"SettingData","ManagedElement",$false,$null)
Write-Host "Before: EnableHibernation = $($vsSettings.EnableHibernation)"

# Modify the hibernation setting
[xml]$xml = $vsSettings.GetText(1)
$xml.SelectNodes("/INSTANCE/PROPERTY[@NAME='EnableHibernation']")[0].VALUE = $Enable.ToString().ToLower()

# Apply changes
$mgmtSvc = gwmi -Namespace "root\virtualization\v2" -Class Msvm_VirtualSystemManagementService
$mgmtSvc.ModifySystemSettings($xml.OuterXml) | Out-Null

# Verify the change
$vsSettings = $wmiCS.GetRelated("Msvm_VirtualSystemSettingData","Msvm_SettingsDefineState",$null,$null,"SettingData","ManagedElement",$false,$null)
Write-Host "After: EnableHibernation = $($vsSettings.EnableHibernation)"
```

Src: https://superuser.com/questions/1515527/how-to-enable-sleep-hibernate-on-a-hyper-v-guest-vm/1515838#1515838

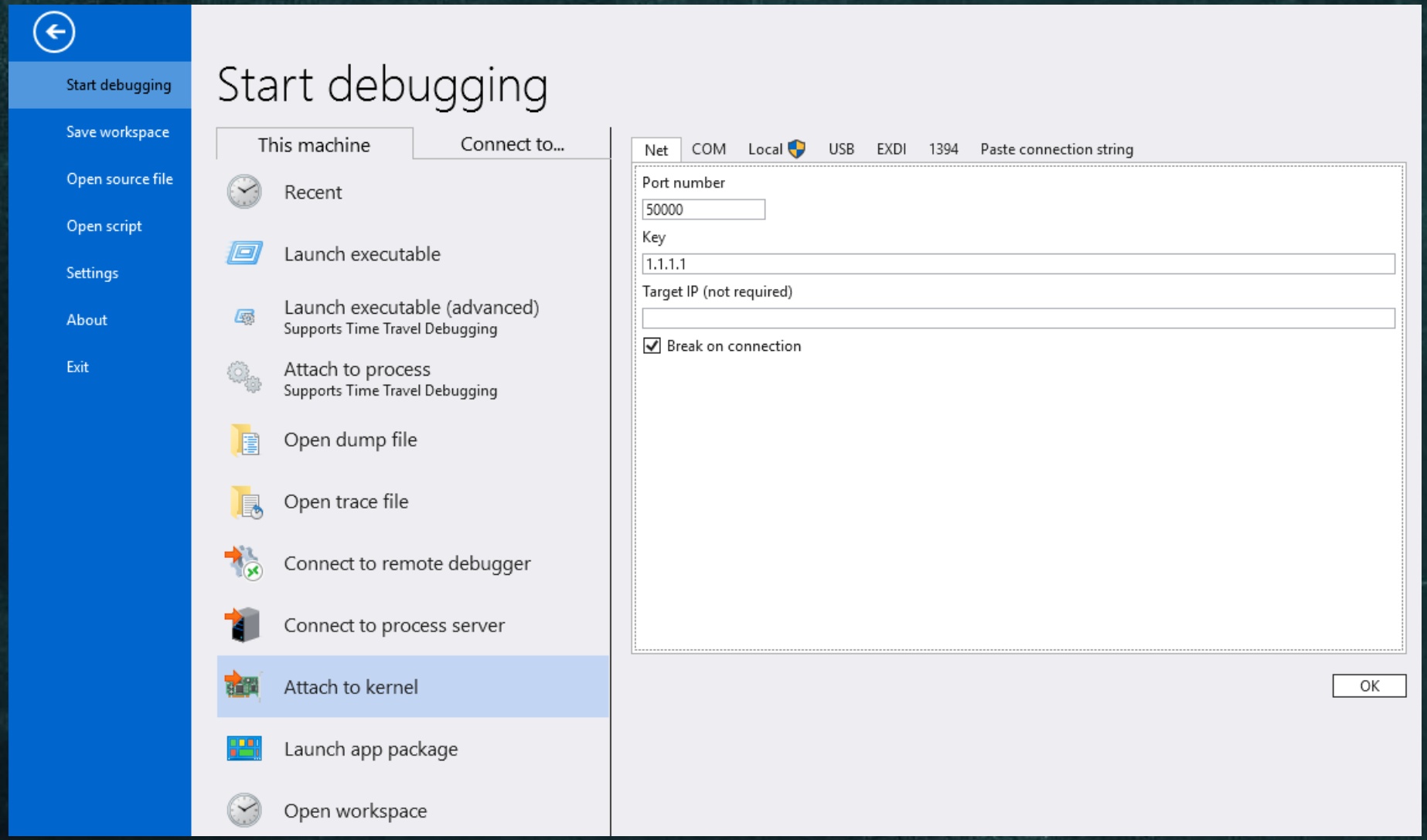
Dynamic Debug Setup – In Guest VM

