Using Drivers

Caleb McGary

About Me

Red Teamer at MSFT

I am bad at twitter; I follow you, you shouldn't follow me

2 dogs:

- https://aka.ms/ludwig
- https://aka.ms/wolfgang

I enjoy coding, reading, astronomy, and occasionally my job

Opinions are **my own** and not representative of my employer.

Code

Slides and code samples here: GITHUB

Capabilities pushed to Faction: https://github.com/FactionC2/

- Made by Jared Haight: https://twitter.com/jaredhaight, https://c2.lol

Additional GitHub references:

https://github.com/Microsoft/Windows-driver-samples

DEMO

Let’s do this live!

Brass Tacks

Evaluated our surface area from past attacks

We were not really doing much space with drivers

Decided to put some effort into playing with them as a result

This talk is generally Windows heavy, but I will cover a few aspects of Linux; no MacOS (ask someone smarter than me for MacOS)

Serpent Process

Look at what attackers are doing

See where we can replicate

Try it out

Fail often / fail fast

20 percent time project (for most innovations)

In the News

Mysterious Avatar rootkit with API, SDK, and Yahoo Groups for C&C communication

Chinese-speaking APT LuckyMouse uses malicious NDISProxy drivers to distribute Trojans

September 11, 2018 | Threat Actors

Home > Malware

TODAY'S TOP STORIES

What is Stuxnet,

Blog >

Duqu Malware Techniques Us

Duqu 2.0 is a really impressive piece of malware,

Kernel Write-What-Where in Qualcomm Driver == LPE

APT REPORTS

Shedding Skin – Turla's Fresh Faces

'Screwed Drivers': Driver Vulnerabilities Affect Intel, AMD, Other Vendors

These newly disclosed vulnerabilities can allow attackers to install malware directly on device firmware, giving malicious software the ability to remain on the device, even after the operating system has been reinstalled, according to Intel-backed security startup Eclypsium.

Attackers

Actions on Objectives

- Process Injection, Lateral Movement, etc

Privilege Escalation

- User to System

Persistence

EDR Evasion

- Disable or disarm defensive measures present

Where we can replicate

Persistence

- This helps us; but requires substantial effort (we can't leave a serious bug around)
- We are testing this option out at the moment, using internally developed tooling

EDR Evasion

- This is much more useful, as we can limit our exposure (driver only needs to be loaded for short period of time)
- Does remove telemetry from box, but doesn't necessarily weaken security posture of system

Choose Your Own Adventure

Use an existing driver

- Discovery required
- More challenging to control
- Stealthier
- Tougher to remediate / be evicted

Build your own driver

- Need a code sign certificate
- Need to write one
- Powerful, full control over features and functionality
- Potentially easier to detect

Use an existing driver

Possibly the easiest and most common thing

Use an existing driver

Look for:

- Stuff that is SPC signed
  - WQHL vs SPC
- Things outside of systemroot/system32/drivers (if 64 bit)
- Things for older versions of the OS

