Driving Vulnerabilities

Navigating the Road of BYO[V]D Attacks

Dana Behling (she/her)

Senior Threat Researcher, VMware Carbon Black

30 April 2023

Wife, Mom, Threat Researcher

Married with two boys, 10 and 13

vmware Carbon Black

Government Contractor leidos

NSA Civilian

Alumni of University of Hawai‘i at Mānoa

Driving Vulnerabilities: Navigating the Road of BYOVD Attacks

Agenda

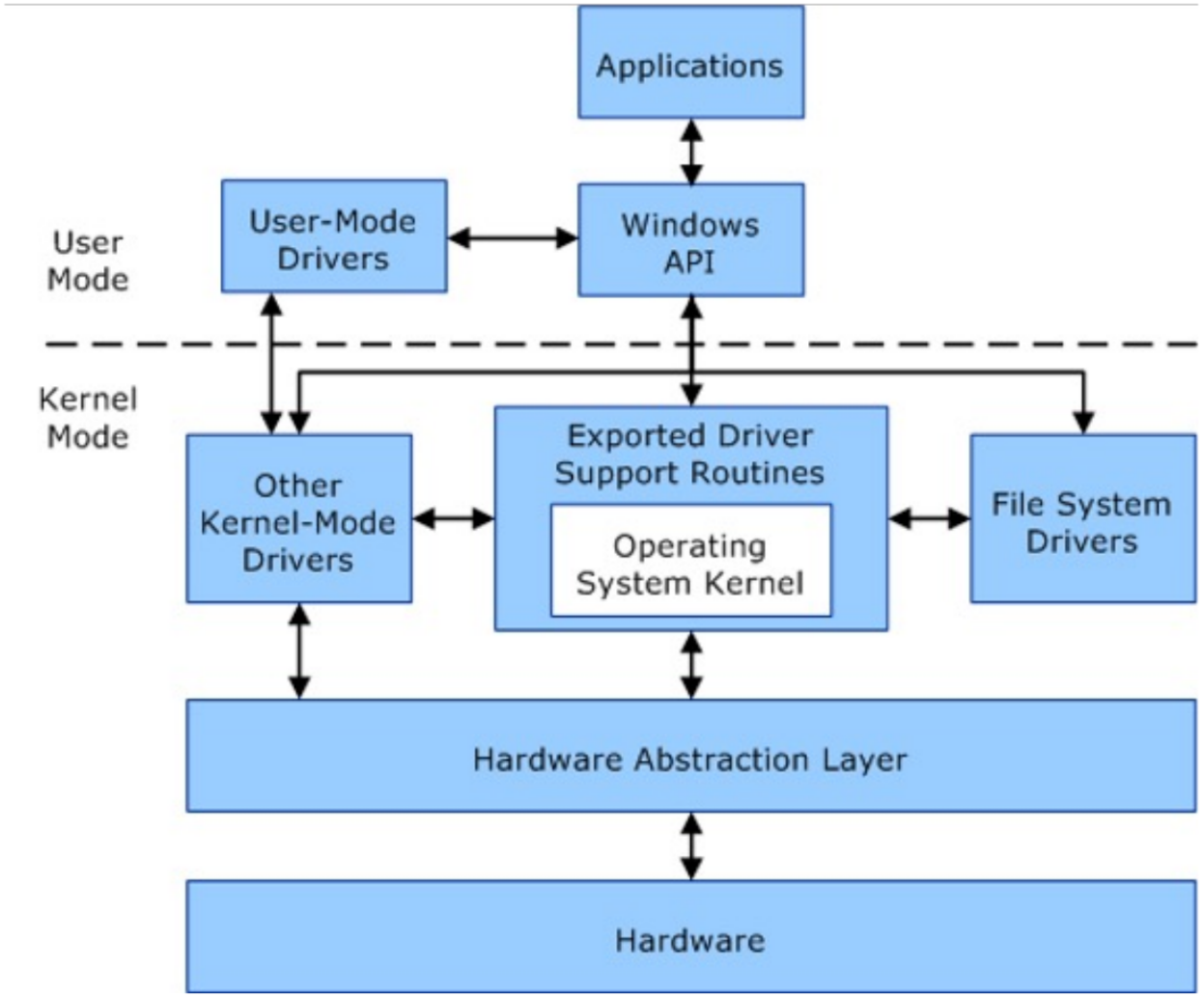
- Bringing It All Together
- Essential Knowledge
- Special Kernel Only Registers
- IOCTL Vulnerabilities
- Plug and Play Vulnerabilities
- Wrap-up and Questions

Drivers Used in Attacks

Kernel Rootkits

Simplistic Idealization

Reality


Applications
User-Mode Drivers
Windows API
User Mode
Kernel Mode
Other Kernel-Mode Drivers
Exported Driver Support Routines
Operating System Kernel
File System Drivers
Hardware Abstraction Layer
Hardware

MSRs & BYOVD

Model-specific registers

Instructions: RDMSR & WRMSR

IA32_LSTAR	IA-32e Mode System Call Target Address (R/W) Target RIP for the called procedure when SYSCALL is executed in 64-bit mode.
IA32_CSTAR	IA-32e Mode System Call Target Address (R/W) Not used, as the SYSCALL instruction is not recognized in compatibility mode.

WannaCrypt

"The exploit uses a *function-pointer overwrite* technique to direct control flow to the first-stage shellcode. This shellcode installs a second-stage shellcode as a *SYSENTER* or *SYSCALL* routine hook by overwriting *model-specific registers* (MSRs). If the target system is x86-based, it hooks the SYSENTER routine by overwriting *IA32_SYSENTER_EIP*. On x64-based systems, it overwrites *IA32_LSTAR* MSR to hook the SYSCALL routine."

https://www.microsoft.com/en-us/security/blog/2017/06/30/exploring-the-crypt-analysis-of-the-wannacrypt-ransomware-smb-exploit-propagation/

WRMSR

WRMSR — Write to Model Specific Register

Opcode	Instruction	Op/En	64-Bit Mode	Compat/Leg Mode	Description
0F 30	WRMSR	ZO	Valid	Valid	Write the value in EDX:EAX to MSR specified by ECX.