```
c:\>bcdedit /set {bootmgr} bootdebug on
The operation completed successfully.

c:\>bcdedit /set bootdebug on
The operation completed successfully.

c:\>bcdedit /dbgsettings net hostip:172.23.0.1 port:50000 key:1.1.1.1
Key=1.1.1.1
```

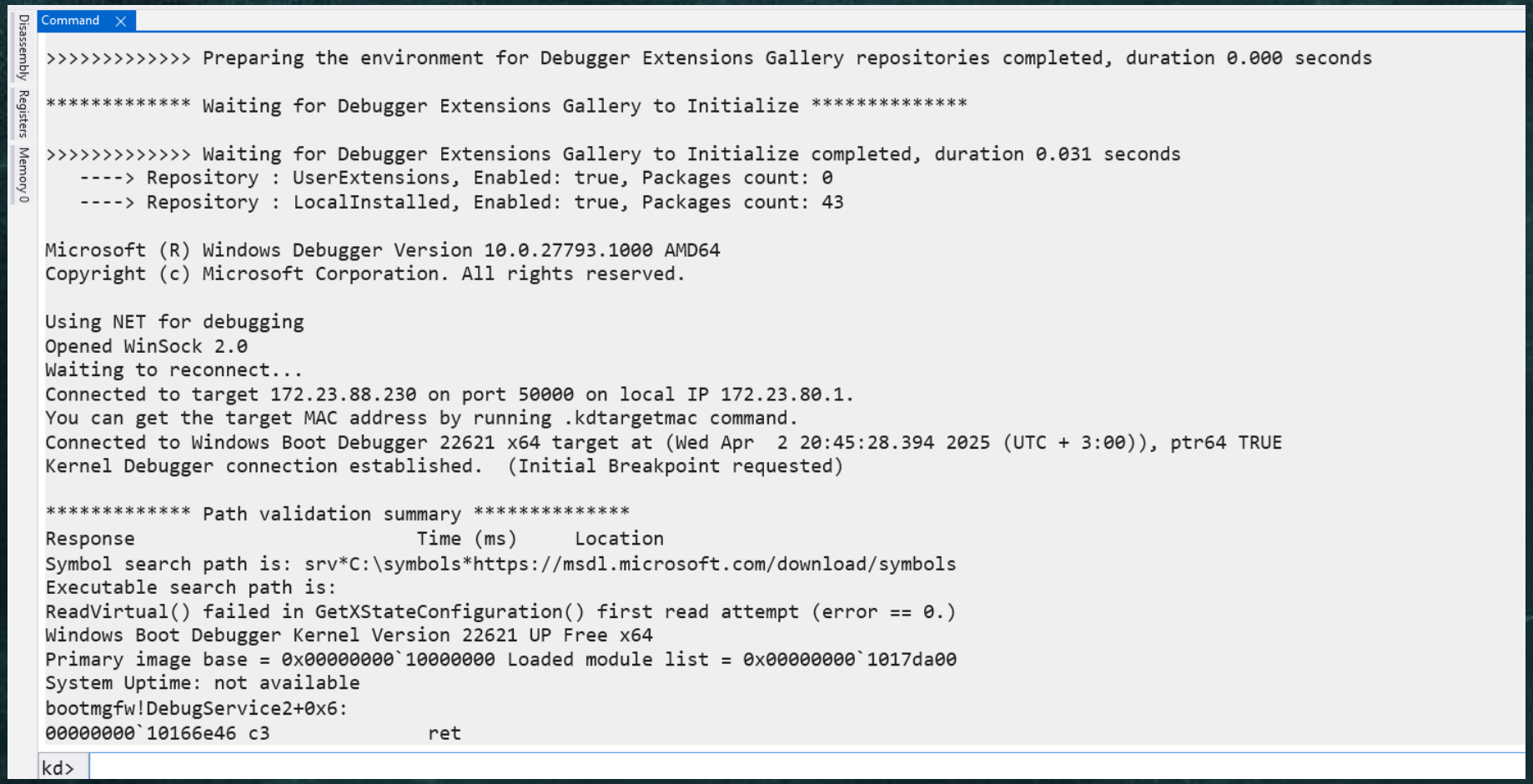
```
c:\>bcdedit /enum all

Resume from Hibernate
---------------------
identifier              {91c5aed0-f20b-11ef-8892-d8db38b9011f}
device                  partition=C:
path                    \Windows\system32\winresume.efi
description             Windows Resume Application
```

```
c:\>bcdedit /set {91c5aed0-f20b-11ef-8892-d8db38b9011f} bootdebug on
The operation completed successfully.
```

Dynamic Debug Setup - In Host - WinDBG Settings



Dynamic Debug Setup - Windbg




THE END!