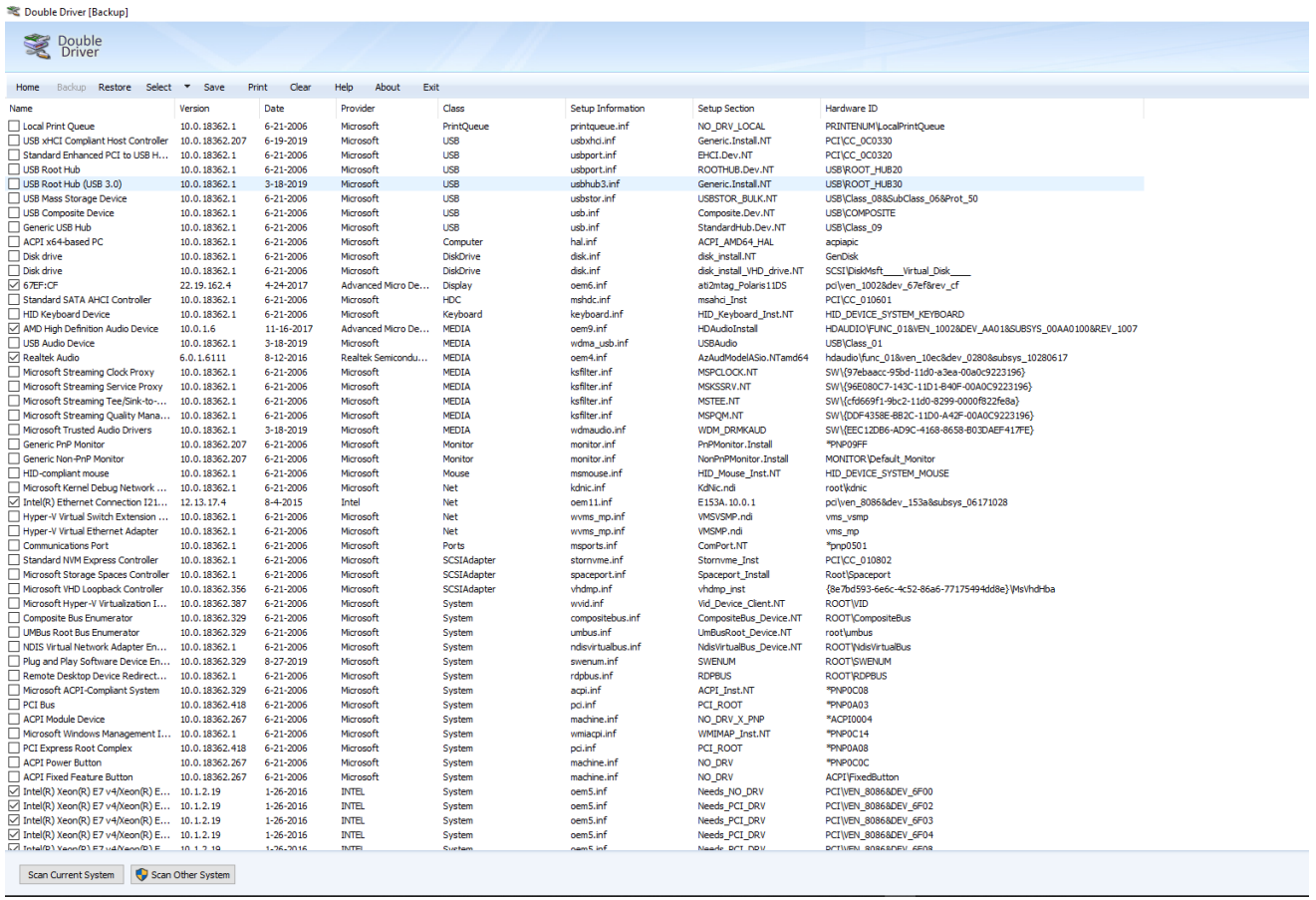
Enumeration

What drivers are installed on my system?

(Windows Demo) – driverquery /?