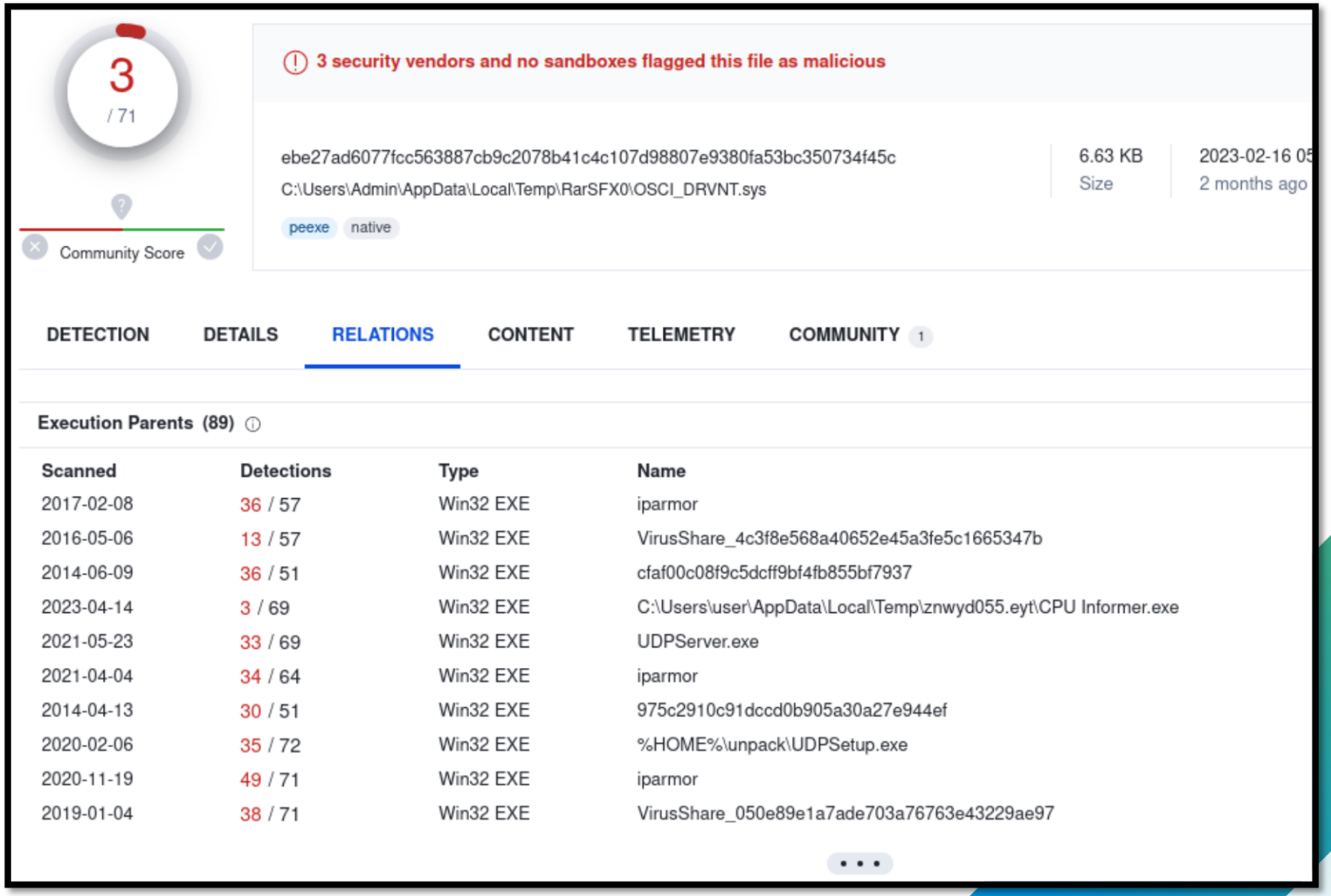
```
0045f545 55             PUSH    EBP
0045f546 f0             LOCK
0045f547 8b 42 08       MOV     EAX,dword ptr [EDX + 0x8]
0045f54a 89 45 f4       MOV     dword ptr [EBP + -0xc],EAX
0045f54d 8b 45 f8       MOV     EAX,dword ptr [EBP + -0x8]
0045f550 8b 55 f4       MOV     EDX,dword ptr [EBP + -0xc]
0045f553 8b 4d fc       MOV     ECX,dword ptr [EBP + -0x4]
0045f556 0f 30          WRMSR
0045f558 8b 4d 18       MOV     ECX,dword ptr [EBP + 0x18]
0045f55b c7 01 00       MOV     dword ptr [ECX],0x0
         00 00 00
0045f561 33 c0          XOR     EAX,EAX
0045f563 8b e5          MOV     ESP,EBP
0045f565 5d             POP     EBP
0045f566 c3             RET
```

