The Landscape of USB Driver Vulnerabilities

"I though YOU were going to handle security…"

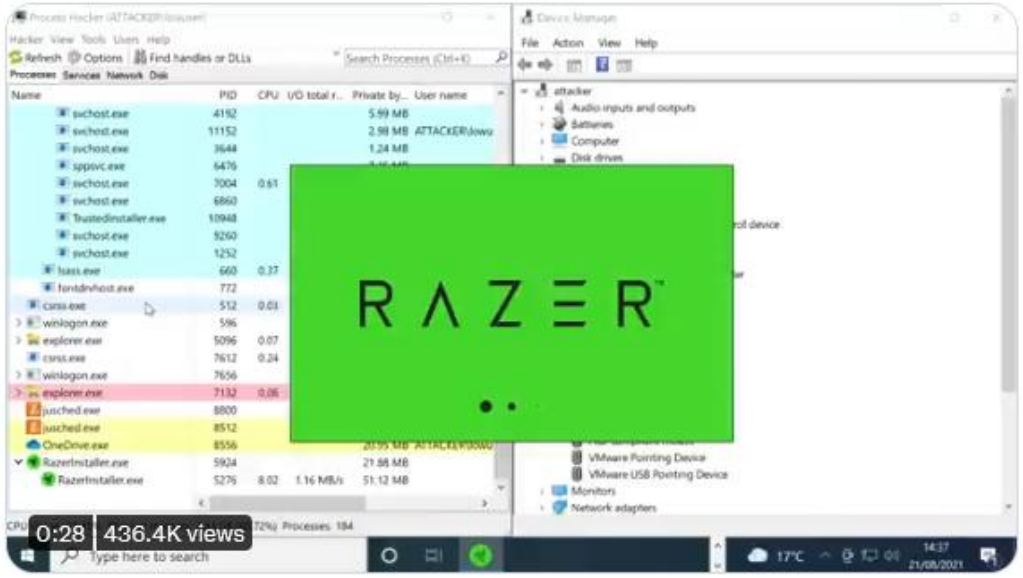
Topics

- Razer Device Local Privilege Escalation
- How Windows Selects a Driver
- How Windows Downloads Drivers (and you can too)
- Emulating a USB Device ID
- Hands On: Programming a Digispark Kickstarter USB Device

The "Original" Exploit

Need local admin and have physical access?
- Plug a Razer mouse (or the dongle)
- Windows Update will download and execute RazerInstaller as SYSTEM
- Abuse elevated Explorer to open Powershell with Shift+Right click

Tried contacting @Razer, but no answers. So here's a freebie



5:55 AM · Aug 21, 2021 · Twitter Web App

The Drop

On Aug 21, @jonh4t dropped this tweet along with a video demo.

https://twitter.com/jonh4t/status/1429049506021138437

Razer's Response

We were made aware of a situation in which our software, in a very specific use case, provides a user with broader access to their machine during the installation process.

We have investigated the issue, are currently making changes to the installation application to limit this use case, and will release an updated version shortly. The use of our software (including the installation application) does not provide unauthorized third-party access to the machine.

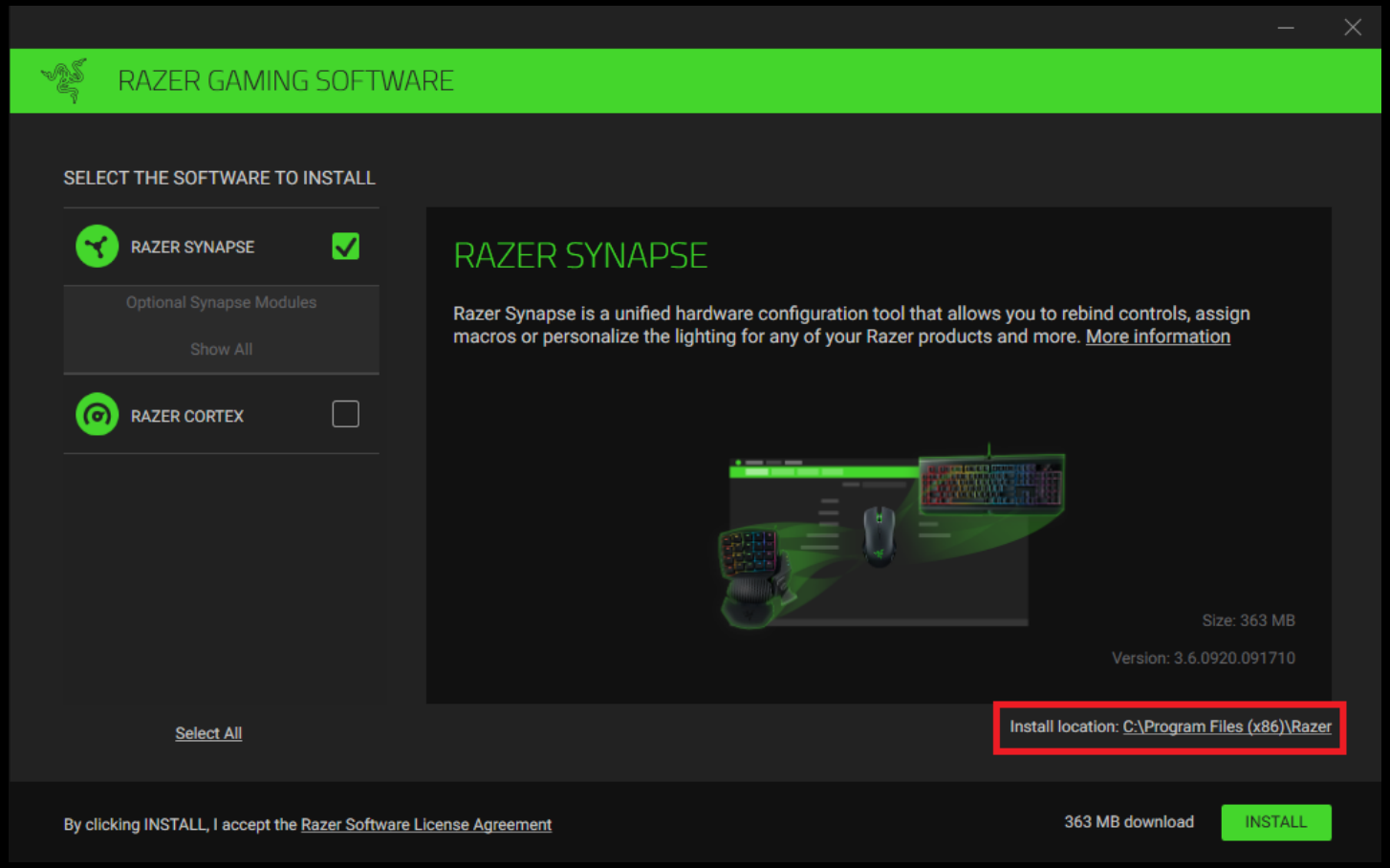
We are committed to ensuring the digital safety and security of all our systems and services, and should you come across any potential lapses, we encourage you to report them through our bug bounty service, Inspectiv: https://app.inspectiv.com/#/sign-up.

