



not-powershell

detecting the *#not-powershell* gang

Mangatas Tondang (Tas)
@tas_kmanager



not-powershell

Get-ADUser -Identity @tas_kmanager

Mangatas Tondang (Tas) Threat Hunter

@ Major Canadian Telco Company

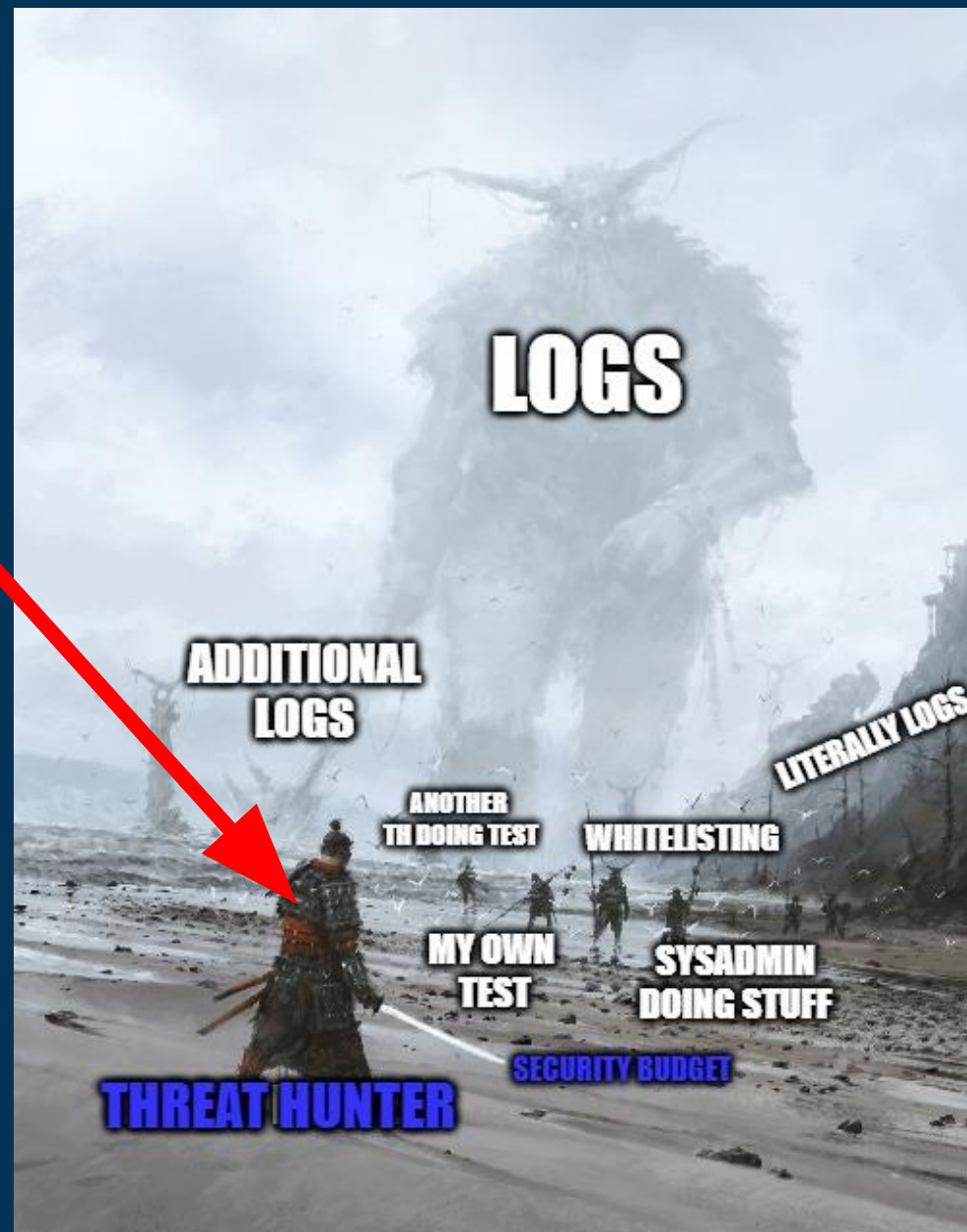
- Threat Hunting
- Threat Intel
- Automation

Previous Experience:

- Application Security - Bank/Insurance
- Security Auditing - Health Tech

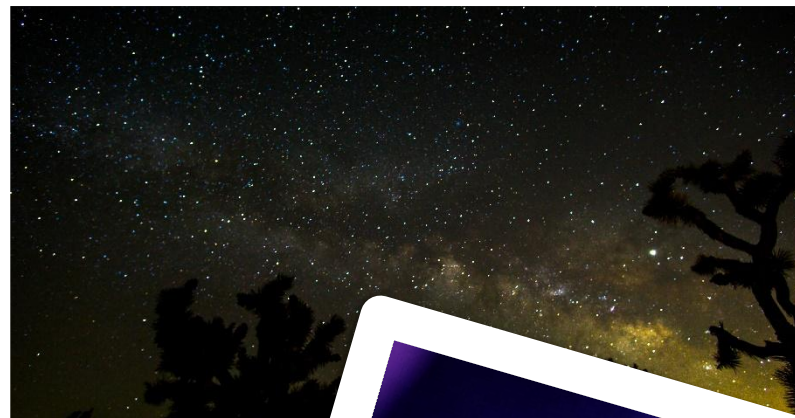
Education:

Sheridan College (Toronto)
Information Systems Security (ISS)



All work, YES PLAY!

- Fun Security Stuff
 - (Presenting, Attending, CTF playing and making)
- Photography
 - Astrophotography
 - Landscape
 - DRONEEEEEEE!
- Music
 - Guitarist at times
 - Festivals & Concerts
- Gaming
 - FPS and Strategy...
 - Also Board Game!



Disclaimer:
Opinions are my own and not the views of my employer



not-powershell

Notes

This presentation is focused more on the detection side of the not-powershell tools

If you would like to see more on the tools and some demonstration, you can watch my presentation last year at Hackfest on Hackfest youtube channel (youtu.be/00q4b12EexI)

Slide is also available on my github page (github.com/tas-kmanager)



not-powershell

Get-Content

- Intro
 - Not-PowerShell Gang
- Detecting The Gang
 - InvisiShell
 - NoPowerShell
 - PowerShDLL
 - PowerLessShell
 - Bonus
- Outro
 - Closing and Q&A



not-powershell

Intro

Not-PowerShell Gang



not-powershell

Not-PowerShell Gang?