From Blackmoon also known as KrBanker

YARA

```
strings:
    $IoCreateDevice = "IoCreateDevice" ascii wide
    $IoCreateDeviceSecure = "IoCreateDeviceSecure" ascii wide
    $WdmlibIoCreateDeviceSecure = "WdmlibIoCreateDeviceSecure" ascii wide
    $wr0 = {8B 4D ?? 8B 55 ?? 8B 45 ?? 0F 30}
    $wr1 = {8B 4D ?? 8B 45 ?? 8B 55 ?? 0F 30}
    $wr2 = {8B 55 ?? 8B 4D ?? 8B 45 ?? 0F 30}
    $wr3 = {8B 55 ?? 8B 45 ?? 8B 4D ?? 0F 30}
    $wr4 = {8B 45 ?? 8B 55 ?? 8B 4D ?? 0F 30}
    $wr5 = {8B 45 ?? 8B 4D ?? 8B 55 ?? 0F 30}
    $wr6 = {B8 ?? ?? ?? BA ?? ?? ?? B9 ?? ?? ?? 0F 30}
    $wr7 = {B8 ?? ?? ?? B9 ?? ?? ?? BA ?? ?? ?? 0F 30}
    $wr8 = {B9 ?? ?? ?? B8 ?? ?? ?? BA ?? ?? ?? 0F 30}
    $wr9 = {B9 ?? ?? ?? BA ?? ?? ?? B8 ?? ?? ?? 0F 30}
    $wra = {BA ?? ?? ?? B8 ?? ?? ?? B9 ?? ?? ?? 0F 30}
    $wrb = {BA ?? ?? ?? B9 ?? ?? ?? B8 ?? ?? ?? 0F 30}
    $DeviceIoControl = "DeviceIoControl" ascii wide
    $CreateFile = "CreateFile" ascii wide
condition:
    ($IoCreateDevice or
    $IoCreateDeviceSecure or
    $WdmlibIoCreateDeviceSecure) and
    (1 of ($wr*)) and
    $DeviceIoControl and
    $CreateFile and
    filesize < 10MB and not
    for any tag in vt.metadata.tags:
        ( tag == "corrupt" )
```

OSCI_DRIVNT.sys



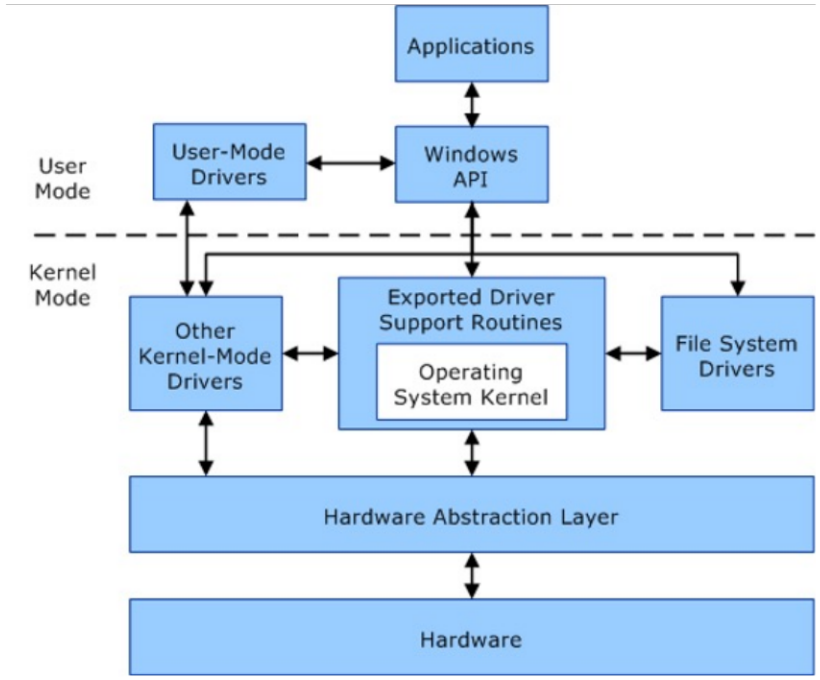
Scanned	Detections	Type	Name
2017-02-08	36 / 57	Win32 EXE	iparmor
2016-05-06	13 / 57	Win32 EXE	VirusShare_4c3f8e568a40652e45a3fe5c1665347b
2014-06-09	36 / 51	Win32 EXE	cfaf00c08f9c5dcff9bf4fb855bf7937
2023-04-14	3 / 69	Win32 EXE	C:\Users\user\AppData\Local\Temp\znwyd055.eyt\CPU Informer.exe
2021-05-23	33 / 69	Win32 EXE	UDPServer.exe
2021-04-04	34 / 64	Win32 EXE	iparmor
2014-04-13	30 / 51	Win32 EXE	975c2910c91dccd0b905a30a27e944ef
2020-02-06	35 / 72	Win32 EXE	%HOME%\unpack\UDPSetup.exe
2020-11-19	49 / 71	Win32 EXE	iparmor
2019-01-04	38 / 71	Win32 EXE	VirusShare_050e89e1a7ade703a76763e43229ae97

Virtualization-based Security (VBS)

- Supervisory Mode Execution Prevention (SMEP)
- Hypervisor [Protected] Code Integrity (HVCI)

  Device Guard

  Memory Integrity



"When I have a tough job in a plant and can't find an easy way to do it, I have a lazy man put on it. He'll find an easy way to do it in 10 days. Then we adopt that method."