Probably Aug 23. Can't find press release, only quotes.

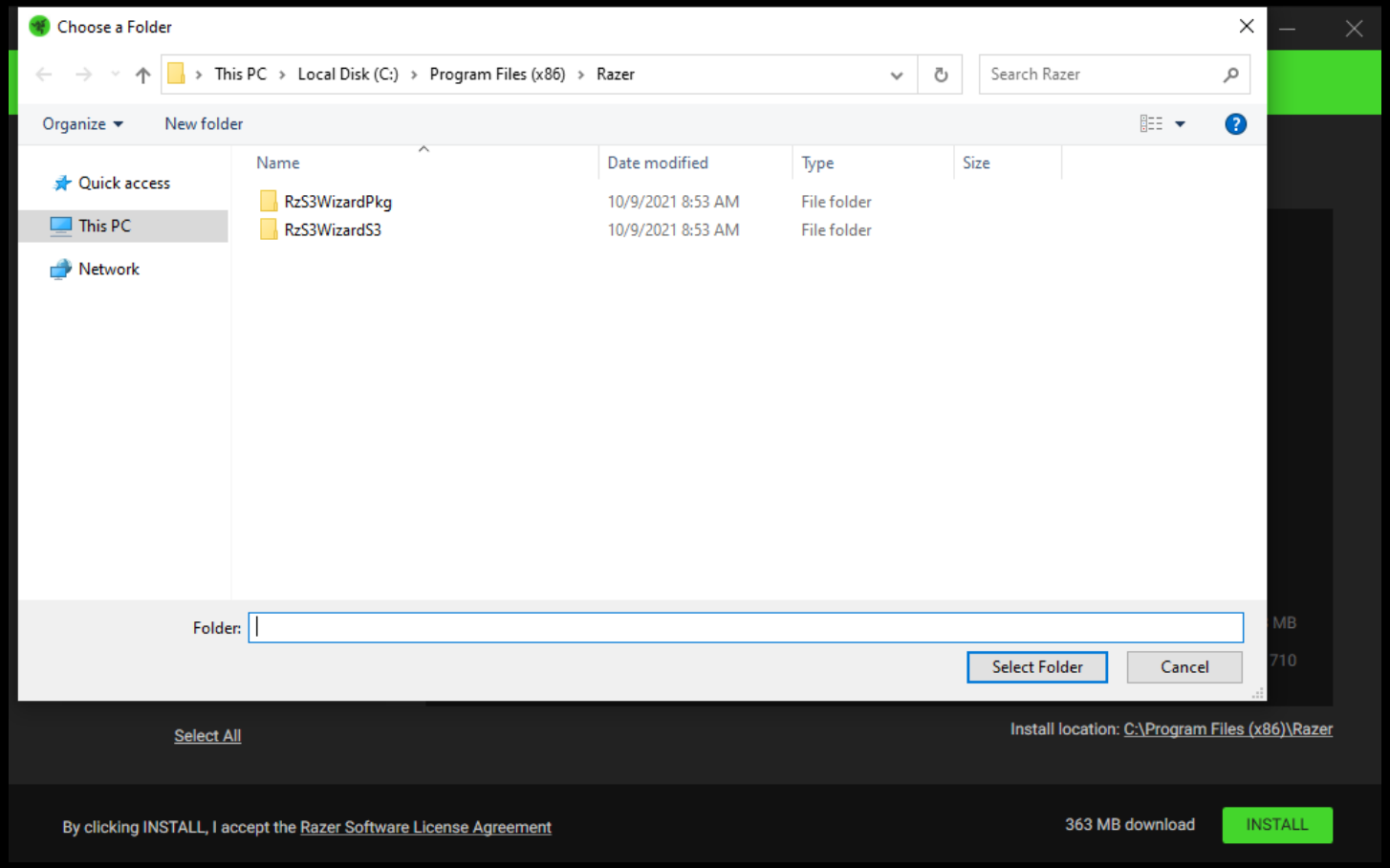
How does it work?

1. Plug in the Razer device.
2. Windows recognizes it as a Razer Mouse.
3. Windows downloads the driver package.
4. The driver package contains an installer executable which is automatically executed.
5. The installer has a "Choose Install Location" link.
6. Clicking the link opens a Windows Explorer window.
7. Navigate to, or type the path to your desired executable.
8. It runs as the current user, nt authority/system.