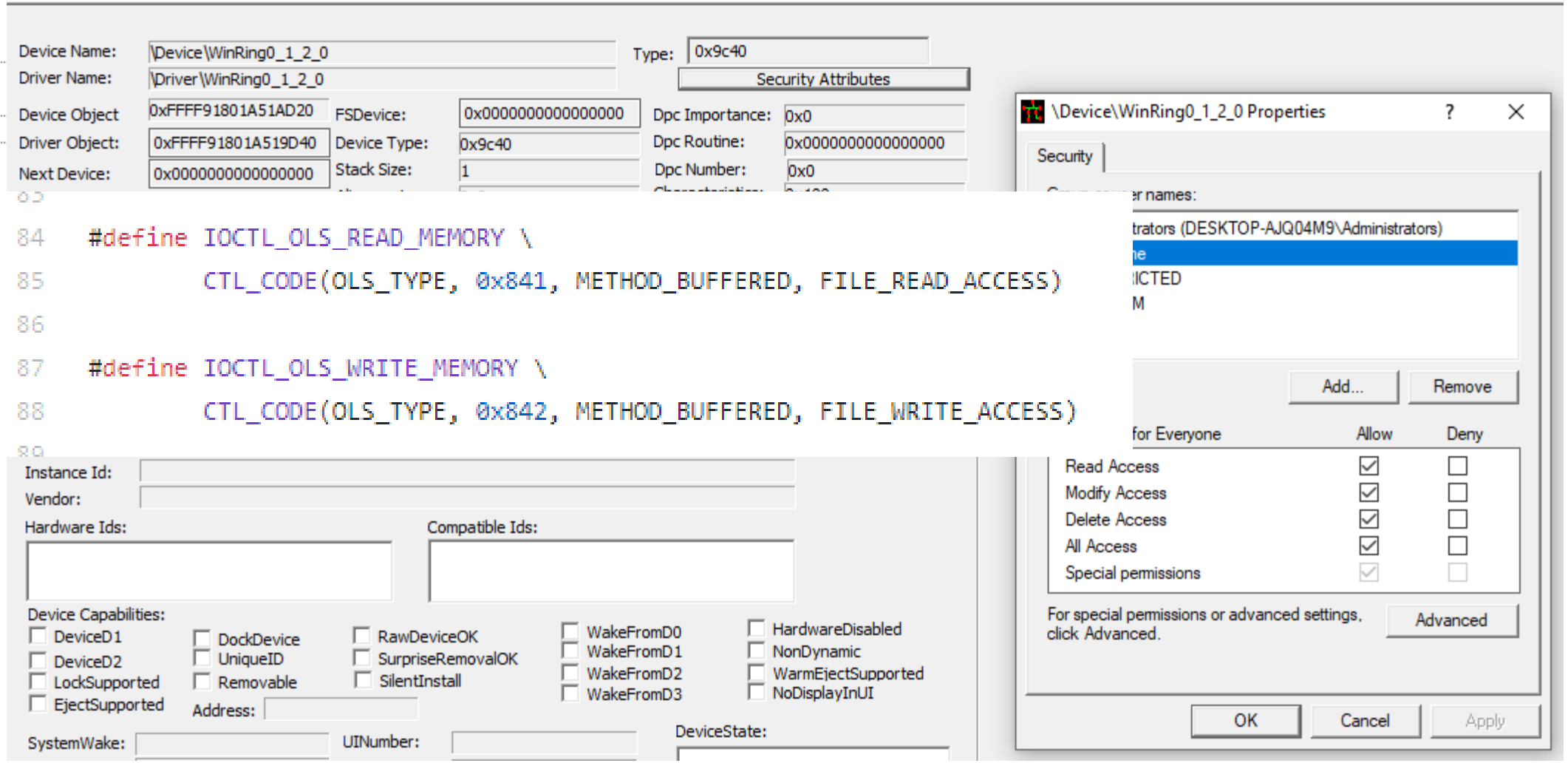
- driverquery /S system /U domain\user /P password /FO LIST
- pnputil /enum-drivers
- Double Driver Freeware to back stuff up: http://boozet.org/download.htm
- Device Tree – legacy tool

(Linux Demo) – lsmod or cat /proc/modules



```
PS D:\demo>
```

Example


Device Name: \Device\WinRing0_1_2_0
Driver Name: \Driver\WinRing0_1_2_0
Type: 0x9c40
Security Attributes
Device Object 0xFFFF91801A51AD20
Driver Object: 0xFFFF91801A519D40
Next Device: 0x0000000000000000
FSDevice: 0x0000000000000000
Device Type: 0x9c40
Stack Size: 1
Dpc Importance: 0x0
Dpc Routine: 0x0000000000000000
Dpc Number: 0x0
#define IOCTL_OLS_READ_MEMORY \
CTL_CODE(OLS_TYPE, 0x841, METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_OLS_WRITE_MEMORY \
CTL_CODE(OLS_TYPE, 0x842, METHOD_BUFFERED, FILE_WRITE_ACCESS)
Instance Id:
Vendor:
Hardware Ids:
Compatible Ids:
Device Capabilities:
DeviceD1
DeviceD2
LockSupported
EjectSupported
DockDevice
UniqueID
Removable
RawDeviceOK
SurpriseRemovalOK
SilentInstall
WakeFromD0
WakeFromD1
WakeFromD2
WakeFromD3
HardwareDisabled
NonDynamic
WarmEjectSupported
NoDisplayInUI
Address:
SystemWake:
UINumber:
DeviceState:
\Device\WinRing0_1_2_0 Properties
Security
trators (DESKTOP-AJQ04M9\Administrators)
Add...
Remove
for Everyone
Allow
Deny
Read Access
Modify Access
Delete Access
All Access
Special permissions
For special permissions or advanced settings, click Advanced.
Advanced
OK
Cancel
Apply