Feb 1, 1947

Clarence E. Bleicher (1890-1952)

LAZY MAN MAKES TOUGH JOB SIMPLE

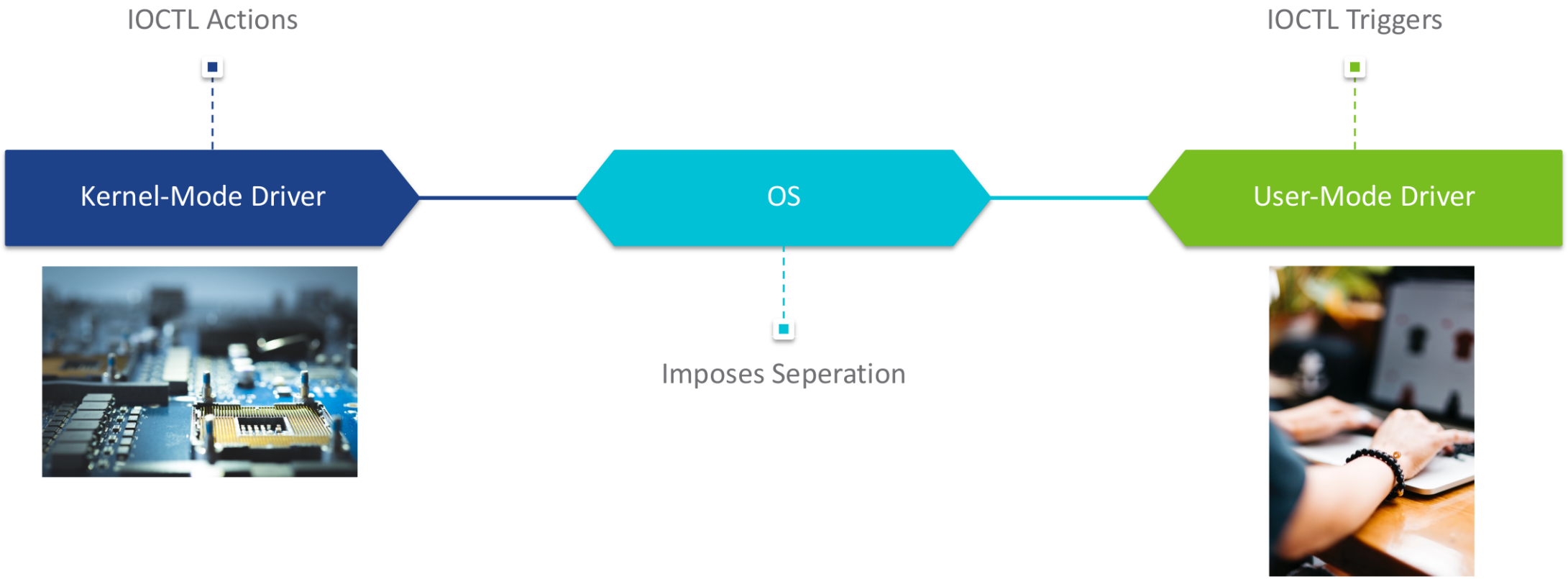
Chicago Tribune Press Service

WASHINGTON, Jan. 31.—A tip on how to solve difficult production problems was given the Senate Labor Committee today by Clarence E. Bleicher, president of the Chrysler Corp.'s De Soto division.

"When I have a tough job in the plant and can't find an easy way to do it," Bleicher said, "I have a lazy man put on it. He'll find an easy way to do it in 10 days. Then we adopt that method."

Unprotected IOCTL Requests in Drivers

CWE-782: Exposed IOCTL with Insufficient Access Control



Example IOCTL in Driver

```
case 0x80002000:
  uVar12 = get_zw_map_view_of_section
                     (param_1,(longlong *)pplVar9,uVar2,(ulon
  *(int *)(param_2 + 0x30) = (int)uVar12;
  plVar8 = (longlong *)pplVar9;
  if ((int)uVar12 < 0) {
    *(undefined4 *)(param_2 + 0x30) = 0xc000000d;
  }
  else {
    *(undefined8 *)(param_2 + 0x38) = 8;
  }
  break;
default:
  *(undefined4 *)(param_2 + 0x30) = 0xc000000d;
  break;
case 0x80002004:
  if (uVar2 < 8) {
    *(undefined4 *)(param_2 + 0x30) = 0xc0000001;
  }
  else {
    plVar8 = *pplVar9;
    uVar6 = ZwUnmapViewOfSection(0xffffffffffffffff,plVar8);
    *(undefined4 *)(param_2 + 0x30) = uVar6;
  }
  break;
```

Kernel Side

IoCreateDevice function (wdm.h)

Article • 01/06/2023 Feedback

The IoCreateDevice routine creates a device object for use by a driver.

Syntax

C++ Copy

```
NTSTATUS IoCreateDevice(
  [in]           PDRIVER_OBJECT  DriverObject,
  [in]           ULONG           DeviceExtensionSize,
  [in, optional] PUNICODE_STRING DeviceName,
  [in]           DEVICE_TYPE     DeviceType,
  [in]           ULONG           DeviceCharacteristics,
  [in]           BOOLEAN         Exclusive,
  [out]          PDEVICE_OBJECT  *DeviceObject
);
```

WdmlibIoCreateDeviceSecure function (wdmsec.h)

Article • 04/17/2022 Feedback

The **WdmlibIoCreateDeviceSecure** function (or **IoCreateDeviceSecure**) creates a named device object and applies the specified security settings.

Syntax

C++ Copy

```
NTSTATUS WdmlibIoCreateDeviceSecure(
  [in]           PDRIVER_OBJECT   DriverObject,
  [in]           ULONG            DeviceExtensionSize,
  [in, optional] PUNICODE_STRING  DeviceName,
  [in]           DEVICE_TYPE      DeviceType,
  [in]           ULONG            DeviceCharacteristics,
  [in]           BOOLEAN          Exclusive,
  [in]           PCUNICODE_STRING DefaultSDDLString,
  [in, optional] LPCGUID          DeviceClassGuid,
                 PDEVICE_OBJECT   *DeviceObject
);
```

Easy as 1-[2]-3

1. Connect to device
2. Authenticate (optional)
3. Send Command

Authenticate

- Security Reference Monitor (tokens)
  - SeAccessCheck
  - SePrivilegeCheck
  - Others
- Digital signature of calling application
- TLS like communication

User Side

DeviceIoControl function (ioapiset.h)

Article • 07/26/2022 Feedback

Sends a control code directly to a specified device driver, causing the corresponding device to perform the corresponding operation.