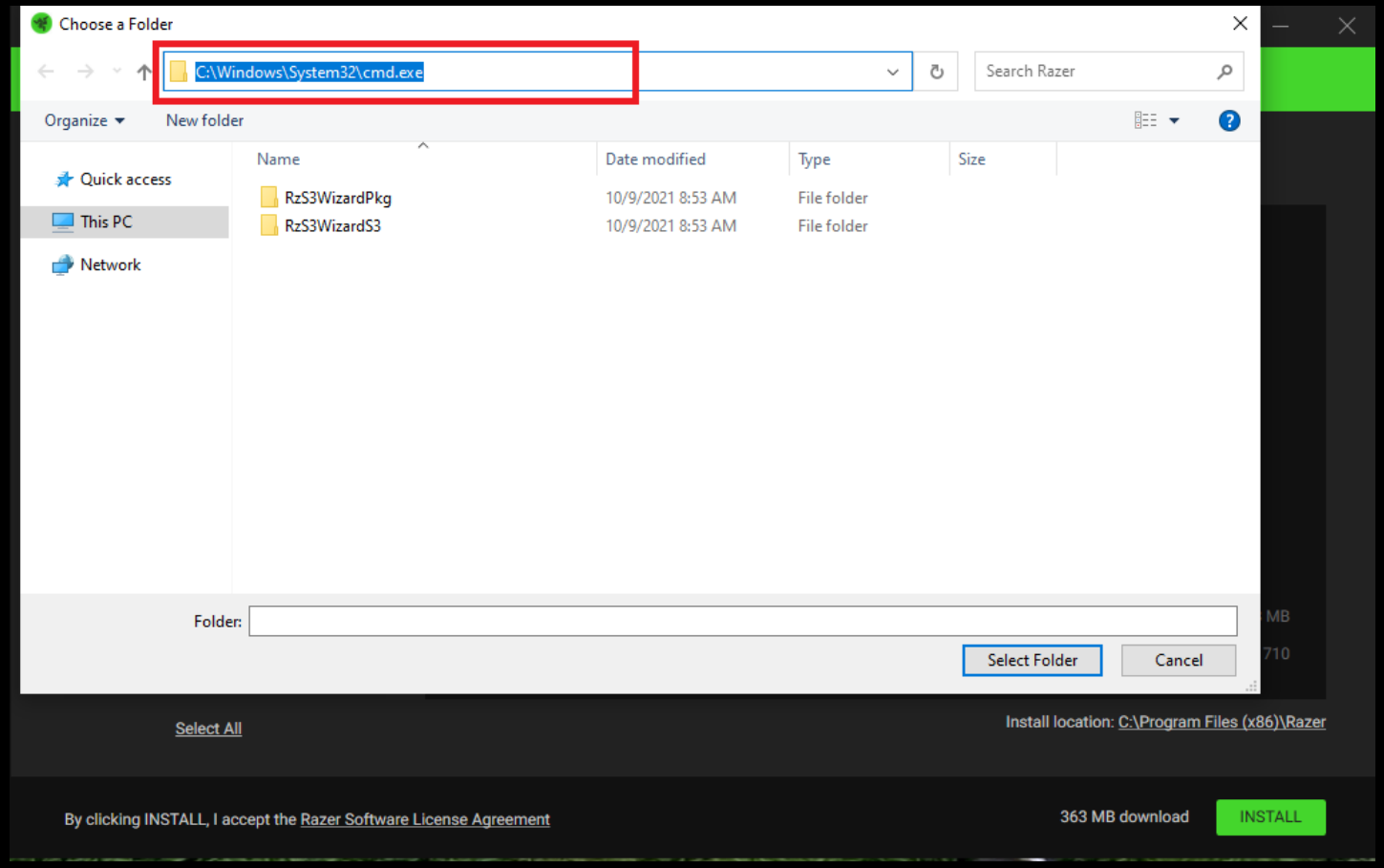
Exploit – Step 1



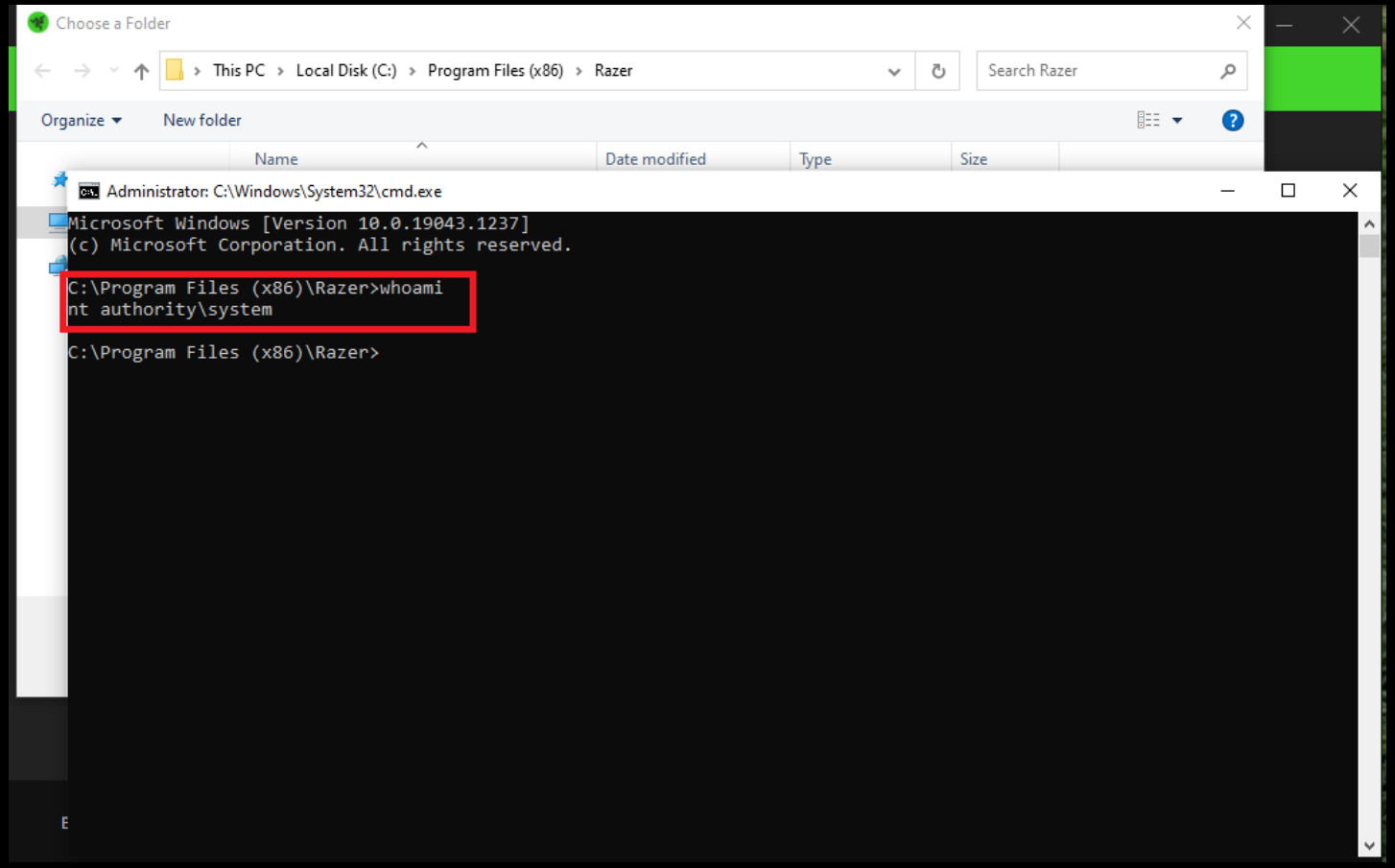
Exploit – Step 2



Exploit – Step 3



Exploit – Step 4



One time use?

If anyone has plugged in a similar device, the driver will be on the system, so the installer will not be executed.