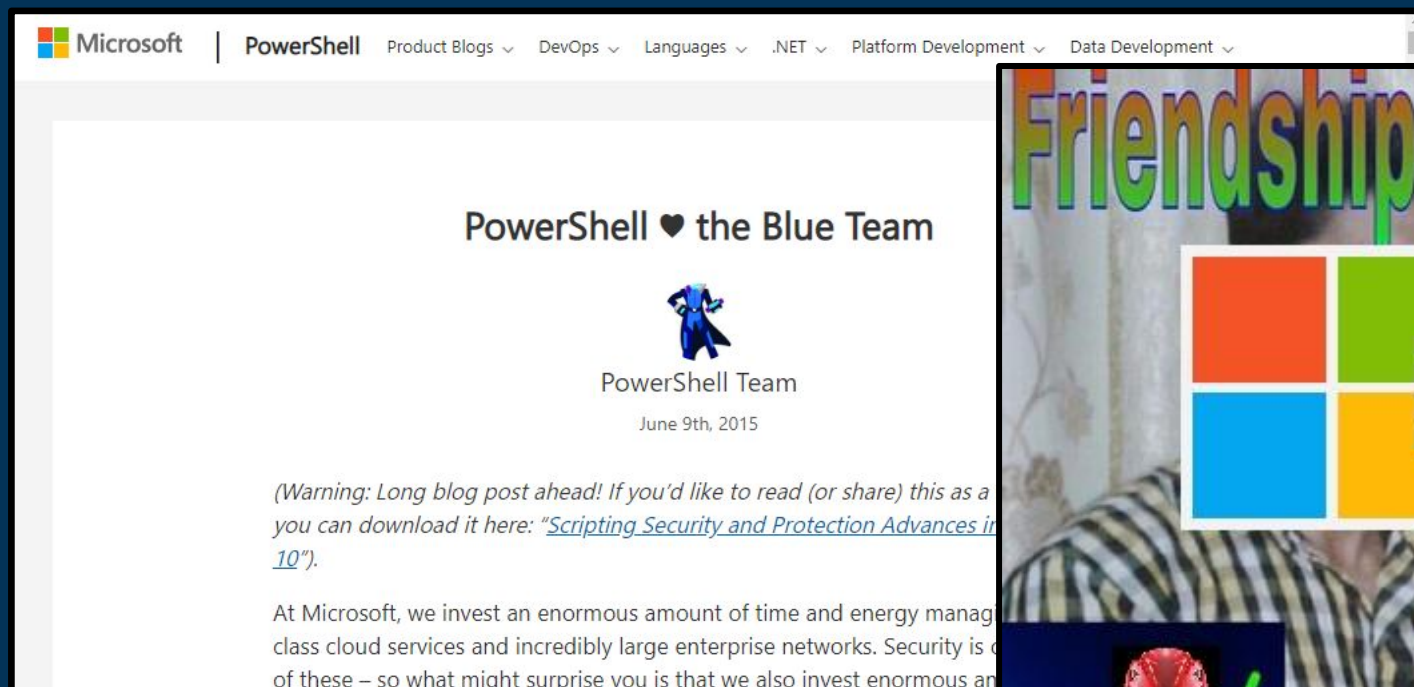
"Tools that are created to achieve PowerShell-like functionality that are able to avoid PowerShell security logging and mechanism"

- Myself



not-powershell

Because PowerShell ❤️ the Blue Team



Microsoft DevBlogs - PowerShell ❤️ the Blue Team



not-powershell

PowerShell's ❤️ for Blue Team

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

- Protected Logging (Script Block and Module Logging)
 - Log commands, decode it first if it's encoded
 - If centralized to SIEM, defender can deploy detection
- AMSI Integration
 - Never ending race between Microsoft & Red Team
 - Bypass > Patch > Bypass > Patch ...
- Constrained Language Mode (CLM)
 - Limited capability on sensitive environment



not-powershell

5 years later..

 **Tas_kManager** Apr 29th at 1:26 PM
From your recent engagements experience, can/do you still use PowerShell freely? Compared to before the release of v5 (improved logging and AMSI)

6 replies

 3 months ago
We tend to avoid PowerShell the majority of clients we test have EDR + AV + Whatever other rules on top. Normally we can operate via other means and avoid its use completely.

 3 months ago
I've almost completely dropped Powershell from my tool kit.
 1 

 3 months ago
I look at Powershell as dead. Only in rare cases that I use it... Lets say the customer forgot to remove powershell version 2....

 3 months ago
So stop monitoring powershell, got it 😊
 4  3 

 **Tas_kManager** 3 months ago
follow up questions, what tools do you use to achieve this usually? direct dll or library loadings? LOLBAS?

 3 months ago
There may still be room for amsi bypasses/downgrades but a lot of time its not worth the risk knowing if we get caught we may lose access completely.

Some comments from Red Team



not-powershell

Detecting The Gang

The Four Horsemen of Not-PowerShell

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



not-powershell

Get-Requirements

Conditions where the detections are verified working

- PowerShell version is V5
- PowerShell logging is active
- Sysmon is deployed with good configuration
- SIEM is in place and properly configured
- Alerting system is in place
 - Integrated with SIEM



SIEM Solutions

Splunk, ELK, Arcsight, LogRhythm



UTILIZE 🙌 YOUR 🙌 LOGS 🙌



not-powershell

Get-Types

2 types of detection will be provided for each tools

Low Hanging Fruits Detection

Simple Detection such as hash, file name, etc.

TTP\Behavioural Detection

Advanced detection that are the characteristics of the tools



not-powershell

Tools #1 – InvisiShell

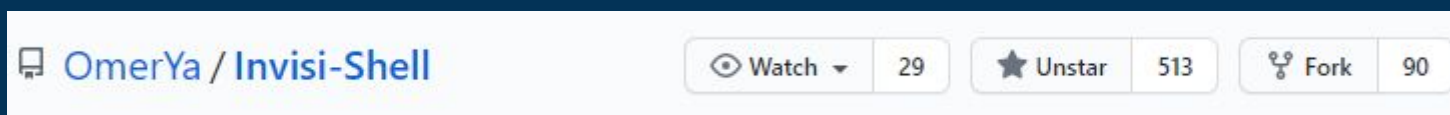
“Sure, we can hook it”



not-powershell

Tool #1 - 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

Characteristics

- 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

Hooks and overwrites the input length for the attributes above to always 0! = No PS detection



not-powershell

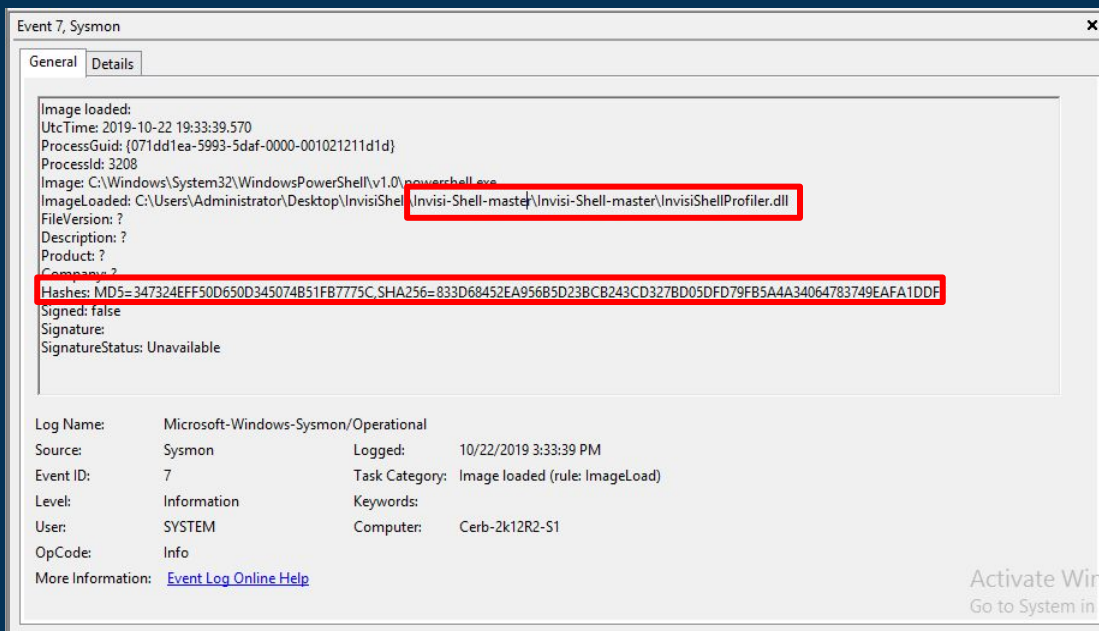
Detections

1. Low hanging fruits
2. Unique InvisiShell initialization trace
3. Reg.exe for Privilege Escalation
4. InprocServer32 Registry Key
5. Load InvisiShellProfiler.dll