See the Assign drive letter sample.

Syntax

C++ Copy

```
BOOL DeviceIoControl(
  [in]                HANDLE       hDevice,
  [in]                DWORD        dwIoControlCode,
  [in, optional]      LPVOID       lpInBuffer,
  [in]                DWORD        nInBufferSize,
  [out, optional]     LPVOID       lpOutBuffer,
  [in]                DWORD        nOutBufferSize,
  [out, optional]     LPDWORD      lpBytesReturned,
  [in, out, optional] LPOVERLAPPED lpOverlapped
);
```

The Call

```
10013ec6 51              PUSH        this
10013ec7 6a 0c           PUSH        0xc
10013ec9 ff 75 08        PUSH        dword ptr [EBP + param_1]
10013ecc 6a 0c           PUSH        0xc
10013ece ff 75 08        PUSH        dword ptr [EBP + param_1]
10013ed1 68 34 20        PUSH        0x80002034
         00 80
10013ed6 50              PUSH        EAX
10013ed7 ff 15 cc        CALL        dword ptr [->KERNEL32.DLL::DeviceIoControl]
         d0 01 10
```

```
BVar2 = DeviceIoControl(*(HANDLE *)(this + 4),0x80002034,param_1,0xc,param_1,0xc,
                        (LPDWORD)&param_1,(LPOVERLAPPED)0x0);
```

Hunting

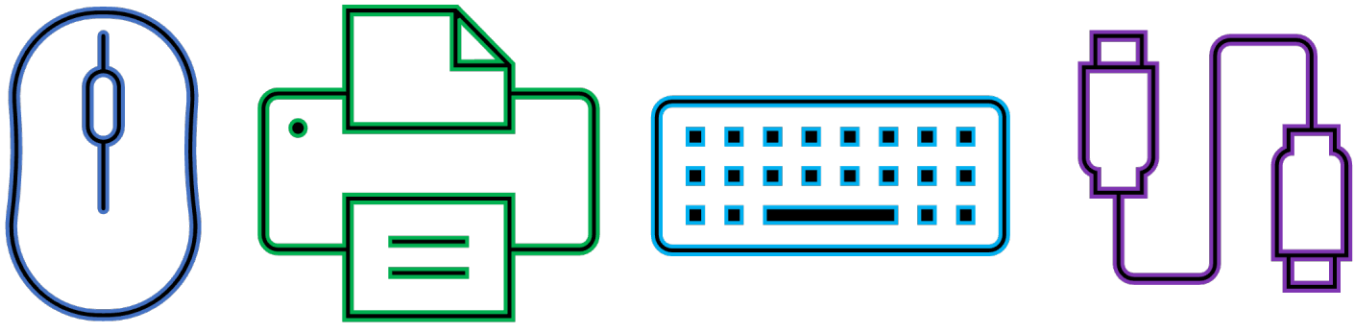
```
rule process_thread_manipulation_drivers_valid_sig
{
  strings:
      $IoCreateDevice = "IoCreateDevice" ascii wide
      $IoCreateDeviceSecure = "IoCreateDeviceSecure" ascii wide
      $WdmlibIoCreateDeviceSecure = "WdmlibIoCreateDeviceSecure" ascii wide
      $PsLookupProcessByProcessId = "PsLookupProcessByProcessId" ascii wide
      $ZwTerminateProcess = "ZwTerminateProcess" ascii wide
      $PsSuspendProcess = "PsSuspendProcess" ascii wide
      $PsSetCreateThreadNotifyRoutine = "PsSetCreateThreadNotifyRoutine" ascii wide
      $PsSetCreateProcessNotifyRoutineEx = "PsSetCreateProcessNotifyRoutine" ascii wide
      $IoValidateDeviceIoControlAccess = "IoValidateDeviceIoControlAccess" ascii wide
      $WdmlibIoValidateDeviceIoControlAccess = "WdmlibIoValidateDeviceIoControlAccess" ascii wide
  condition:
      ( $IoCreateDevice or $WdmlibIoCreateDeviceSecure) and
      $PsLookupProcessByProcessId and
      ( $ZwTerminateProcess or $PsSuspendProcess ) and
      $PsSetCreateThreadNotifyRoutine and
      $PsSetCreateProcessNotifyRoutineEx and
      not ($IoValidateDeviceIoControlAccess or
           $WdmlibIoValidateDeviceIoControlAccess or
           $IoCreateDeviceSecure)
      and for all i in (0..pe.number_of_signatures - 1):
        (pe.signatures[i].valid_on(pe.timestamp))
```