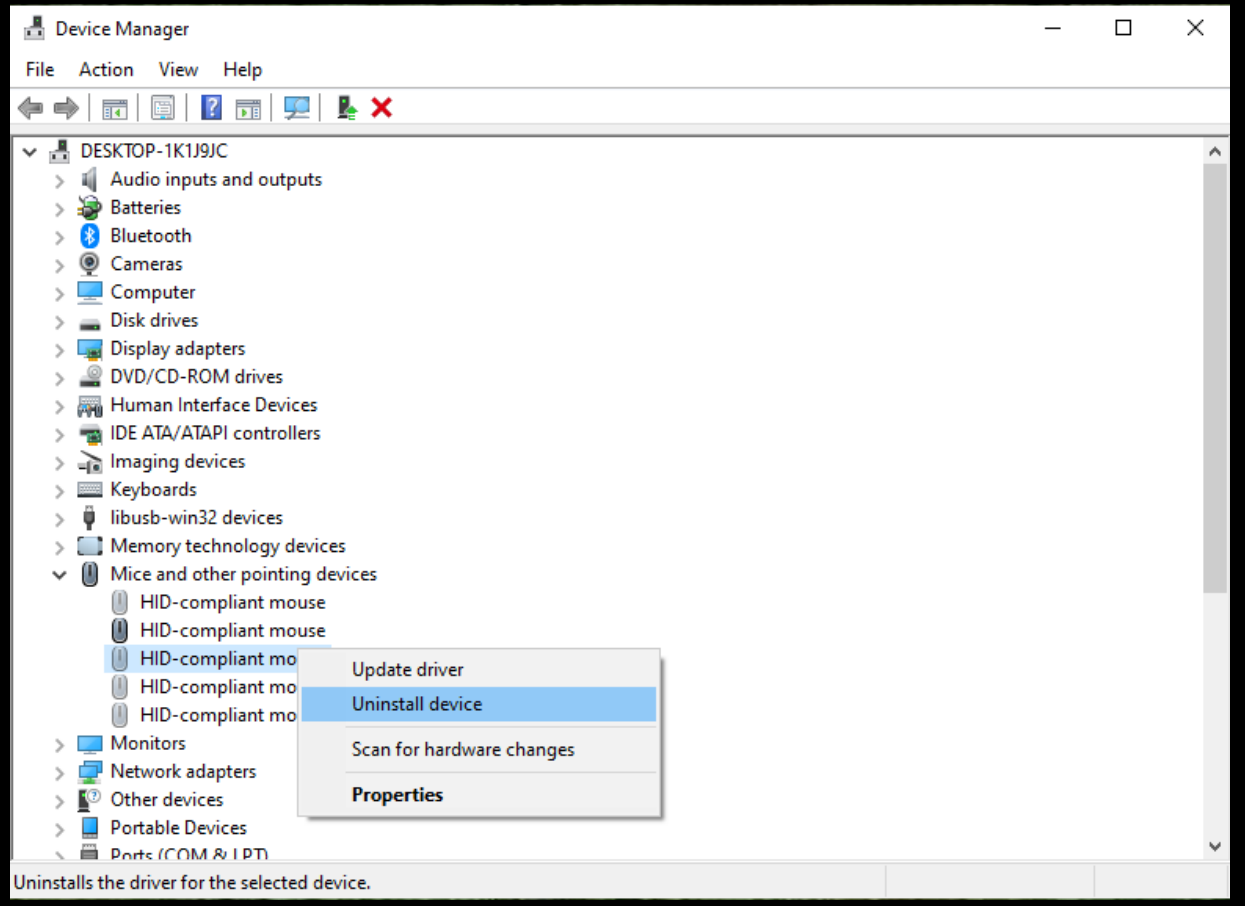
But regular users can reset…

1. Open Device Manager
2. Find the device by plugging it in.
3. Right click –> Uninstall Device

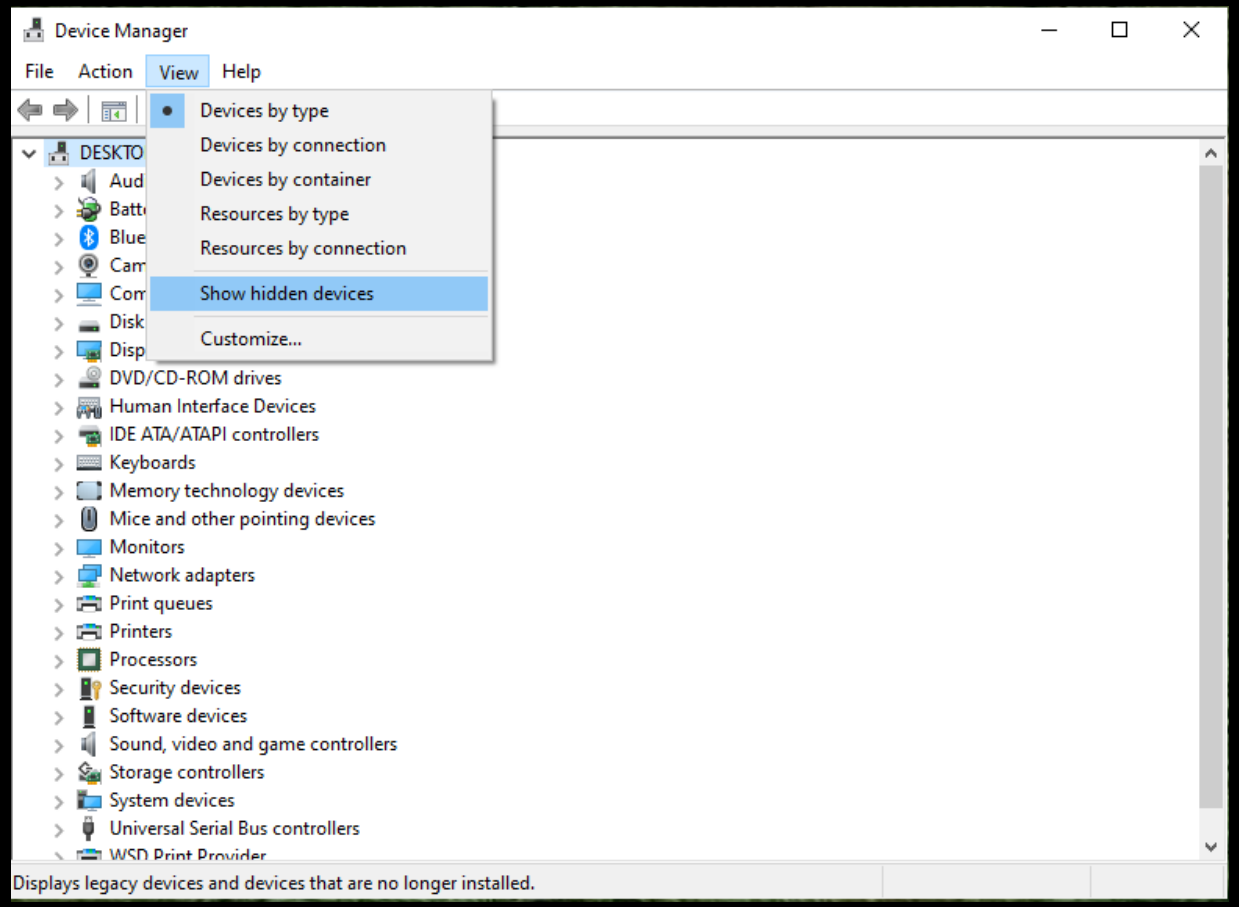
Sometimes useful to show devices that aren't plugged in:

- In Device Manager: View -> Show hidden devices
- Right click on an unknown device
  - Click on the "Events" tab (or Properties)
  - USB vendor/device ID will be displayed