What you need to know

```
bResult = DeviceIoControl(hDevice,                       // device to be queried
                          IOCTL_DISK_GET_DRIVE_GEOMETRY, // operation to perform
                          NULL, 0,                       // no input buffer
                          pdg, sizeof(*pdg),            // output buffer
                          &junk,                         // # bytes returned
                          (LPOVERLAPPED) NULL);          // synchronous I/O

CloseHandle(hDevice);
```

How to call Device

```
NULL,
OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL,
NULL
);
```

```
FILE_GENERIC_WRITE,
&isb,
FILE_SHARE_READ | FILE_SHARE_WRITE,
FILE_NON_DIRECTORY_FILE
);
```

Tooling to help

Device Tree: http://www.osronline.com/article.cfm%5earticle=97.htm

IRPTracker: http://www.osronline.com/article.cfm%5earticle=199.htm

IOCTL Fuzzer: https://code.google.com/archive/p/ioctlfuzzer/

IOCTLBF: https://github.com/koutto/ioctlbf

WinObj: https://docs.microsoft.com/en-us/sysinternals/downloads/winobj

Windows Object Explorer 64bit: https://github.com/hfiref0x/WinObjEx64

Driver capabilities you want

Read / Write Memory

Allocate Memory

Accessible by Everyone

- If you're hunting privilege escalation 0-day

Existing Useful Drivers

WinRing0_1_*

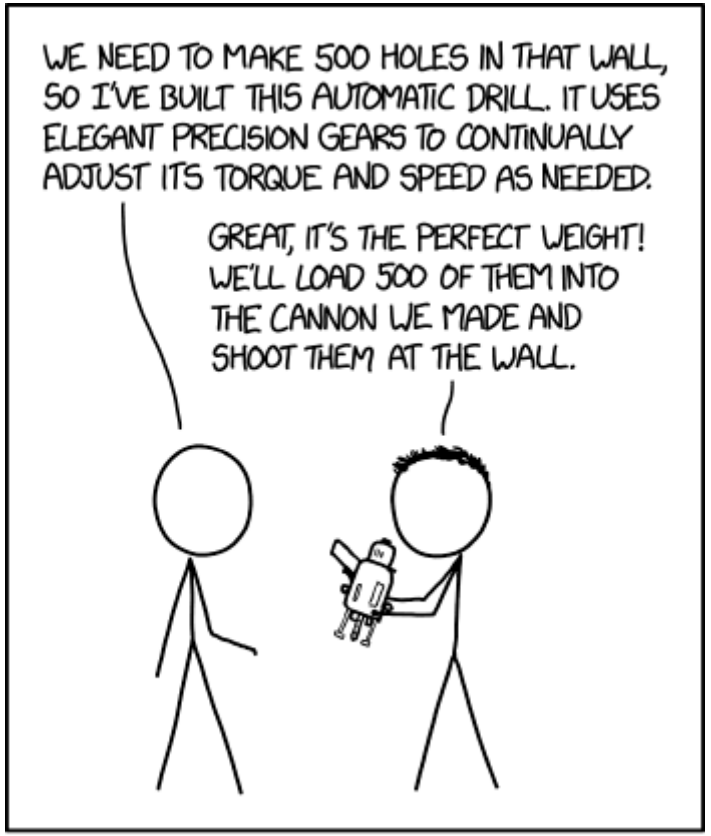
ProcessHacker

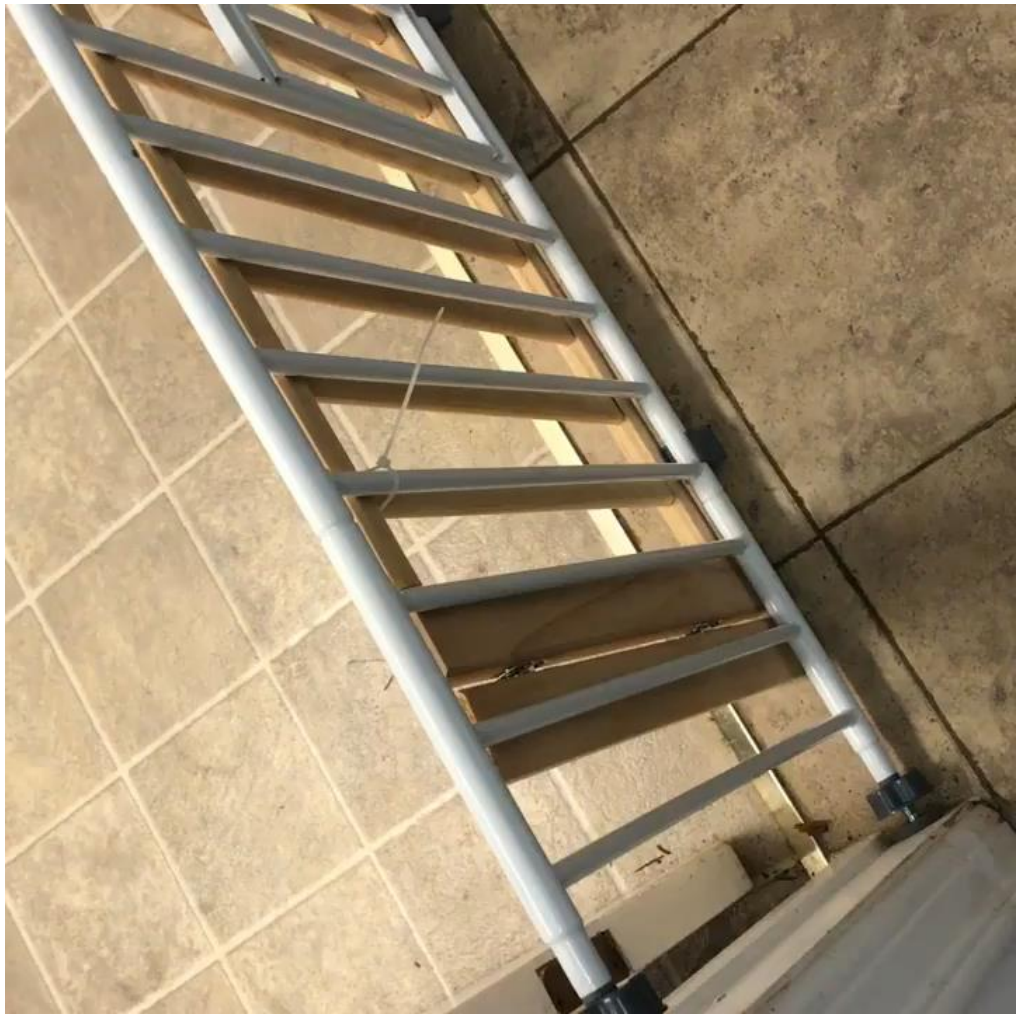
- Many other types of “older system / perf monitoring software”

VirtualBox (older versions)

BYOD

Disclaimer: this is NOT a comprehensive course





My preferred route

Windows: Drivers

Kernel Mode Drivers (Windows Driver Model - WDM)

- Plug & Play, I/O (filter drivers), Security, OS stuff
- Chaining -> Higher Level Drivers (NTFS) call lower level drivers (filter drivers) call lowest level drivers (hardware / legacy)

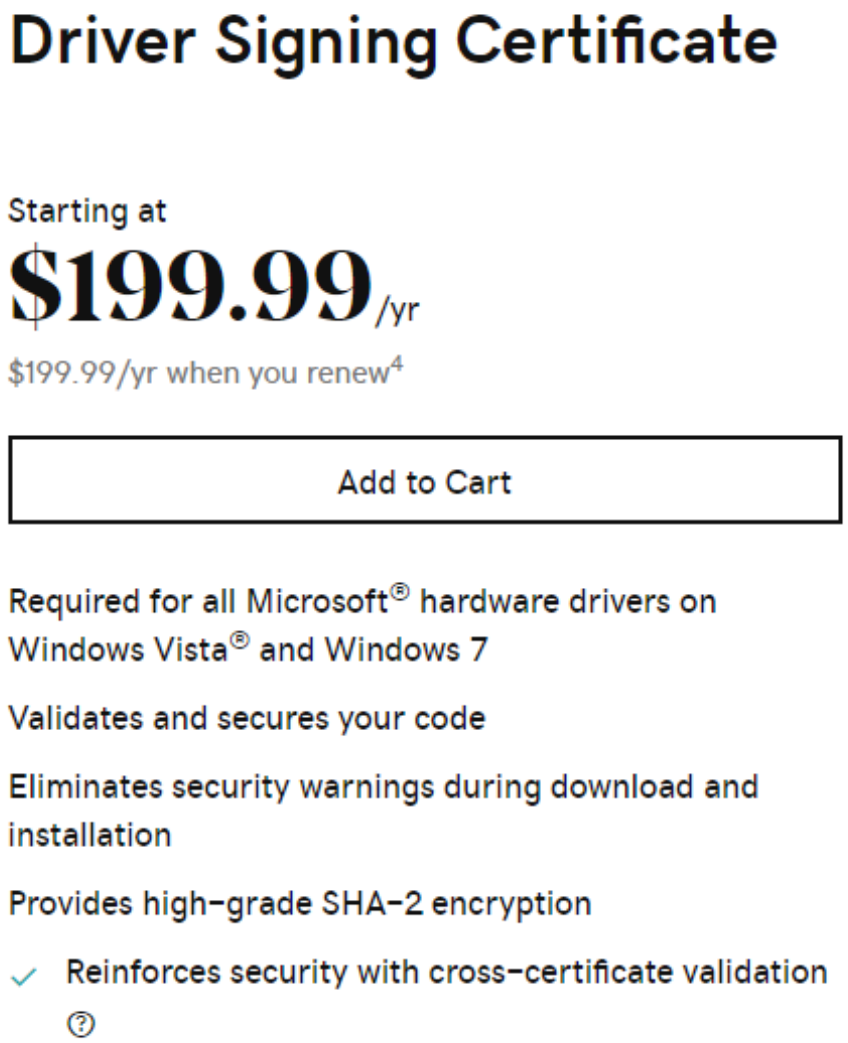
User Mode Drivers (UMDF)