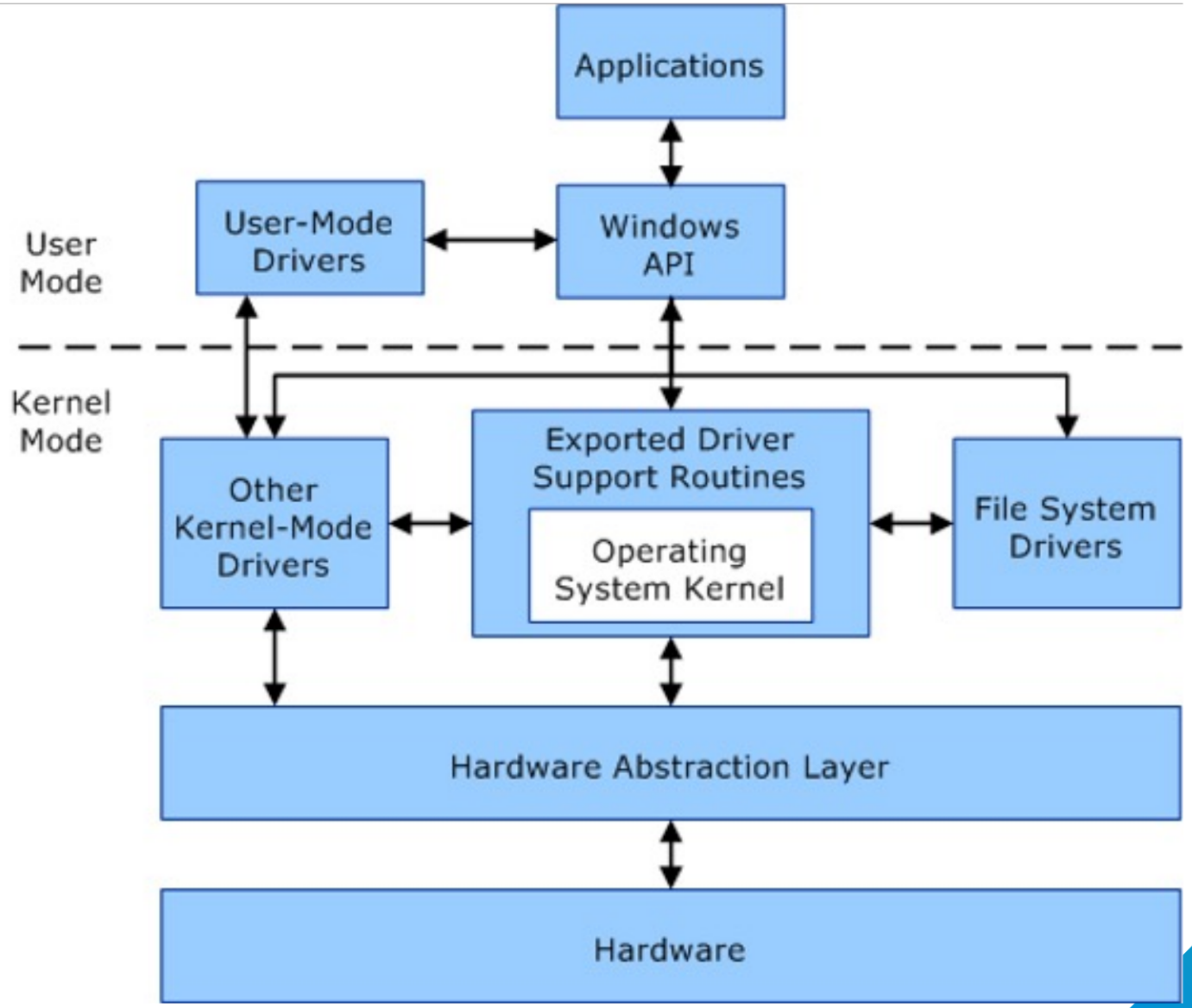
Sample of Results

7562F909AD5864BB70C0925EAAF78390CB12EF1… DriverInstaller.dll pedll signed overlay	process_thread_manipulation_drivers_valid_sig	2 / 55	6.98 MB	2022-10-08 08:51:50
A2909E2E2267EADB757D53EE099BC4E27318227… AppCheckD.sys peexe 64bits assembly signed native	process_thread_manipulation_drivers_valid_sig	0 / 72	119.39 KB	2022-10-26 07:57:19
64C5418FDD39897A28D7B7E87E23892B268C09D… active64_sha1_novm.sys peexe assembly overlay signed 64bits native	process_thread_manipulation_drivers_valid_sig	4 / 72	6.15 MB	2022-11-24 08:45:32
4590CC083229ACB1490EAD7E93F290A34FB1E60… dlflt.sys peexe assembly overlay signed 64bits native	process_thread_manipulation_drivers_valid_sig	0 / 70	2.84 MB	2022-12-22 19:00:30
FEA1FAAC491D37E2BDDD89A4252D8209CE3E945… filnk.sys peexe assembly overlay signed 64bits native	process_thread_manipulation_drivers_valid_sig	0 / 71	854.33 KB	2022-12-26 08:32:06
729C86DC36153A6CA7230E78AB9FD7EC66996B7… … engine source\cheat engine\bin\realmengine64.sys peexe assembly overlay signed 64bits native	process_thread_manipulation_drivers_valid_sig	15 / 70	92.61 KB	2023-01-19 16:55:50