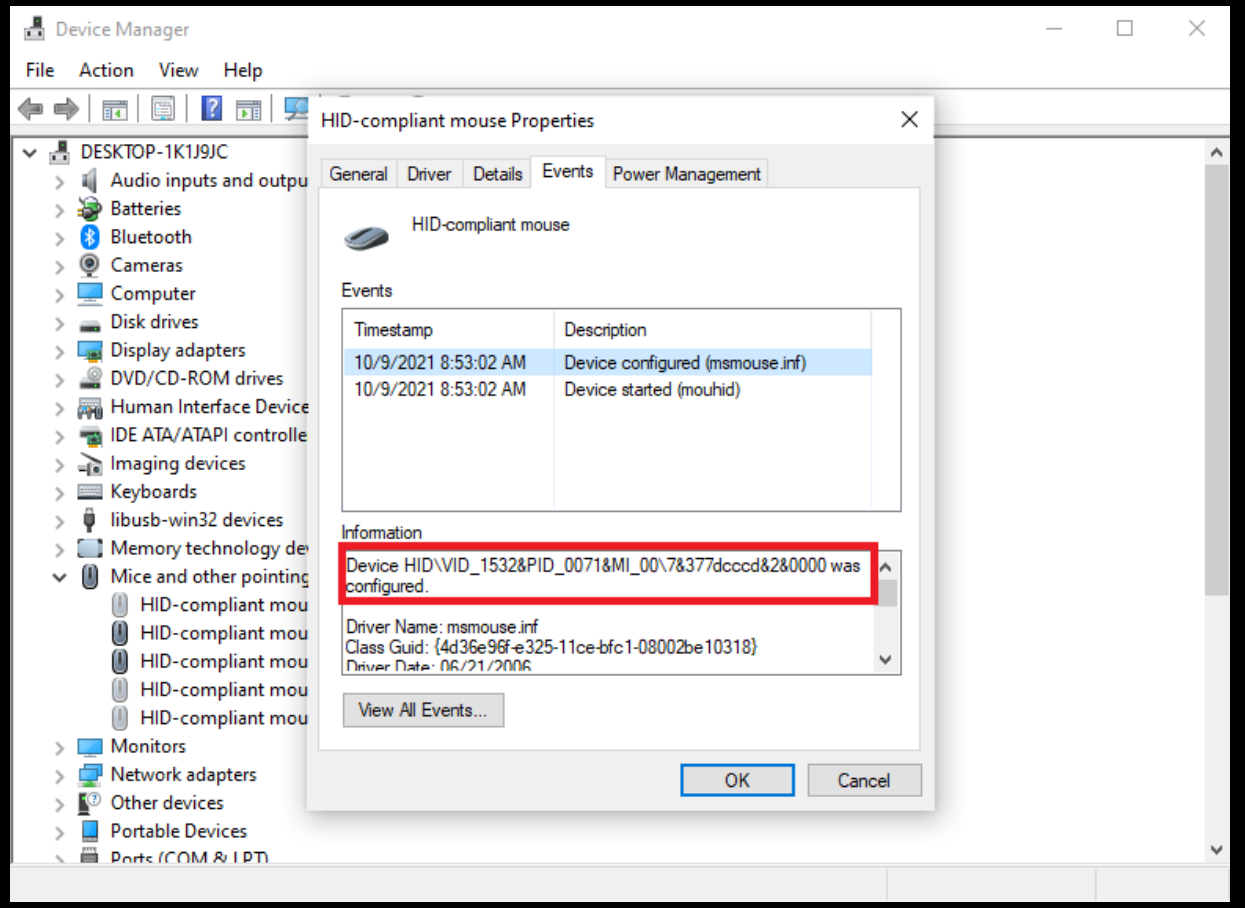
Reset – Step 1


Device Manager
File
Action
View
Help
DESKTOP-1K1J9JC
Audio inputs and outputs
Batteries
Bluetooth
Cameras
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Human Interface Devices
IDE ATA/ATAPI controllers
Imaging devices
Keyboards
libusb-win32 devices
Memory technology devices
Mice and other pointing devices
HID-compliant mouse
HID-compliant mouse
HID-compliant mo
HID-compliant mo
HID-compliant mo
Monitors
Network adapters
Other devices
Portable Devices
Ports (COM & LPT)
Update driver
Uninstall device
Scan for hardware changes
Properties
Uninstalls the driver for the selected device.

Reset – Show Devices not Present



Reset – Identify Device



Not just Razer

- SteelSeries Keyboards, etc.
  - https://0xsp.com/security%20research%20&%20development%20(SRD)/local-administrator-is-not-just-with-razer-it-is-possible-for-all
    - This site has other interesting resources (payloads, escalation paths, etc.)
  - Instead of changing install location, click on "Learn More" link in EULA.
    - Opens IE running as system.
    - File -> Save As -> Explorer window (one of many exploit paths).
- Vendor Response

  "We are aware of the issue identified and have proactively disabled the launch of the SteelSeries installer that is triggered when a new SteelSeries device is plugged in. This immediately removes the opportunity for an exploit and we are working on a software update that will address the issue permanently and be released soon."

  PATCHED – August 28

The Blame Game

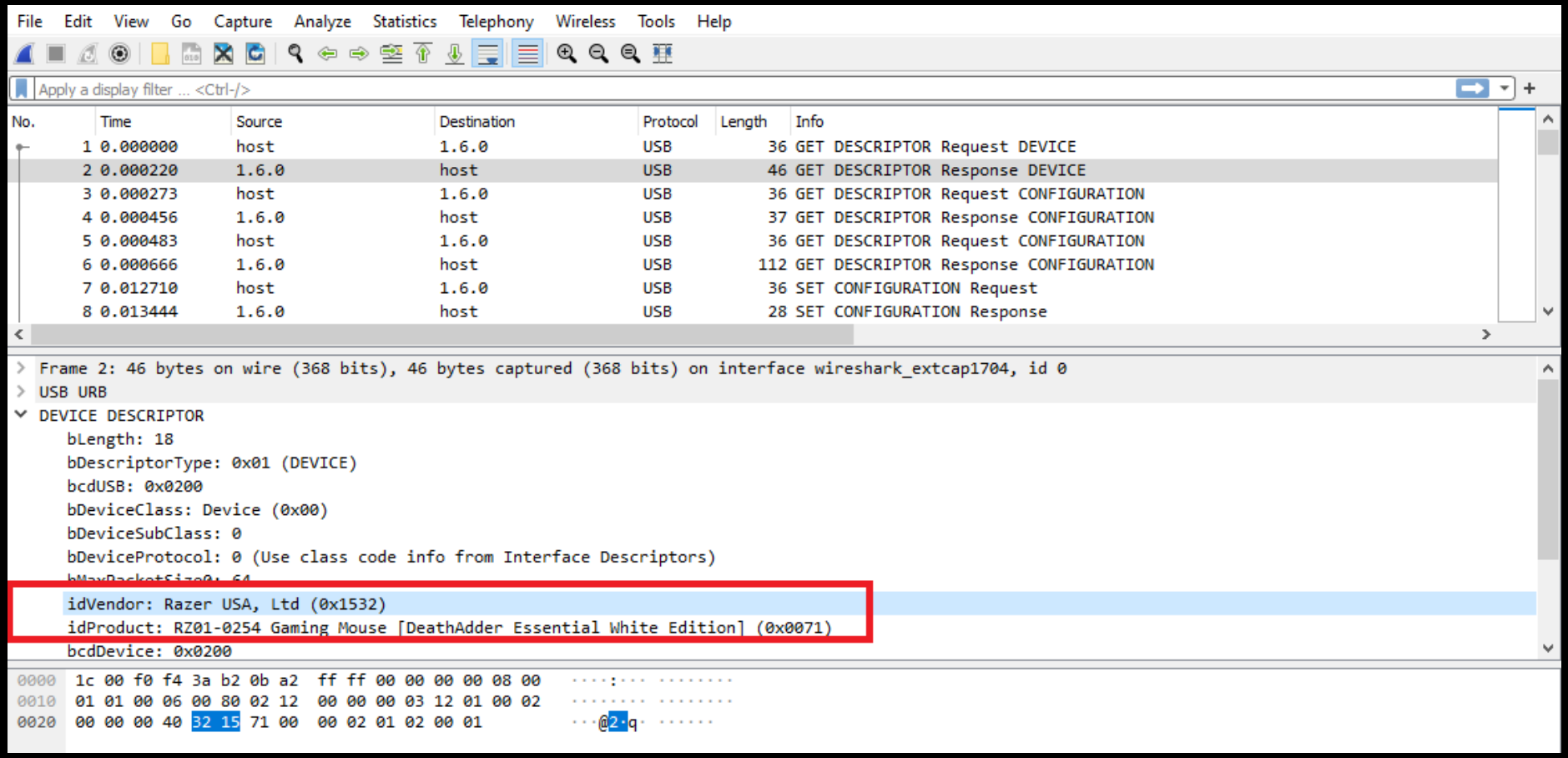
- WINDOWS runs driver installers as system.
  - Drivers run with low level privileges on most (all?) operating systems to grant hardware access and performance benefits.
  - The installer "probably" has to have the ability to write to protected locations.
- VENDORS bundle extra software with their driver.
  - Provides "enhanced user experience".
  - Installing with the driver is the most natural interaction.
  - Installation process uses built in Windows components.
- WINDOWS components were not designed to prevent lateral movement.