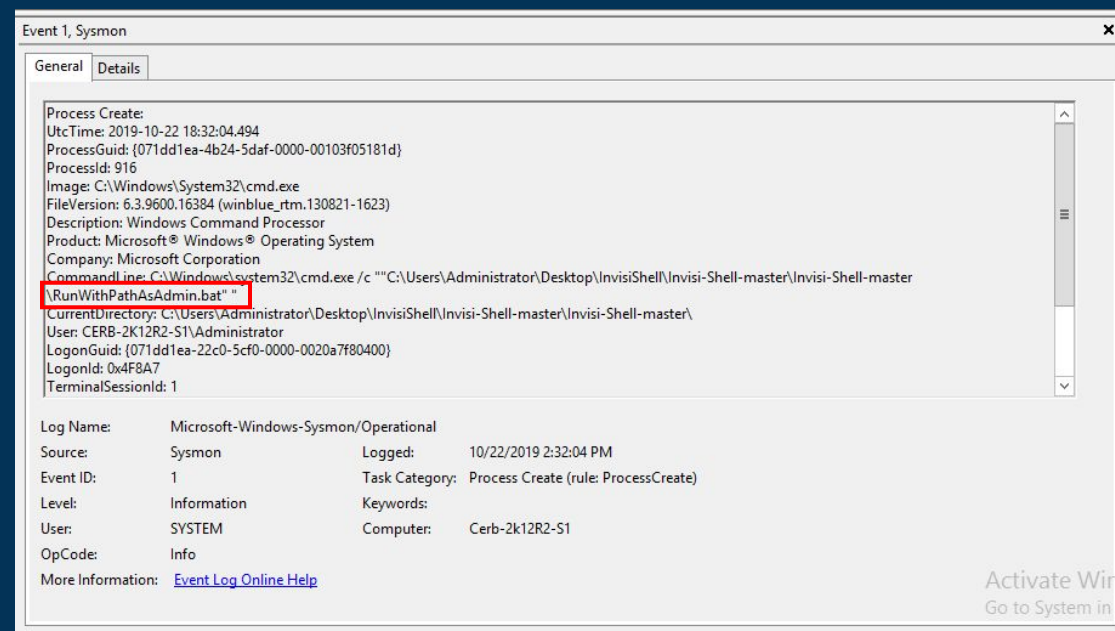


1. Low Hanging Fruits

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



Sysmon Event ID 7



Sysmon Event ID 1



not-powershell

2. Unique Initialization Trace

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

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_srvooob.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""

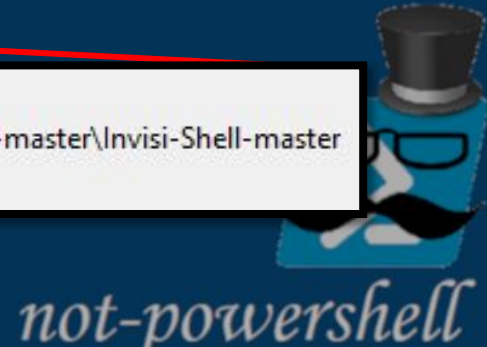
Log Name: Microsoft-Windows-Sysmon
Source: Sysmon
Event ID: 1
Level: Information
User: SYSTEM
OpCode: Info
More Information: [Event Log Online Help](#)

Computer: Cerb-2k12R2-S1

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

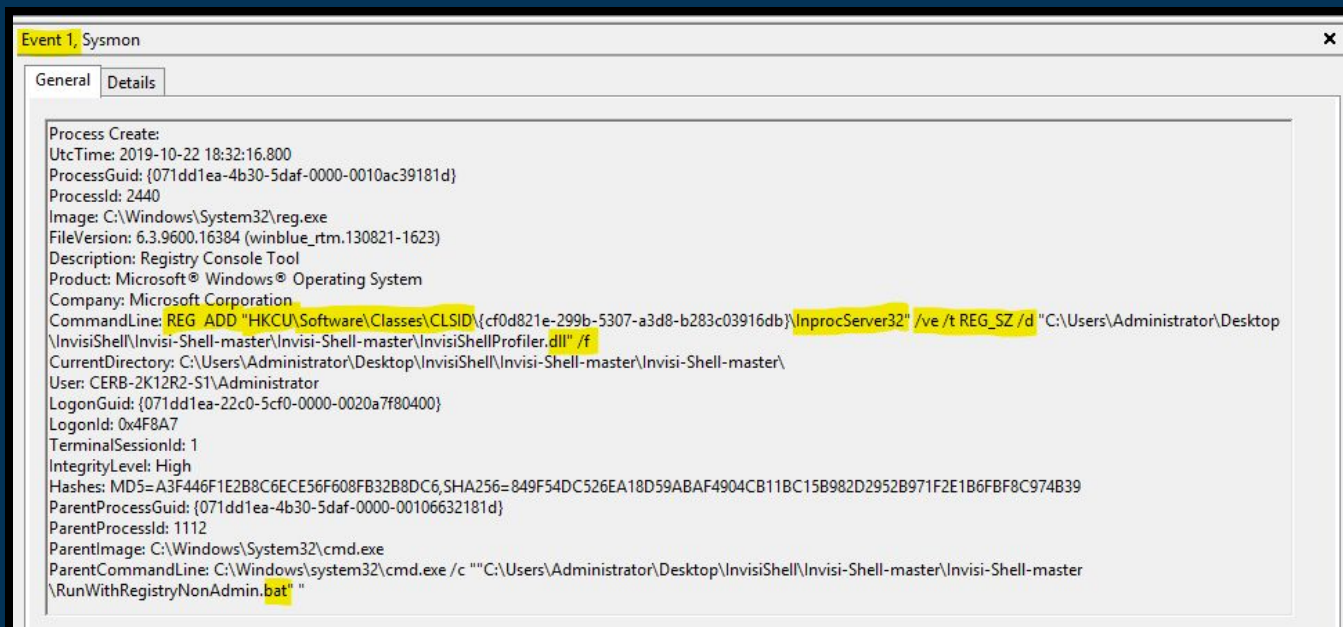
ParentProcessId: 352
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: C:\Windows\system32\cmd.exe /c ""C:\Users\Administrator\Desktop\InvisiShell\Invisi-Shell-master\Invisi-Shell-master\RunWithRegistryNonAdmin.bat""