- User-mode environment ☹
- File system, display (full devices) and print drivers CANNOT be UMDF

Windows: Drivers

Signing:

- Not required for 32-bit versions of Windows
- Cross Certificates Overview (short)
- Buy one, don't spend more than $200.
- Timestamp Certificates are also useful
- Pay attention to who is in the valid list; in 2021 this changes



Linux: Drivers

Generally part of Kernel

- Often a kernel module
  - Walkthrough
- Often proprietary ☹
- Can be written in a variety of things, just not C++ (Rust, C, Assembler, etc)

Install Shim Required

Demo

- Loading Process Hacker KMDF
  - CreateService
  - Start Service
  - NTOpenFile
  - This is noisy, but often undetected (loading a USB drive loads a driver)
- Code sample on github; you can probably do it without creating a service btw, pnputil + rundll32 is an option

Features you will want

Read / Write / Allocate Memory

- Possibly from nonpaged pool (rather than NX pool)
- Buffered I/O -> for working with user land

Windows: Process Context Switching

Windows: File System operations

Windows: Security - https://github.com/Microsoft/SymCrypt

Generic: System / method call hooking

Tooling

WDF Verifier: debugging drivers

WDK (DDK): this is your core toolset if you write your own

Device Fundamentals Tests: black box tests; need a testing box

- This finds lot of things; many are not security specific
- Penetration Tests (Fuzz Tests and I/O Spy + I/O Attack)

Visual Studio (worth it)