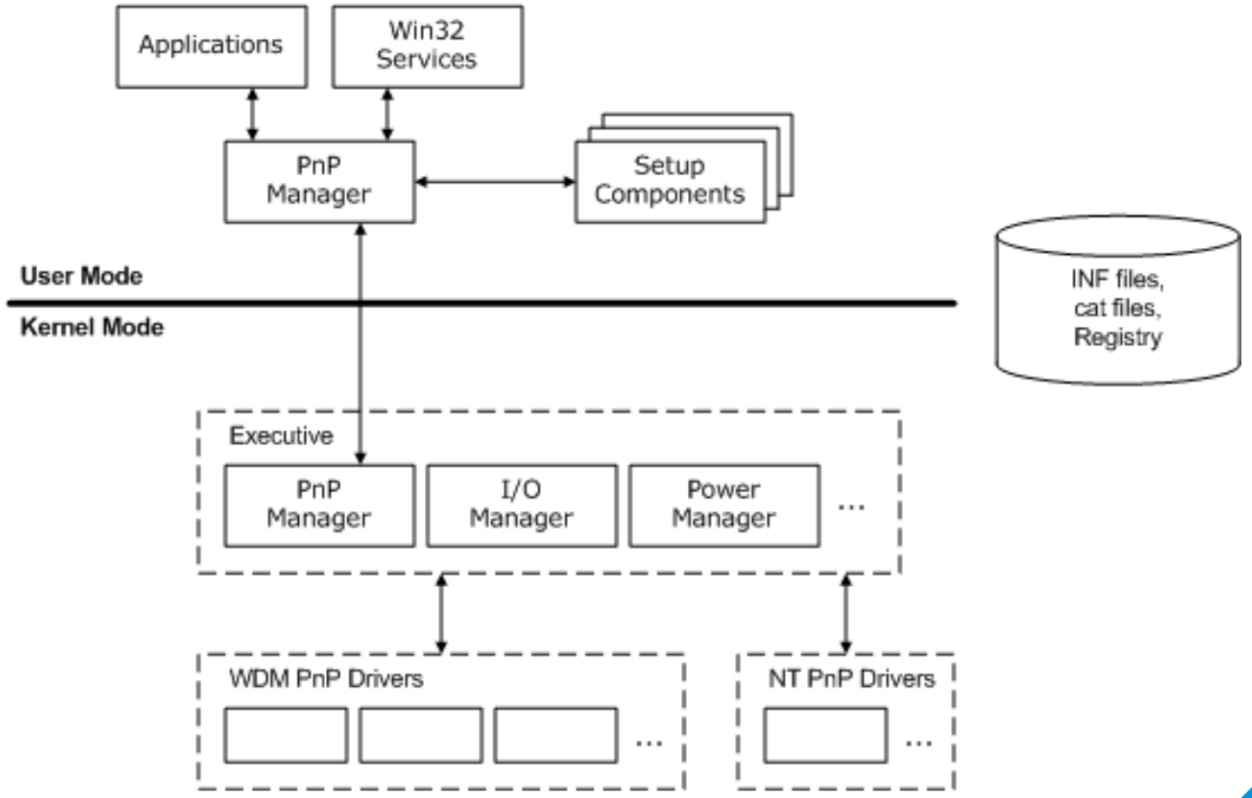
Plug and Play Drivers

Privilege Escalation




Applications
User-Mode Drivers
Windows API
User Mode
Kernel Mode
Other Kernel-Mode Drivers
Exported Driver Support Routines
Operating System Kernel
File System Drivers
Hardware Abstraction Layer
Hardware

Extra layer of complexity



Co-installers

- WdfPreDeviceInstall[Ex]
- Device installation function codes
- Complete pre-installation steps
- Provide property pages
- Finish the install actions