Sysmon Event ID 1



3. Reg.exe For Priv. Escalation

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



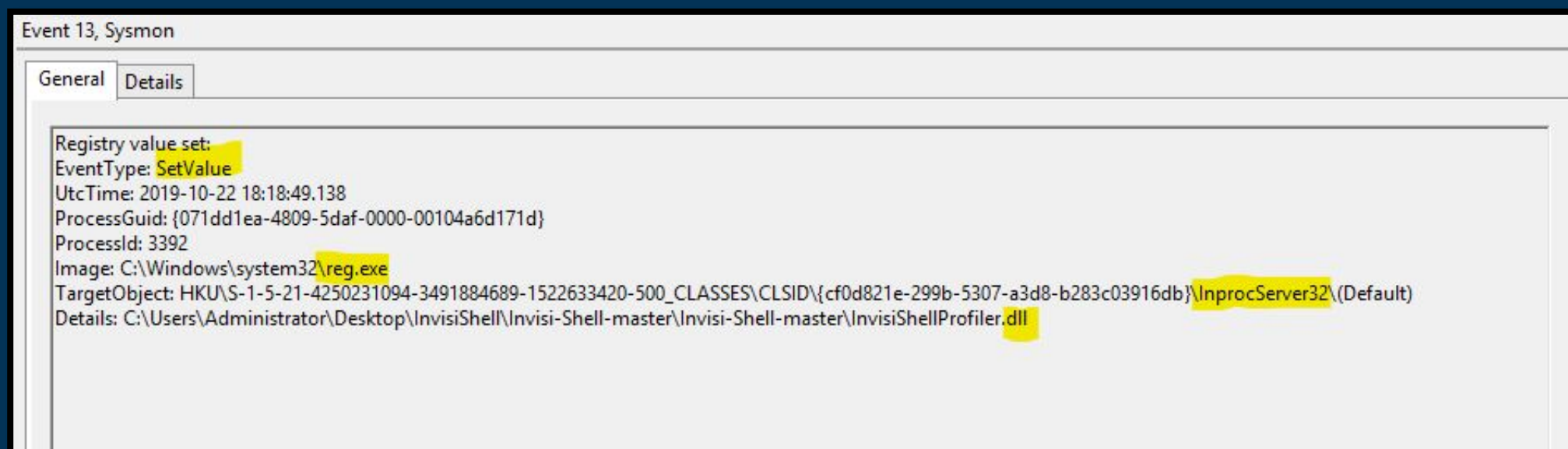
Sysmon Event ID 1



not-powershell

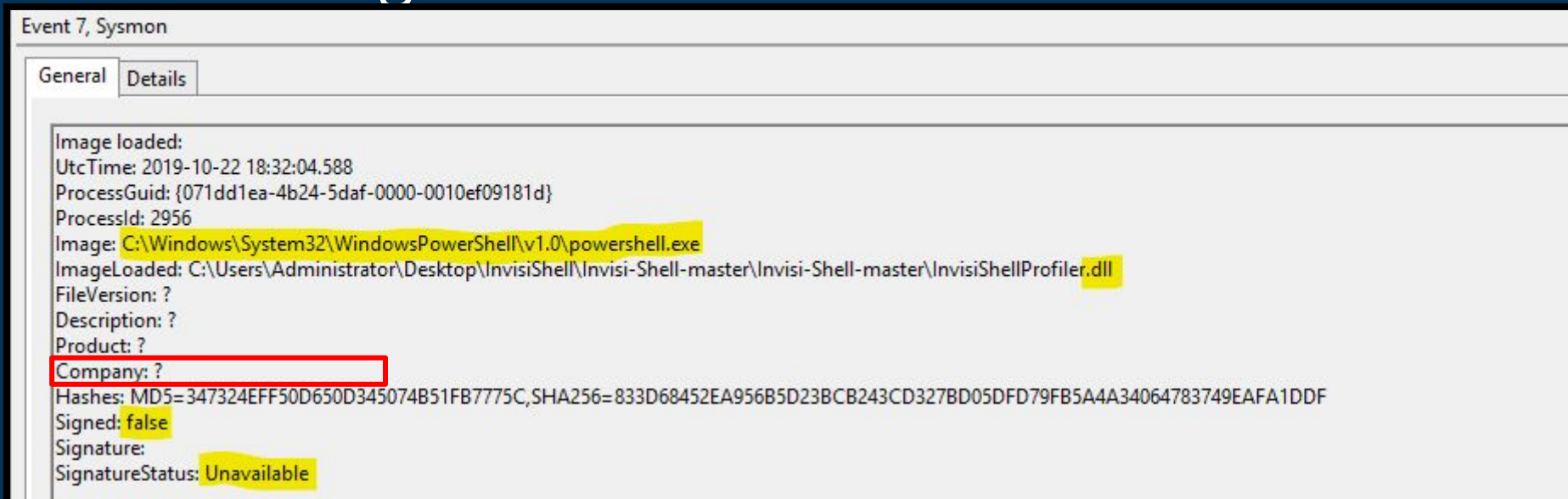
4. InprocServer32 Registry Key

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



5. Load InvisiShellProfiler.dll

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



Sysmon Event ID 7



not-powershell

Tools #2 – PowerShDLL

“Yeah, we got DLL for that”



not-powershell

Tool #2 - PowerShDLL

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



not-powershell

2 Execution Modes

DLL Mode

- Use DLL loader and load the DLL

VS

EXE Mode

- Execute pre-compiled EXE



not-powershell

DLL Mode Characteristics

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



EXE Mode Characteristics

- 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 57 PowerShell Automation and other supporting DLLs
 - DLL list on Appendix



not-powershell

Detections

1. Low hanging fruits
2. PowerShDLL loading DLLs ??!!
3. Watch the loaders!
4. EXE Mode loading 57 dlls



1. Low Hanging Fruits

- Information from AssemblyInfo.cs

Assembly info on PowerShdll Source Code

Event 1, Sysmon

General Details

Process Create:

UtcTime: 2019-10-25 22:26:41.319

ProcessGuid: {071dd1ea-76a1-5db3-0000-001082f45f1d}

ProcessId: 3000

Image: C:\Users\Administrator\Desktop\PowerShDLL\PowerShdll.exe

FileVersion: 1.0.0.0

Description: PowerShdll

Product: PowerShdll

Company:

CommandLine: C:\Users\Administrator\Desktop\PowerShDLL\PowerShdll.exe"

CurrentDirectory: C:\Users\Administrator\Desktop\PowerShDLL\

User: CERB-2K12R2-S1\Administrator

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

LogonId: 0x4F8A7

TerminalSessionId: 1

IntegrityLevel: High

Hashes: MD5=A5C16FF0D01610B28BF2F9846E6D306F, SHA256=711616A3DD485D569DBFB139CDC569C473C325A2EA3984320E6590EDF25F3CEA

ParentProcessGuid: {071dd1ea-22c0-5cf0-0000-001099110300}

ParentProcessId: 1824

ParentImage: C:\Windows\explorer.exe

ParentCommandLine: C:\Windows\Explorer.EXE

```
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



2. 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#\b65e41f2dfbf332f77c36113c53762f\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



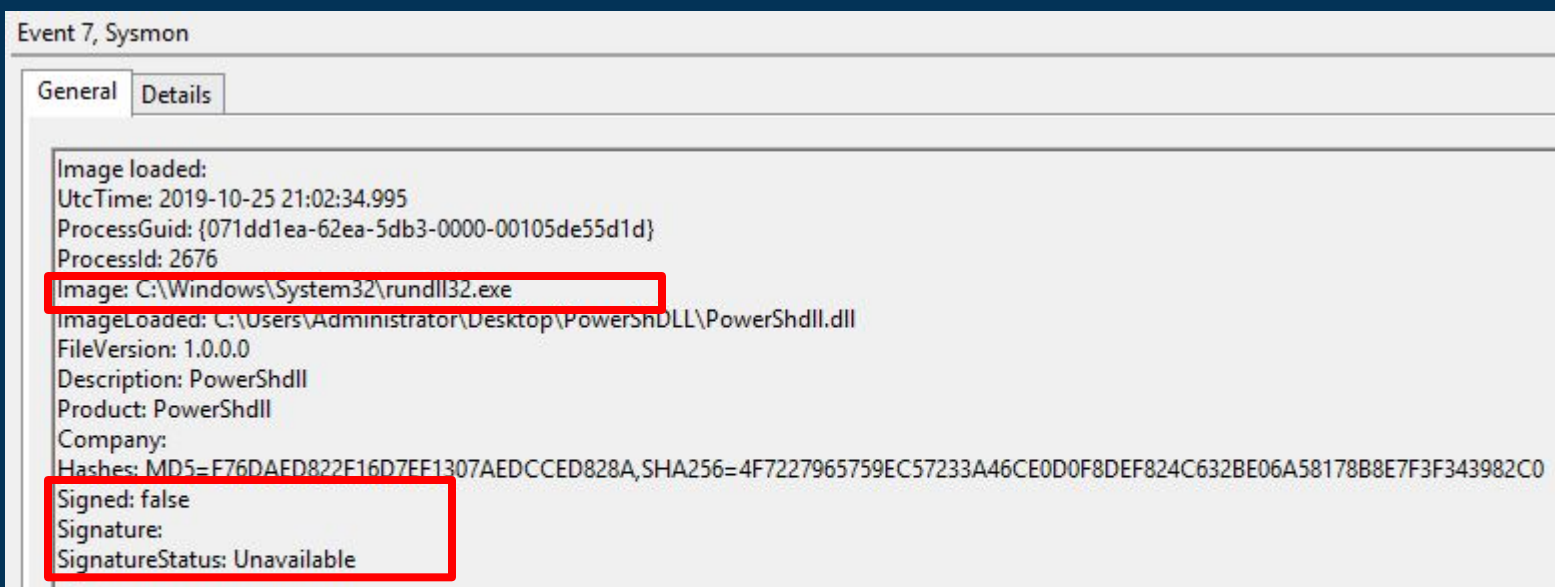
not-powershell

3. 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



Sysmon Event ID 7



not-powershell

4. EXE Mode loading 57 dlls

- This will happen in milliseconds
- Load all the 57 dlls
 - list is on resource slides
- Use Sysmon Event ID 7
- Can use correlation or cardinality rule model
 - When 1 dll is loaded, look if the other 56 dlls are loaded in the last 1 minute



Tools #3 – PowerLessShell

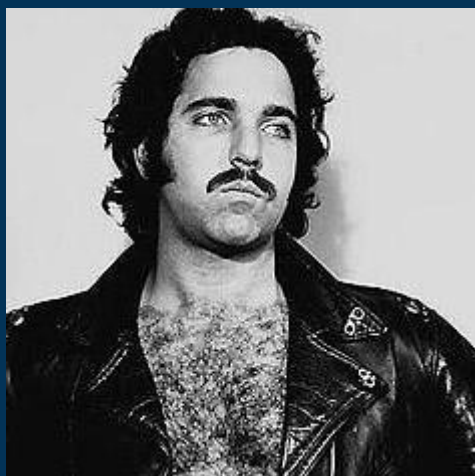
“Don’t worry, we got LOLBAS here”



not-powershell

Tool 3 - PowerLessShell

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



Mr-Un1k0d3r / PowerLessShell

Watch

52

Star

780

Fork

169



not-powershell

PowerLessShell Characteristics

- 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



PowerLessShell Characteristics

- Will get instruction from script file
- Encode command using certutil.exe to perform some kind of obfuscation
- Encode function and variable names