The Driver Installation Process

Driver Installation Overview

1. Plug in a USB Device
2. The USB protocol gathers the devices vendor ID, device ID, and other information such as the number of interfaces
3. Windows compares that information to known device definitions (.inf files)
4. If no matches are found locally, the Windows Driver Store is consulted.
5. Assuming a match is found, the driver file is downloaded.
6. The installation process follows the steps defined in the .inf file. The driver file may contain supporting resources (images, executables, DLLS, etc.) as well.

USB Device Identification



INF Files

- VERY robust file format providing a wide array of features
  - USB, PCI, PCMCIA, BlueTooth, Other device identifiers
  - Can copy files, execute files, add registry keys, install services, etc.
  - More Information at: https://docs.microsoft.com/en-us/windows-hardware/drivers/install/overview-of-inf-files
- Each file contains a section identifying the device it applies to.
- Local library of INF files located at C:\Windows\INF
  - This directory is Read-Only except for Administrators, System, "Trusted Installer", and "Creator Owner"

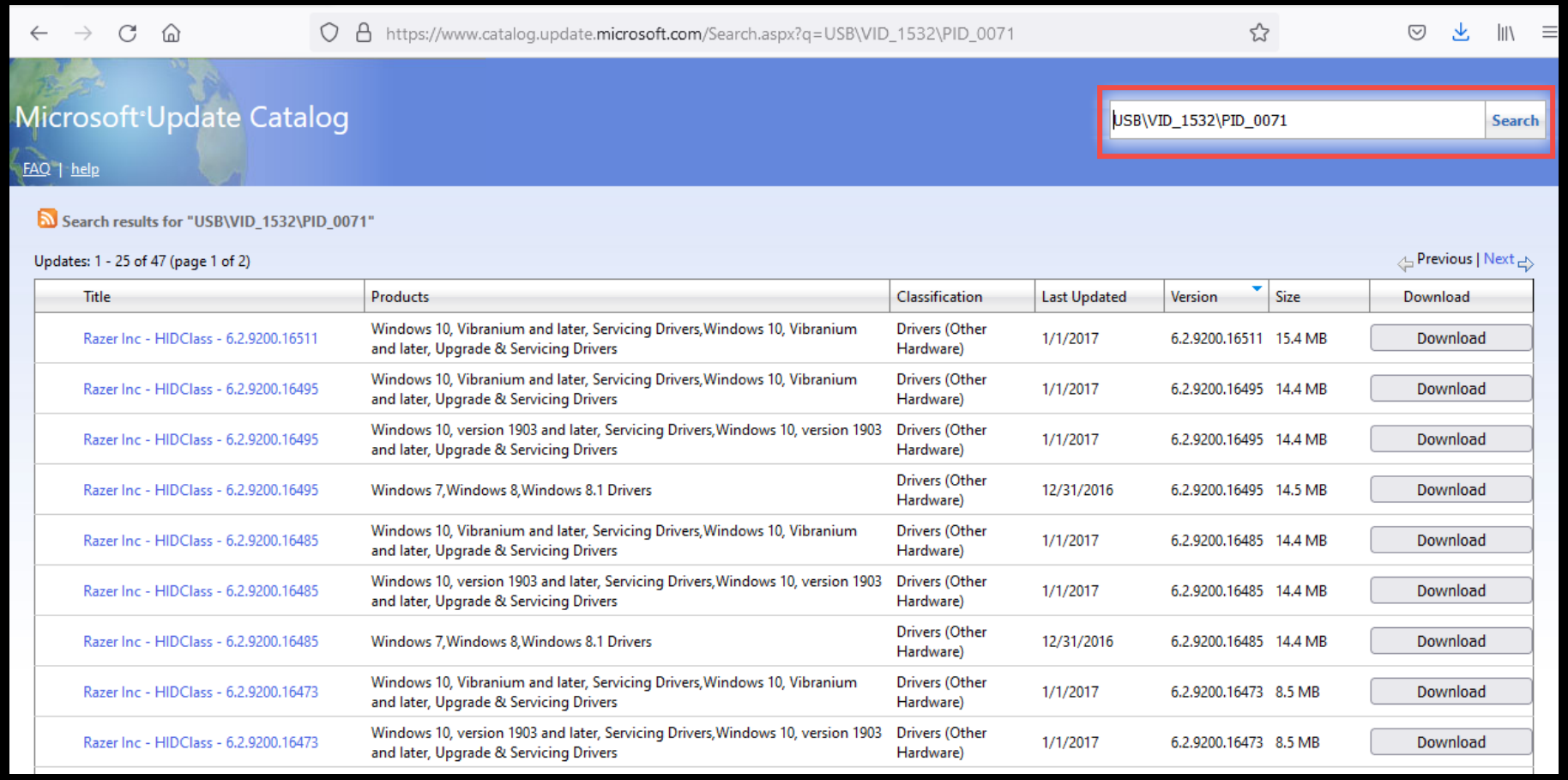
INF File Example

```
; Copyright: (c) 2018 Razer Inc. All rights reserved

[Version]
Signature   = "$WINDOWS NT$"
Class       = HIDClass
ClassGuid   = {745a17a0-74d3-11d0-b6fe-00a0c90f57da}
Provider    = %Razer%
DriverVer=01/02/2017,6.2.9200.16503
CatalogFile = rz0070dev.cat

[DestinationDirs]
Razer_CoInstaller_CopyFiles = 11
Razer_Installer_CopyFilesWOW64 = 16426,"Razer\RzS3WizardPkg"

[Manufacturer]
%Razer%=Standard,NTAMD64

[Standard.NTAMD64]
%Razer0070.DeviceDesc%=Razer, USB\Vid_1532&Pid_0070&MI_02

[Razer.NTAMD64]
Include = input.inf
Needs = HID_Inst.NT
CopyFiles = Razer_Installer_CopyFilesWOW64
```

Windows Driver Store

- If a matching driver cannot be found in local INF files, Windows will query the "Driver Store"
- The Windows Catalog site provides a human interface into a similar, but not identical, repository.
  - https://www.catalog.update.microsoft.com/
- Queries can be made on vendor name, device type, and USB string to locate drivers.
- Multiple versions of the driver are available (for manual installation), but in our case, we're only interested in the latest one since that is the one that would be triggered when the device is plugged in.