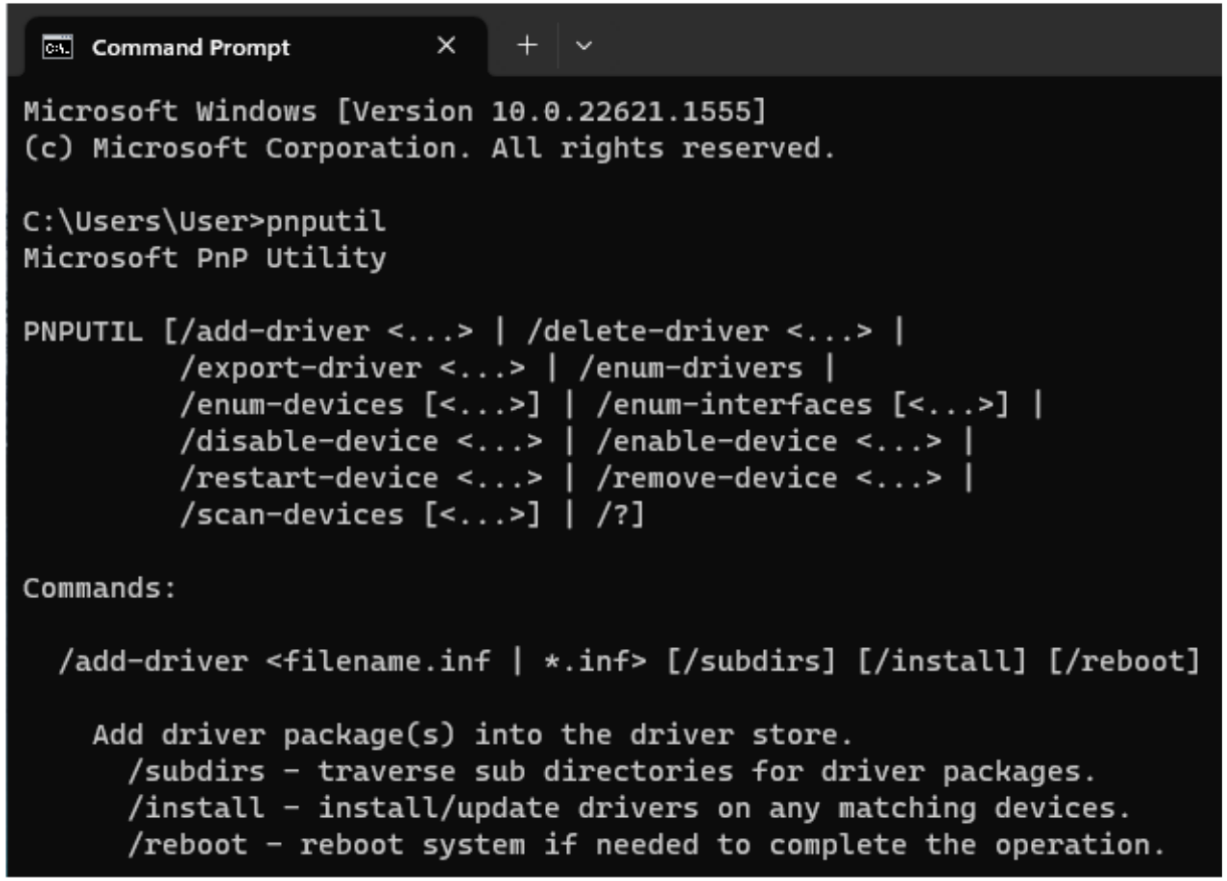
Hunting

100 % **Finished** | **danab-1682761631** 8 hours ago

import "vt" import "pe" /* Possible malicious driver with valid signature. */ rule pnp_coinstaller_insecure_driver { strings: $IoCreateDevice = "IoCreateDevice… **160** matches

```
 9  strings:
10      $IoCreateDevice = "IoCreateDevice" ascii wide
11      $WdmlibIoCreateDeviceSecure = "WdmlibIoCreateDeviceSecure" ascii wide
12      $IoValidateDeviceIoControlAccess = "IoValidateDeviceIoControlAccess" ascii wide
13      $WdmlibIoValidateDeviceIoControlAccess = "WdmlibIoValidateDeviceIoControlAccess" ascii wide
14      $IoCreateDeviceSecure = "IoCreateDeviceSecure" ascii wide
15      $WdfPreDeviceInstallEx = "WdfPreDeviceInstall" ascii wide
16  condition:
17      ( $IoCreateDevice or $WdmlibIoCreateDeviceSecure) and
18      not ($IoValidateDeviceIoControlAccess or
19           $WdmlibIoValidateDeviceIoControlAccess or
20           $IoCreateDeviceSecure)
21      and $WdfPreDeviceInstallEx
22      and for all i in (0..pe.number_of_signatures - 1):
23        (pe.signatures[i].valid_on(pe.timestamp))
24      and not for any tag in vt.metadata.tags:
25        ( tag == "corrupt" )
```

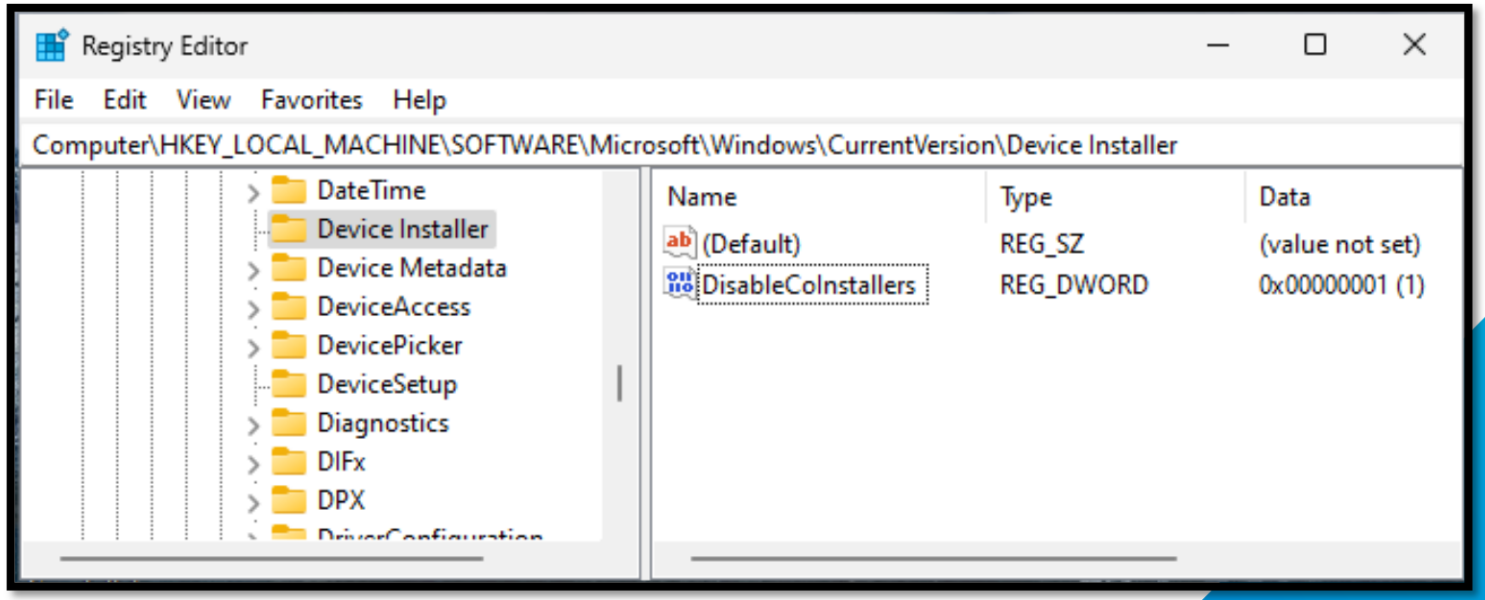
PnPUtil


Command Prompt
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.
C:\Users\User>pnputil
Microsoft PnP Utility
PNPUTIL [/add-driver <...> | /delete-driver <...> |
/export-driver <...> | /enum-drivers |
/enum-devices [<...>] | /enum-interfaces [<...>] |
/disable-device <...> | /enable-device <...> |
/restart-device <...> | /remove-device <...> |
/scan-devices [<...>] | /?]
Commands:
/add-driver <filename.inf | *.inf> [/subdirs] [/install] [/reboot]
Add driver package(s) into the driver store.
/subdirs - traverse sub directories for driver packages.
/install - install/update drivers on any matching devices.
/reboot - reboot system if needed to complete the operation.

Prohibit Co-Installers

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Device Installer

DisableCoInstallers



Wrap-up

- Firmware, bootloaders, UEFI
- Driver Signature Enforcement (DSE)
- Vulnerable Driver Blocklists
- Indicators of compromise
- Threat Landscape

Thank you.

Driving Vulnerabilities

Navigating the Road of BYO[V]D Attacks

Dana Behling
Twitter: @danabehling
Mastodon: @danafaye@infosec.exchange