```
39     public class DYwsDEPrRh0ZdRXVcvof Task, ITask {
40         public override bool Execute() {
41             string[] McFB1BcSBs = Environment.GetCommandLineArgs();
42
43             Runspace oIhyCJКСRUMfrrNSYdTRKZ = RunspaceFactory.CreateRunspace();
44             oIhyCJКСRUMfrrNSYdTRKZ.Open();
45             RunspaceInvoke rufPYCoXxxHoERmp = new RunspaceInvoke(oIhyCJКСRUMfrrNSYdTRKZ);
46             Pipeline mWnkhhWfJUjyQsWLnDPeEk = oIhyCJКСRUMfrrNSYdTRKZ.CreatePipeline();
47
48             StreamReader vocrFykeAL = File.OpenText(McFB1BcSBs[1]);
49             StringBuilder CISHuTM1BaUksYsuTdekAAO = 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(CISHuTM1BaUksYsuTdekAAO.ToString());
55                     string CJJlgGyCwTukkkpUFQNhU = Encoding.UTF8.GetString(S1BNF1dqHdSMVSoRRntQCC.WTBVN
```

Sample encoded command



not-powershell

PowerLessShell Components

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



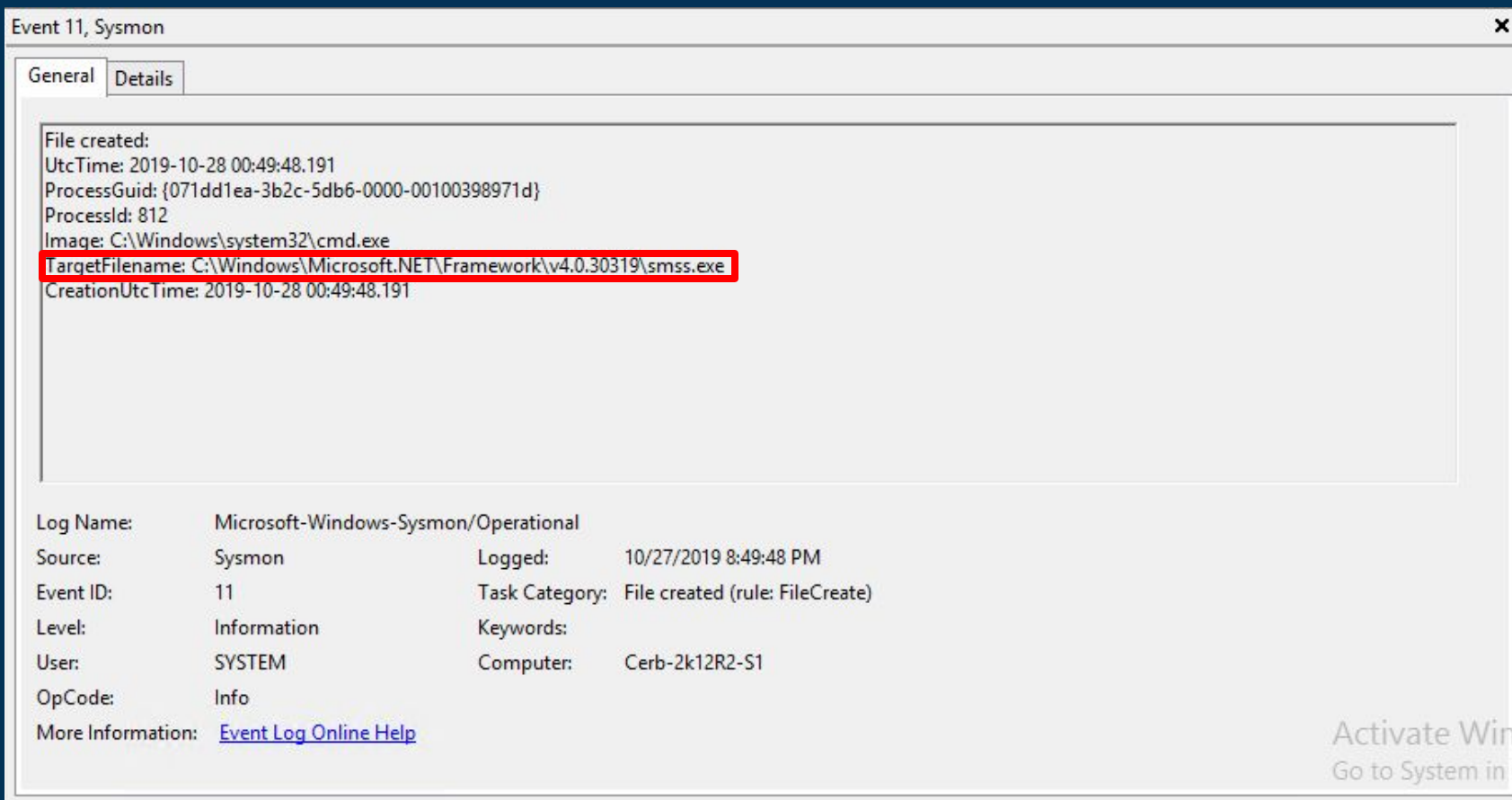
Detections

1. Low hanging fruits
2. Power~~Less~~Shell Logging
3. Suspicious CertUtil.exe and MsBuild.exe
(Correlate both of them)
4. Process Masquerading
5. .NET DLL Loading
6. PowerShell DLL Loading
visible via Process Access



1. Low Hanging Fruits

- Creation of .EXE file on .NET Framework Folder



Sysmon Event ID 11

Activate Windows
Go to System in C



not-powershell

2. PowerLessShell Logging

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

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

General Details

Creating Scriptblock text (1 of 1):