Microsoft Update Catalog



Title	Products	Classification	Last Updated	Version	Size	Download
Razer Inc - HIDClass - 6.2.9200.16511	Windows 10, Vibranium and later, Servicing Drivers,Windows 10, Vibranium and later, Upgrade & Servicing Drivers	Drivers (Other Hardware)	1/1/2017	6.2.9200.16511	15.4 MB	Download
Razer Inc - HIDClass - 6.2.9200.16495	Windows 10, Vibranium and later, Servicing Drivers,Windows 10, Vibranium and later, Upgrade & Servicing Drivers	Drivers (Other Hardware)	1/1/2017	6.2.9200.16495	14.4 MB	Download
Razer Inc - HIDClass - 6.2.9200.16495	Windows 10, version 1903 and later, Servicing Drivers,Windows 10, version 1903 and later, Upgrade & Servicing Drivers	Drivers (Other Hardware)	1/1/2017	6.2.9200.16495	14.4 MB	Download
Razer Inc - HIDClass - 6.2.9200.16495	Windows 7,Windows 8,Windows 8.1 Drivers	Drivers (Other Hardware)	12/31/2016	6.2.9200.16495	14.5 MB	Download
Razer Inc - HIDClass - 6.2.9200.16485	Windows 10, Vibranium and later, Servicing Drivers,Windows 10, Vibranium and later, Upgrade & Servicing Drivers	Drivers (Other Hardware)	1/1/2017	6.2.9200.16485	14.4 MB	Download
Razer Inc - HIDClass - 6.2.9200.16485	Windows 10, version 1903 and later, Servicing Drivers,Windows 10, version 1903 and later, Upgrade & Servicing Drivers	Drivers (Other Hardware)	1/1/2017	6.2.9200.16485	14.4 MB	Download
Razer Inc - HIDClass - 6.2.9200.16485	Windows 7,Windows 8,Windows 8.1 Drivers	Drivers (Other Hardware)	12/31/2016	6.2.9200.16485	14.4 MB	Download
Razer Inc - HIDClass - 6.2.9200.16473	Windows 10, Vibranium and later, Servicing Drivers,Windows 10, Vibranium and later, Upgrade & Servicing Drivers	Drivers (Other Hardware)	1/1/2017	6.2.9200.16473	8.5 MB	Download
Razer Inc - HIDClass - 6.2.9200.16473	Windows 10, version 1903 and later, Servicing Drivers,Windows 10, version 1903 and later, Upgrade & Servicing Drivers	Drivers (Other Hardware)	1/1/2017	6.2.9200.16473	8.5 MB	Download

Unpacking the Driver

- Drivers are download as cabinet (.cab) files
- Cab files can be accessed using the built-in command line tool "expand".
- To list the contents of a cab file

```
expand /d [filename].cab
```

- To extract the contents of a cab file to a directory

```
expand -F:* [filename].cab [destination directory]
```

Driver Installation

- Once the matching driver has been downloaded and installed, Windows will parse the .inf file and execute the steps defined.
- In the case of the Razer driver, this includes running the executable bundled in the driver package named `"RzS3WizardPkgS3.exe"`
  - This is the installer that contains the vulnerable "change location" link.

INF File Example (Part 2)

```
[SourceDisksNames]
1 = %DiskId1%,,,""

[SourceDisksFiles]
RazerS3Coinstaller.dll=1,,
RzS3WizardPkgS3.exe = 1,,

[Razer.NTAMD64.CoInstallers]
AddReg          = Razer_CoInstaller_AddReg
CopyFiles       = Razer_CoInstaller_CopyFiles

[Razer_CoInstaller_AddReg]
HKR,,CoInstallers32,0x00010000, "RazerS3Coinstaller.dll,RazerCoinstaller"

[Razer_CoInstaller_CopyFiles]
RazerS3Coinstaller.dll

[Razer_Installer_CopyFilesWOW64]
RzS3WizardPkgS3.exe

[Strings]
Razer                   = "Razer Inc"
DiskId1                 = "Razer Installer"
Razer.SvcDesc           = "Razer Device Driver"

Razer0070.DeviceDesc    = "Razer Lancehead Wireless"
```

Things that make you go hmm…

Are there more drivers out there with vulnerable installers?

- Yes.
  - Not because I know of any, but the installation pattern is just too common for there not to be.
- Research Path
  1. Browsing the Windows Catalog Site
  2. Download a driver
  3. Unpack the driver
  4. Look for bundled executables and / or suspicious actions in the inf file.
  5. Emulate the appropriate device to trigger the install.
  6. Profit?

Would it be possible to emulate a USB device without hardware?

- It looks like it.
- Windows 10 supports emulated USB devices using the Windows Driver Kit
  - https://docs.microsoft.com/en-us/windows-hardware/drivers/usbcon/developing-windows-drivers-for-emulated-usb-host-controllers-and-devices
  - https://developer.microsoft.com/en-us/windows/hardware/
- The Windows Device Testing Framework is designed to support unit tests for drivers and hardware.
  - This might include functionality for emulation.
  - https://docs.microsoft.com/en-us/windows-hardware/drivers/wdtf/

Could drivers be abused in other ways?

- Very likely.
- The communication between a device and its driver would be "trusted" in the mind of a developer and very prone to "happy path" programming.
  - If a device could be emulated far enough to submit data to the driver, the driver's data handling processes could be attacked.
  - Since the driver is running as System, a buffer overflow or other remote code execution vulnerability would have the same permissions.
- Drivers sometimes automatically install Services as well.
  - These services may accept input both from the device and from other protocols (e.g. RPC, network)
  - These services are likely not as well tested as a built-in Windows service

Could non-USB devices have similar vulnerabilities?

- Yes.
- The INF file format supports device identification through at least the following protocols:
  - USB
  - Bluetooth
  - PCI
  - PCMCIA
  - SWC – Software Component?
- If there were a network/radio protocol that could trigger driver installation, it might be possible to upgrade the Privilege Escalation bug to Remote Code Execution.

