Have No Fear WMI is Here: Identify Lateral Movement and Malicious Backdoors with WMI

Ryan Becwar
Sales Engineer | Splunk

splunk> .conf19

Forward-Looking Statements

splunk> .conf19

Agenda

The Discussion Points

WMI/MI Overview

What exactly is WMI?

splunk> .conf19

“Windows Management Instrumentation (WMI) is the infrastructure for management data and operations on Windows-based operating systems.”

Microsoft

WMI Functionality

So what exactly can you do with WMI?

What does it do?

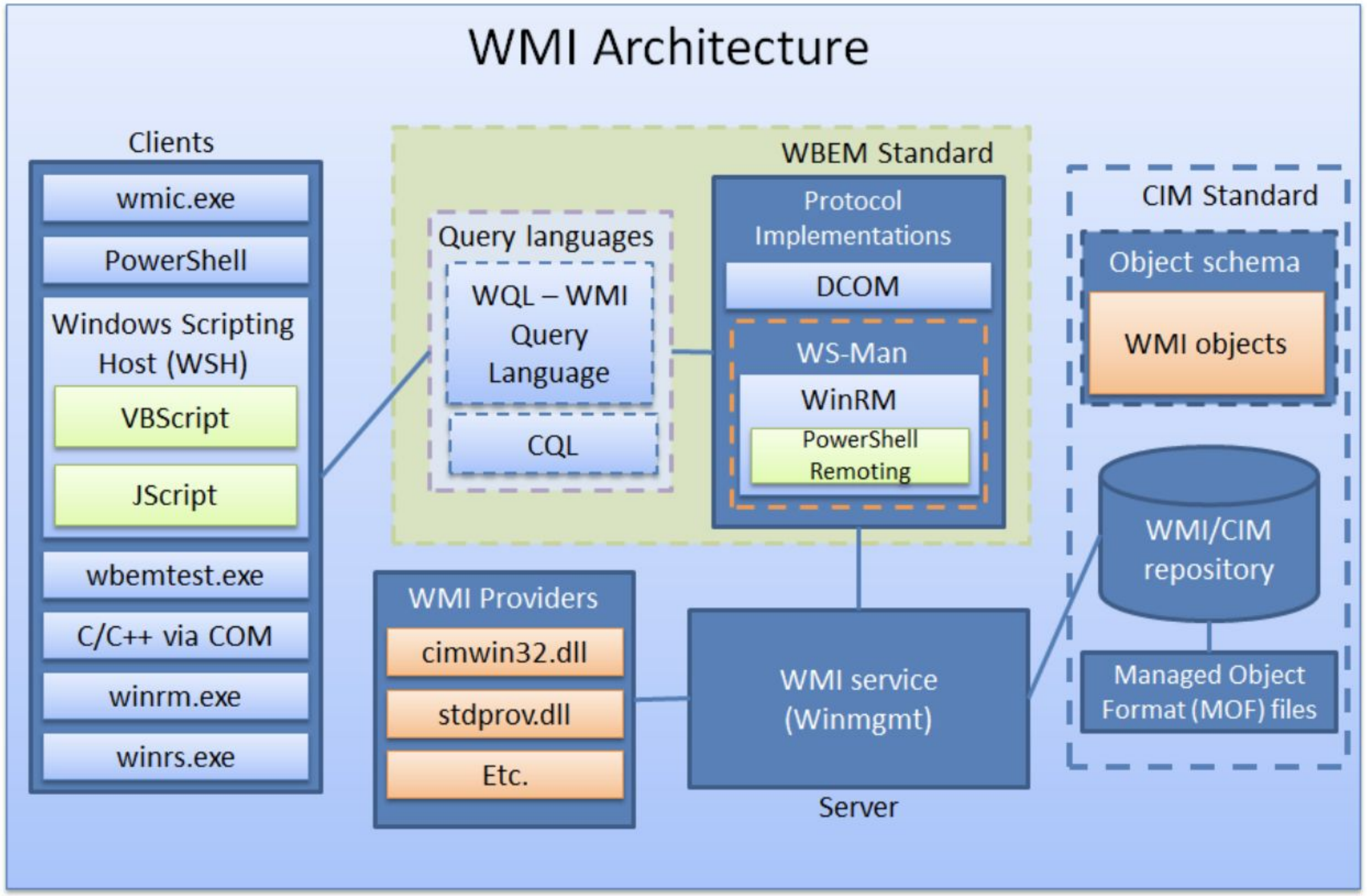
- Automate administrative tasks on remote computers via WMI scripts or applications
- Supplies management data to other parts of the operating system and products