```
echo "Hello World, This is my script launched from PowerLessShell"  
Get-process  
whoami
```

ScriptBlock ID: 823f9ce7-da89-480e-b822-edc30517b3e1
Path:

Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Wind
Event ID: 4104
Level: Verbose
User: CERB-2K12R2-S1\Administr
OpCode: On create calls
More Information: [Event Log Online Help](#)

Logged: 10/27/2019 8:49:48 PM
Task Category: Execute a Remote Command
Keywords: None
Computer: Cerb-2k12R2-S1

PowerShell Event ID 4104



not-powershell

3a. 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
Event ID: 1
Level: Information
User: SYSTEM
OpCode: Info
More Information: [Event Log Online Help](#)

Logged: 10/27/2019 8:49:48 PM
Task Category: Process Create (rule: ProcessCreate)
Keywords:
Computer: Cerb-2k12R2-S1

Activate Windows
Go to System in Settings

Sysmon Event ID 1



not-powershell

3b. 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 acmUMB8idHuxXDQzJNAK

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](#)

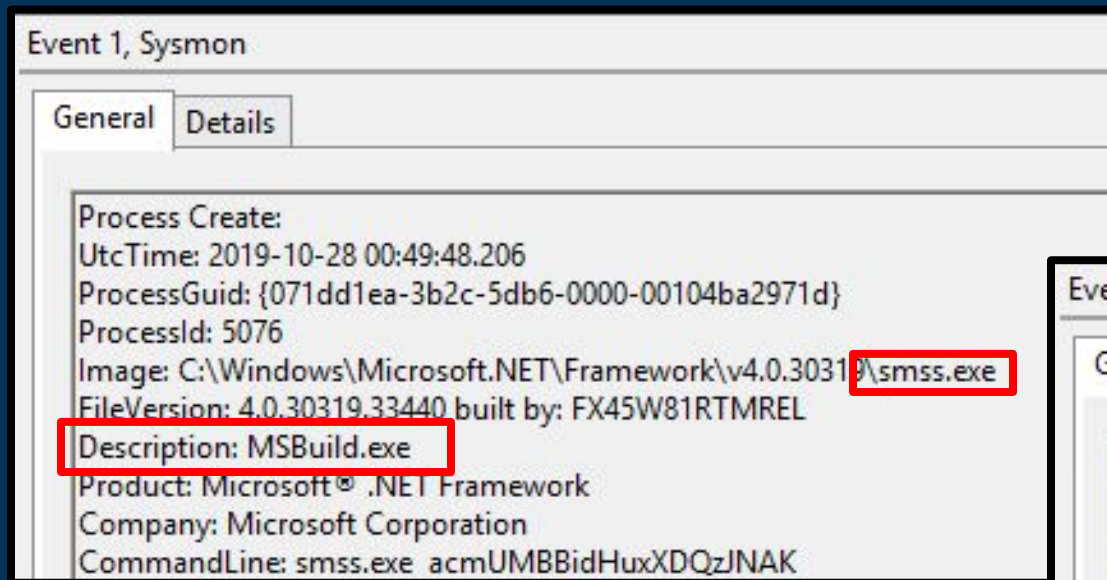
Sysmon Event ID 1



not-powershell

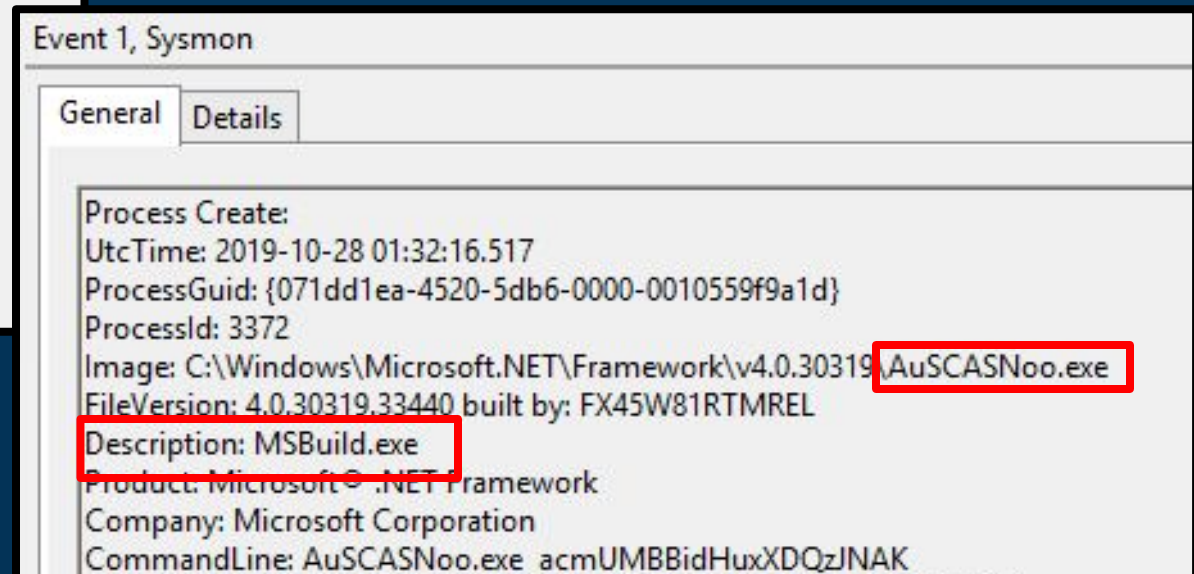
4. Process Masquerading

- When the Process Name != Process Description



Sysmon Event ID 1

Sysmon Event ID 1



5. .NET DLL Loading

- Watch for Microsoft.Build.Tasks.v4.0.dll load events not by common loader (e.g. Visual Studio)

Sysmon Event ID 7

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](#)

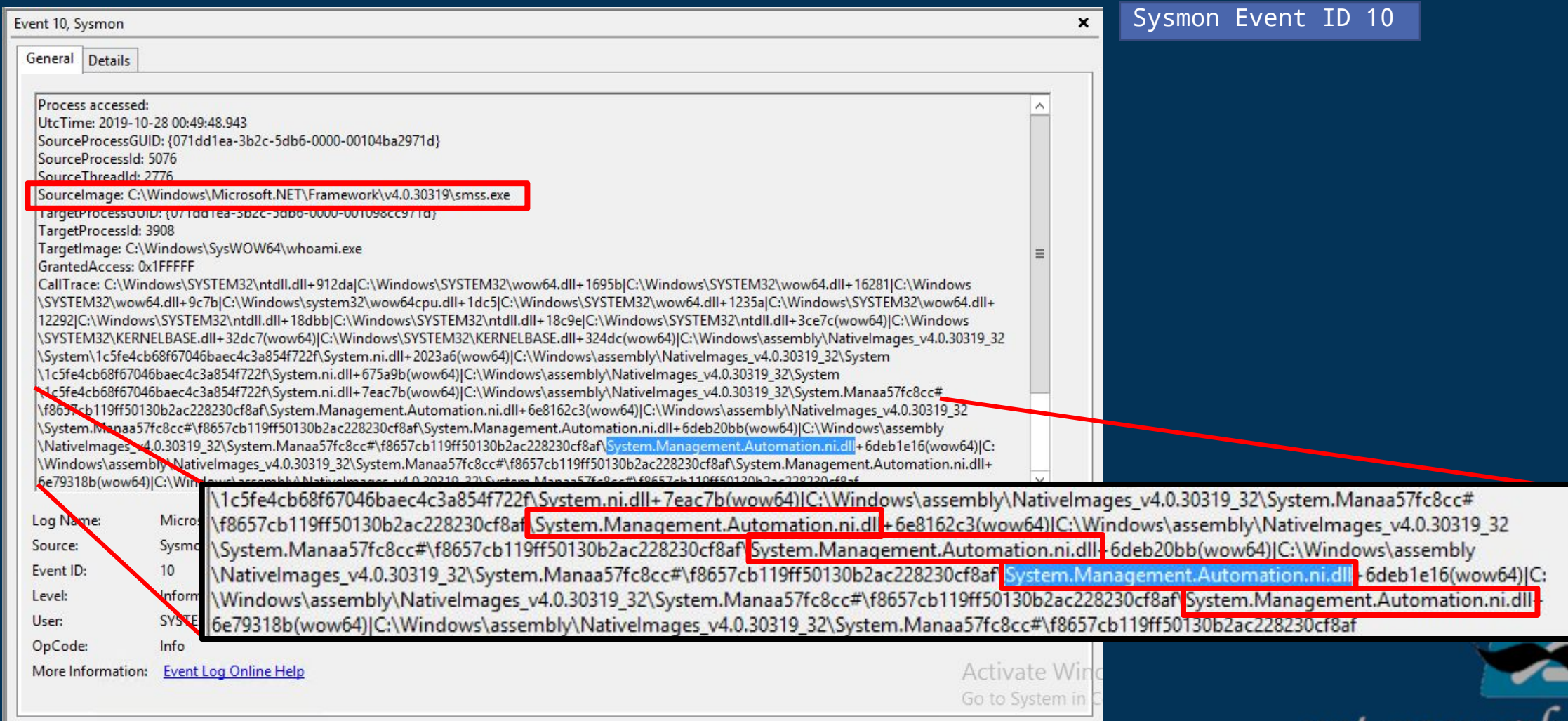
Logged: 10/27/2019 8:49:48 PM
Task Category: Image loaded (rule: ImageLoad)
Keywords:
Computer: Cerb-2k12R2-S1

Activate Windows
Go to System in Control Panel

not-powershell



6. PowerShell DLL Loading visible via Process Access



Tools #4 – NoPowerShell

“Can you C#?”



not-powershell

NoPowerShell

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



not-powershell

NoPowerShell – Features

- Implemented in C# (Really popular these days)
- No System.Management.Automation.dll, only Native .NET library
- PowerShell like cmdlets, extensible!
- 2 modes of execution
 - Run using rundll32.exe
 - Run by loading it to Cobalt Strike!



not-powershell

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



not-powershell

Detections

1. Low hanging fruits
2. Cobalt Strike Mode Detection
3. DLL Modes, watch the loaders!
4. .NET Version Downgrading



1. Low Hanging Fruits

- Information from AssemblyInfo.cs

Event 1, Sysmon

General Details

Process Create:

UtcTime: 2019-10-28 03:25:49.270
ProcessGuid: {071dd1ea-5fbd-5db6-0000-0010107da21d}
ProcessId: 7888
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 Logged: 10/27/2019 11:25:49 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](#)