Detections

Talk about EventLog

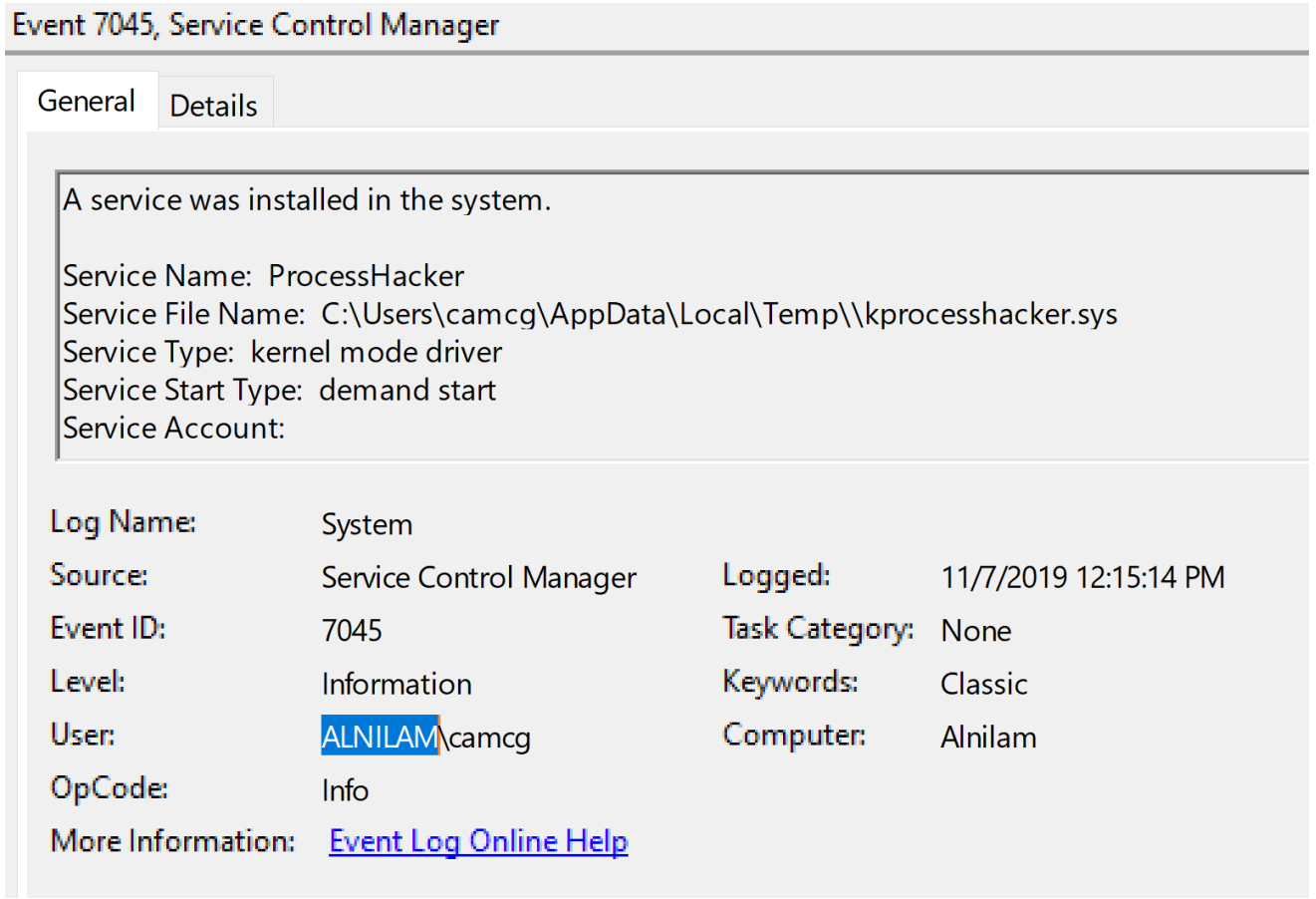
Event ID: 7045

Service was installed on your system

- One-time event
- Child devices don't always show up if you find a vulnerable pnpdevice
- Unload does **not** generate an event

ELAM **should** catch you

- If it has your driver signatured



Conclusion

That’s it y’all

Being inside the inner ring walls

EDR evasion / destruction / what have you
- You may have to win a race condition

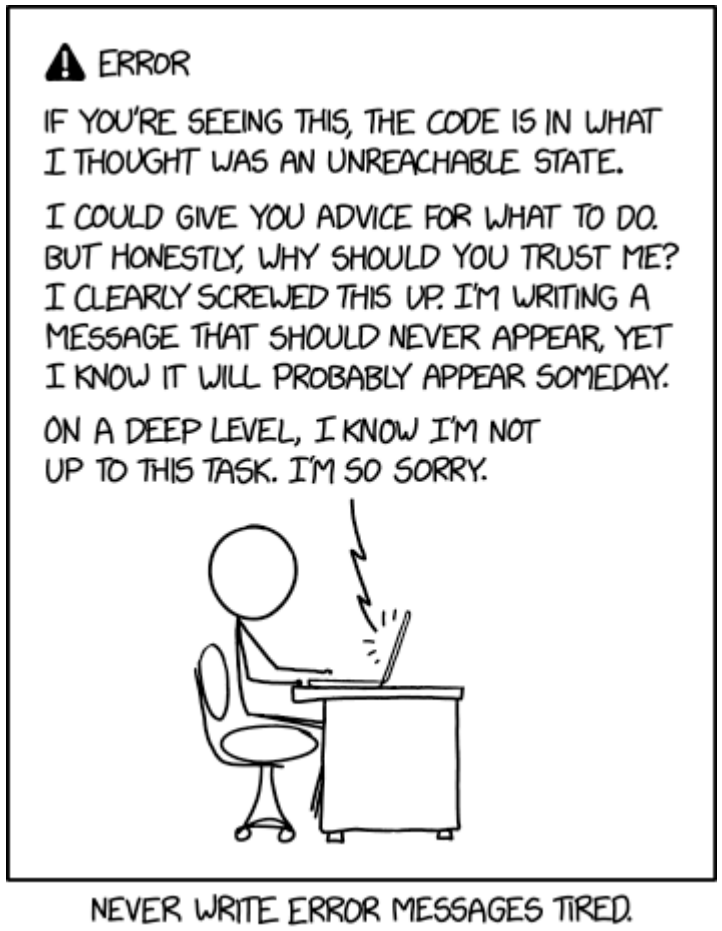
Persistence
- Load on boot (call DriverEntry routine)
- Reference binary from hidden filesystem, etc
- Go to firmware and hide there