Side Note: PrintNightmare

- PrintNightmare was more of an RPC exploit than a driver exploit.
  - Print Spooler exposes two RPC calls that end up using the same vulnerable code: RpcAddPrinterDriverEx and RpcAsyncAddPrinterDriver
  - The permission check on these calls can be short circuited by sending a specific flag in the RPC request.
  - The behavior of the request provides primitives to copy a file to a known location and then load it as a config file.
  - The call is used twice. First to upload an attacker controlled DLL and then to load it into Print Spooler as a config file.
  - Since Print Spooler runs as system, the entry functions of the DLL will be executed with those permissions.
- Good writeups at:
  - https://hidocohen.medium.com/understanding-printnightmare-vulnerability-cf4f1e0e506c
  - https://www.sygnia.co/demystifying-the-printnightmare-vulnerability

Side Note: EvilPrinter

- An attack against the printer driver validation process.
  - Authenticated user triggers the installation of a printer and specifies an attacker controlled location.
  - The attacker serves a malicious printer driver package.
  - The package may contain a DLL that is not part of the signed driver content.
  - That DLL will be loaded as part of the install process and may contain arbitrary code.
- Good writeups at:
  - https://media.defcon.org/DEF%20CON%2028/DEF%20CON%20Safe%20Mode%20presentations/DEF%20CON%20Safe%20Mode%20-%20Zhipeng-Huo%20and%20Chuanda-Ding%20-%20Evil%20Printer%20How%20to%20Hack%20Windows%20Machines%20with%20Printing%20Protocol.pdf
  - Recorded Talk – Defcon 29 – Jacob Baines – Brind your Own Print Driver Vulnerability

Emulating a USB Device

Method 1: Android Device

Pre-requisites

- You must have root privileges on the Android device
- Kernel MUST be built with USB ConfigFS support
  - Based on my experience, most are NOT
- Otherwise the emulation can be configured both through Apps and via shell script.
  - Kali NetHunter has the capability to execute this attack if your device does.

USB Gadget Instructions

1. Download the emulation script
   - https://github.com/tothi/usbgadget-tool
   - https://raw.githubusercontent.com/tothi/usbgadget-tool/master/usbgadget-tool.sh
2. Load the script onto a rooted Android device in an executable directory
   - adb push usbgadget-tool.sh /data/local/tmp
   - OR wget the file to this location directly
3. From a root shell, run usbgadget-tool.sh
4. Select option 1
5. Plug phone into a computer

Emulating a USB Device

Method 2: Digispark Digistump (ATtiny85)

Overview

- Digispark created a device (that many have cloned) to server as a USB development board using an ATtiny85 microcontroller.
- A low level USB driver for this chip has existed for a while – VUSB
  - https://www.obdev.at/products/vusb/index.html
- Digispark built Arduino IDE compatible libraries around this code.
- The Digistump makes programming easy
  - When you first plug it in it acts as a programmable microcontroller.
  - After five seconds, it runs the program that has been flashed to it.
  - The new program can be a USB emulator and Windows will detect it as a new device.

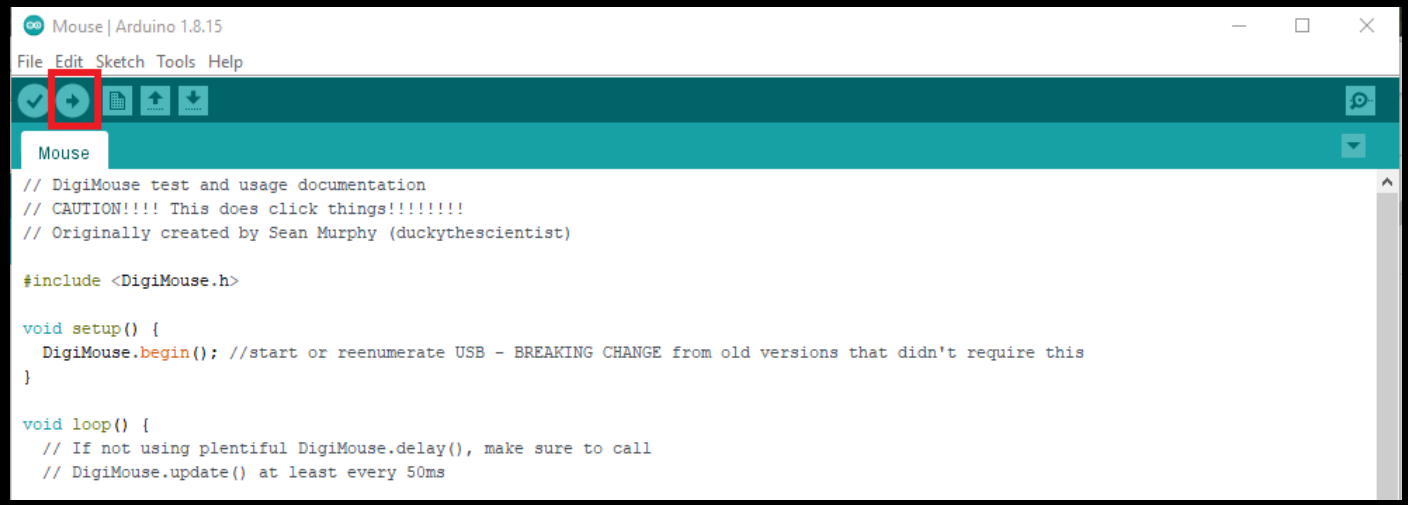
Setting up the Arduino IDE

1. Install the latest version of the Arduino IDE and run the IDE
   - https://www.arduino.cc/en/software
2. Open Application Preferences (File -> Preferences)
3. In the input field named “Additional Boards Manager URLs” enter the following URL:
   - http://digistump.com/package_digistump_index.json
4. Download support for Digistump
   1. Go to Tools -> Board -> Boards Manager
   2. From the Type drop-down menu select "Contributed"
   3. Find and click Digistump AVR Boards
5. In Tools -> Board, select Digispark (Default – 16.5mhz)

https://maker.pro/arduino/projects/how-to-build-a-rubber-ducky-usb-with-arduino-using-a-digispark-module

First Digistump Project

- Under File -> Examples -> Examples for Digispark 16.5Mhz
  - Select "DigisparkMouse -> Mouse
- A new project named "Mouse" should appear.
- To make sure things are working click the "Upload" button, wait for compilation to complete, then plug in your Digistump.
  - After programming (5 seconds), you should hear the device reload and your mouse will start moving around.



Find the driver

- V-USB provides a way to tinker with the vendor and device ID
  - Digispark/Arduino do not.
  - We're going to modify the DigiMouse code.
- Locate the Digistump Mouse library code:
  - %localappdata%\Arduino15\packages\digistump\hardware\avr\1.6.7\libraries\DigisparkMouse
- Make a backup of this director just in case.

Hack the Driver

- Open usbconfig.h
- Change the Vendor ID on line 224 to read

```
#define USB_CFG_VENDOR_ID 0x32, 0x15
```

  - This is Razer's vendor ID
  - Note the reverse byte order. So "1532" from USB docs becomes 0x32,0x15
- Change the Device ID on line 233 to read

```
#define USB_CFG_DEVICE_ID 0x21, 0x0f
```

  - This is the device ID for the ChromaDock

Simplify the Sketch

- We don't actually want any mouse movement so strip the code down to the following:

```
void setup() {
  // put your setup code here, to run once:
  DigiMouse.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  DigiMouse.delay(500);
}
```

Try it out

- Press the "upload" button again.
- Plug in the Digistump
- Wait 5 seconds while it loads.
- You should hear the device disconnect and reconnect.
- Wait 20-60 seconds for the driver to download and load
- You should see the Razer install screen.