```
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

2. Cobalt Strike Mode Detection

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

PS: we would love to talk to Cobalt Strike people



not-powershell

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

But!



not-powershell

2. 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
2018-11-29 21:24:35	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x000000000A10B80	
2018-11-29 21:07:20	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x000000000AF0B80	
2018-11-29 19:32:10	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000560B80	
2018-11-29 19:20:45	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000C10B80	
2018-11-29 15:33:59	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000680B80	
2018-11-29 15:18:22	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x00000000000510B80	
2018-11-29 15:15:50	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000250B80	
2018-11-29 14:48:42	Create Remote Thread	alice	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000100B80	
2018-11-29 14:44:40	Create Remote Thread	alice	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000200B80	

target_process_address	parent_guid
0x000000000A10B80	5-3BF4-5C00-0000-0010BEA0B307}
0x000000000AF0B80	5-3BF4-5C00-0000-0010BEA0B307}
0x0000000000560B80	5-3BF4-5C00-0000-0010BEA0B307}
0x0000000000C10B80	5-3BF4-5C00-0000-0010BEA0B307}
0x0000000000680B80	5-024F-5C00-0000-00103D929F07}
0x00000000000510B80	5-024F-5C00-0000-00103D929F07}
0x0000000000250B80	5-D944-5BFF-0000-0010BBA29507}
0x0000000000100B80	5-D944-5BFF-0000-0010BBA29507}
0x0000000000200B80	5-D944-5BFF-0000-0010BBA29507}

Sysmon Event ID 8

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



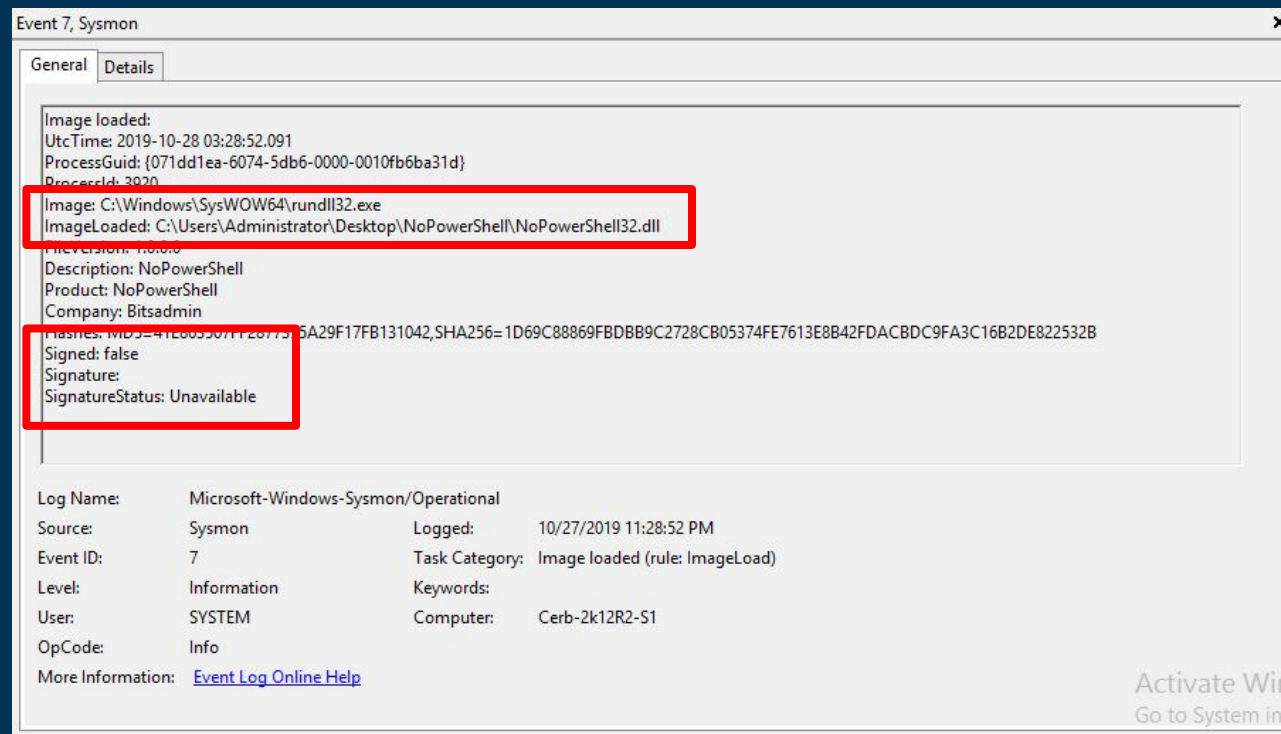
not-powershell

3. DLL Modes, watch the loaders!

Create an alert whenever the rundll32.exe is loading

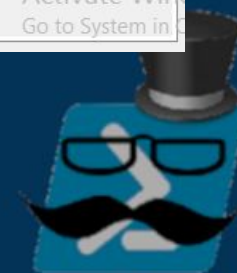
- UNSIGNED DLLs
- UNAVAILABLE SIGNATURE STATUS DLLs

ps: WHITELISTING MIGHT BE REQUIRED



Sysmon Event ID 7

Activate Windows
Go to System in Settings



not-powershell

4. .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:mscorlib.dll`
- Legitimate application might do this...