Actions on Objectives
- This may or may not meet your testing model

Happy Hacking

Slides and source: https://aka.ms/UsingDrivers

Contact: caleb.mcgary@gmail.com



Backup Screenshots

In case of demo fail

Faction Components - Faction C2 | Faction Console

Not secure | 10.0.0.4/#/console/3

FACTION | Agents | Tasks | Payloads | Transports | Files | IOCs | admin

Filter agents..

41_JXcetcin8 Admin

FactionWin10\caleb
FactionWin10 (PID: 1308)
DIRECT Transport
11/07 8:02:25 PM

ping	Pings a computer or list of computers	winlib	true
portscan	Scan port(s) on computer(s)	winlib	true
powershell	Execute a PowerShell command through .NET. Doesn't launch powershell.exe and by default will attempt to bypass AMSI and ScriptBlock logging	winlib	true
reg	Work with the registry	winlib	true
services	List Services on local or remote computers	winlib	true
wmi	Execute a WMI query	winlib	true
driver	Work with Kernel Drivers	winlib	true
assembly	Loads and executes a .NET assembly	stdlib	true
cd	Change the directory you're operating out of	stdlib	true
download	Downloads a file from Faction to the Agent	stdlib	true
ls	List the contents of a directory or path	stdlib	true
mkdir	Create a directory	stdlib	true
ps	List processes	stdlib	true
pwd	Returns the directory that the agent is currently	stdlib	true

Send

Faction Components - Faction C2 | Faction Console

Not secure | 10.0.0.4/#/console/3

FACTION | Agents | Tasks | Payloads | Transports | Files | IOCs | admin

Filter agents..

41_JXcetcin8 Admin

FactionWin10\caleb
FactionWin10 (PID: 1308)
DIRECT Transport
11/07 8:07:37 PM

```
 127.0.0.1                                                    true

[7:32:00 PM] [admin]
F2> driver /Operation:"Enumerate"

[7:32:06 PM] [41_JXcetcin8] [#31]
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26
 \  S  y  s  t  e  m  R  o  o  t  \  s  y  s  t  e  m  3  2  \  n  t  o  s  k  r
 \  S  y  s  t  e  m  R  o  o  t  \  s  y  s  t  e  m  3  2  \  h  a  l  .  d  l
 \  S  y  s  t  e  m  R  o  o  t  \  s  y  s  t  e  m  3  2  \  k  d  .  d  l  l
 \  S  y  s  t  e  m  R  o  o  t  \  s  y  s  t  e  m  3  2  \  m  c  u  p  d  a
 \  S  y  s  t  e  m  R  o  o  t  \  S  y  s  t  e  m  3  2  \  d  r  i  v  e  r
 \  S  y  s  t  e  m  R  o  o  t  \  S  y  s  t  e  m  3  2  \  d  r  i  v  e  r
 \  S  y  s  t  e  m  R  o  o  t  \  S  y  s  t  e  m  3  2  \  d  r  i  v  e  r
 \  S  y  s  t  e  m  R  o  o  t  \  S  y  s  t  e  m  3  2  \  d  r  i  v  e  r
 \  S  y  s  t  e  m  R  o  o  t  \  S  y  s  t  e  m  3  2  \  d  r  i  v  e  r
 \  S  y  s  t  e  m  R  o  o  t  \  s  y  s  t  e  m  3  2  \  P  S  H  E  D  .
```

Send