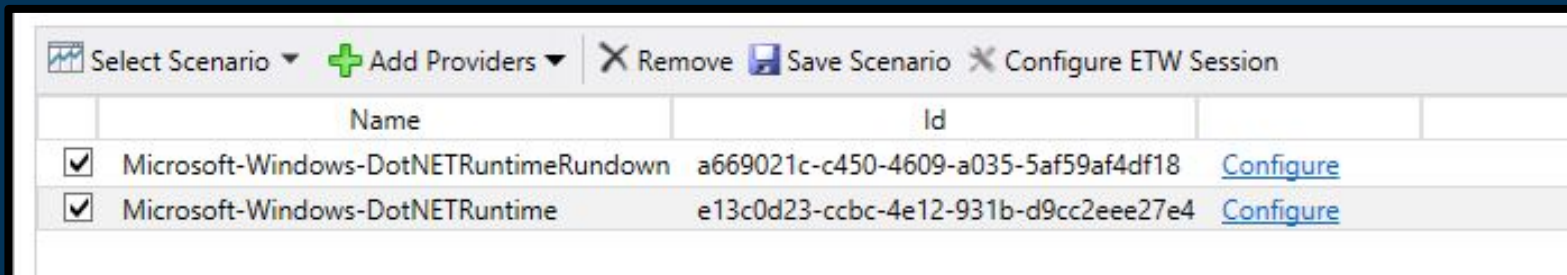
Bonus – Honorary Mention

- 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
 - Similar to PowerLessShell
- 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.
 - Similar to PowerLessShell



Bonus – Detection Ideas

- Sysmon Event ID 10 – Process Access
 - Any application that is accessing PowerShell DLL but not PowerShell!
- Windows PowerShell Event ID 4103 – Pipe Exec
 - Context – Host Application that is not PowerShell!
- 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



not-powershell

PRESENTATION INTRO



PRESENTATION OUTRO



Outro

Conclusion, resources, links, music, etc.



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



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

Because at the end of the day...



not-powershell

Red + Blue + (Others) = MEGAZORDS

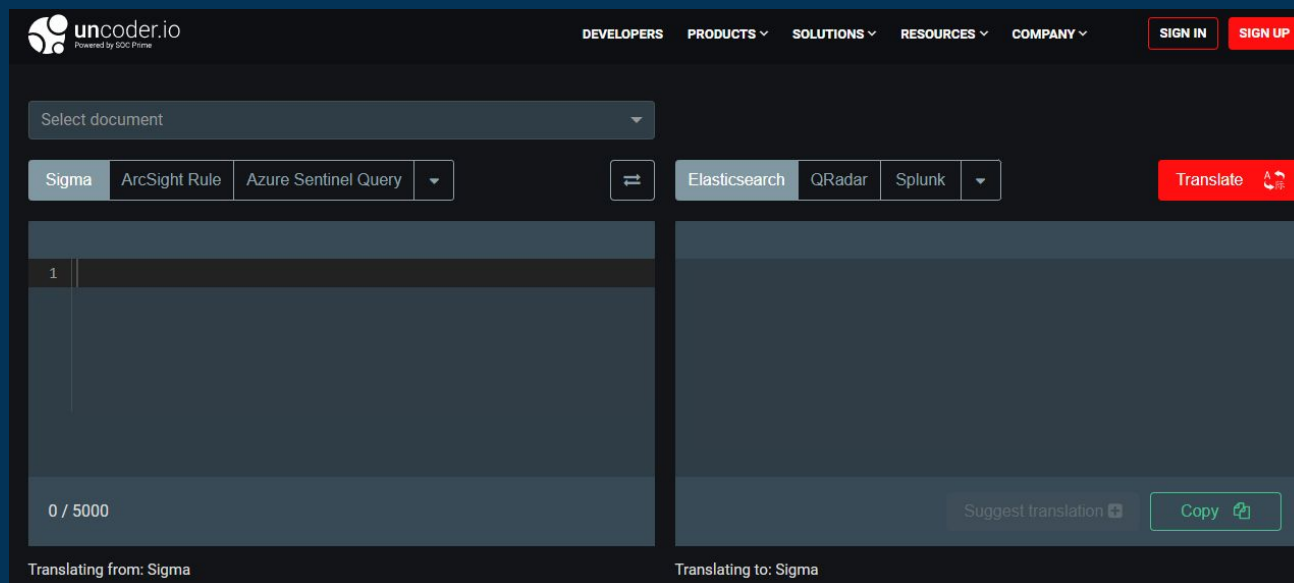


not-powershell

Grab some Sigma Rules on your way out!



Convert using online converter or Sigmac



Uncoder.io

Sigmac



not-powershell

Did you just say MITRE ATT&CK?!

- All rules are mapped to the closest ATT&CK TTPs
- Using the newest version (with sub-technique)
- Mapping the detection to MITRE ATT&CK is bit hard but I tried my best! 😊v
- Feel free to change the mapping

ATT&CK®

This slide is sponsored by mandatory-att&ck-slide gang



not-powershell

Grab some Sigma Rules on your way out!



Special Thanks To..

- My Employer for full support
- @ScoubiMtl and @13Avneet for inspiration, guide, and feedback
- My Co-workers helping me understanding PowerShell and .NET
- The Amazing Not-PowerShell tool creators
- Infosec Community
 - Olaf Hartong's detection
 - Sigma Team
 - MITRE ATT&CK
 - SOC Prime's Uncoder.io
- DEF CON and BTV Organizers, Volunteers and Attendees



not-powershell

Connect with me (and my team)!

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 ?...

Head to Blue Team Village Discord



Flamingo Hotel group
text-talks-track-1 channel

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

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

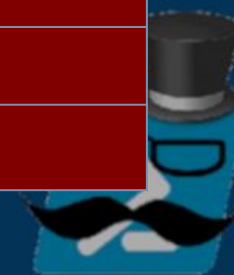
Defense Evasion

Execution, Defense Evasion

Impact, Collection, Persistence, Privilege Escalation

Defense Evasion, Discovery

Lateral Movement, Initial Access



57 DLLs Loaded by PowerShDLL

System.Management.Automation.ni.dll, msvcrt80.dll, System.ni.dll,
System.Data.ni.dll, System.Transactions.dll,
Microsoft.PowerShell.Commands.Utility.ni.dll,
Microsoft.WSMan.Management.ni.dll, mscorjit.dll, System.Data.dll,
System.Transactions.ni.dll, mscorlib.ni.dll, shfolder.dll,
System.Configuration.Install.ni.dll, Culture.dll, System.Core.ni.dll,
mscorlib.dll, Microsoft.PowerShell.Security.ni.dll, System.Xml.ni.dll,
Microsoft.PowerShell.ConsoleHost.ni.dll, Powershell.exe,
Microsoft.PowerShell.Commands.Diagnostics.ni.dll, System.Management.ni.dll,
Microsoft.PowerShell.Commands.Management.ni.dll,
System.DirectoryServices.ni.dll, psapi.dll, profapi.dll, mscoreei.dll,
shell32.dll, mscoree.dll, version.dll, secur32.dll, msctf.dll, imm32.dll,
SHCore.dll, shlwapi.dll, kernel32.dll, ws2_32.dll, advapi32.dll,
rsaenh.dll, cryptsp.dll, kernel.appcore.dll, msasn1.dll, crypt32.dll,
bcrypt.dll, nsi.dll, combase.dll, sspicli.dll, rpcrt4.dll, msvcrt.dll,
KernelBase.dll, bcryptprimitives.dll, ole32.dll, cryptbase.dll, user32.dll,
gdi32.dll, sechost.dll, ntdll.dll



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 Commands Diagnostics
Microsoft.PowerShell.ConsoleHost.ni.dll	Microsoft.PowerShell.ConsoleHost
System.Management.Automation.ni.dll	System.Management.Automation



Presentation

- InvisiShell - <https://youtu.be/Y3oMEiySxcc>



not-powershell

Music Playlists

Aries – WELCOME HOME

- <https://open.spotify.com/album/2ND100ZNBvq6B26feV4gJc>

idk.

- <https://open.spotify.com/playlist/37i9dQZF1DX59NCqCqJtoH>



not-powershell