Providers

- Active Directory
- BitLocker Drive Encryption (BDE)
- Boot Configuration Data (BCD)
- Domain Name System (DNS)
- Group Policy provider
- Hyper-V provider
- Mobile Device Management (MDM)
- Shadow Copy
- And many, many more…

WMI Architecture

How does it work?



MI vs. WMI

Infrastructure or Instrumentation — that is the question

Tight alignment with standards

New native-code provider APIs

API support for rich PowerShell semantics

New approach to creating PowerShell cmdlets from MI providers using XML

Backward combability with older WMI implementation

MI is a downloadable update to Windows 7, Windows Server 2008 R2, and Windows Server Standard

WMI/MI Based Attacks

How WMI/MI is leveraged in attacks?

splunk> .conf19

Attackers Love WMI/MI

Can you feel the love?

It’s built in to Windows no install necessary!

- Living off the land (LotL)!

It’s scriptable via PowerShell and Sysmon

- Great for persistence

Fileless

- Information made available through providers

Attackers particularly love Win32_ Process Create

Can be leveraged in remote attacks

- winrm and wmic

Often not monitored

- Lack of knowledge

Leveraged for Internal Recon

- WMI queries to gather system information

Log are often verbose in nature

- Can be noisy

WMI Attack Examples

Who uses WMI anyways?

APT32 used WMI to deploy their tools on remote machines and to gather information about the Outlook process

APT29 used WMI to steal credentials and execute backdoors at a future time

FIN8's spear phishing payloads use WMI to launch malware and spawn cmd.exe execution. They also use WMIC during and post compromise cleanup activities

BlackEnergy uses WMI to gather victim host details

Deep Panda is known to utilize WMI for lateral movement

Empire and Cobalt Strike can use WMI to deliver a payload to a remote host

FireEye Bring Your Own Land (BYOL) attack leverages WMIC to execute payloads while obfuscating PowerShell commands… more on this in a second

WMI/MI Attacks Detections

How can we detect WMI based attacks

splunk> .conf19

Previous .conf Talk on WMI

Conventional way to identify WMI based attacks

Windows WMI Activity Events

- Event IDs 5859 and 5861 (can be extremely noisy)
- Event ID 5860

Sysmon WMI Events

- Event ID 19 Look for new Event Filters being registered
- Event ID 21 Look for Binding

.conf18 splunk>

WMI – The Hacker's Chocolate to Their Powershell Peanut Butter

Understanding And Mitigating Attacks Leveraging Windows Management Instrumentation (WMI)

Rico Valdez | Splunk
rico@splunk.com
October 2018 | Version 1.0

Limitations with Conventional WMI Alerting

Where does conventional alerting fall off the rails?

Not all the activity is logged

- Some techniques are obfuscated

Can be verbose in nature

- Feels like you're drinking from a water hose

Events sometimes contain less actionable information

- WMI Activity logs do NOT contain details of newly created Consumers, Filters or Bindings used for WMI persistence
- Complex correlation is sometimes necessary

ETW are debug logs… more on this in a second

- Format is in binary
- Not ingested by Splunk… or is it? 🤔

Ingesting WMI Events into Splunk

splunk> .conf19

What are Trace Events?

ETW

Event Tracing for Windows (ETW)

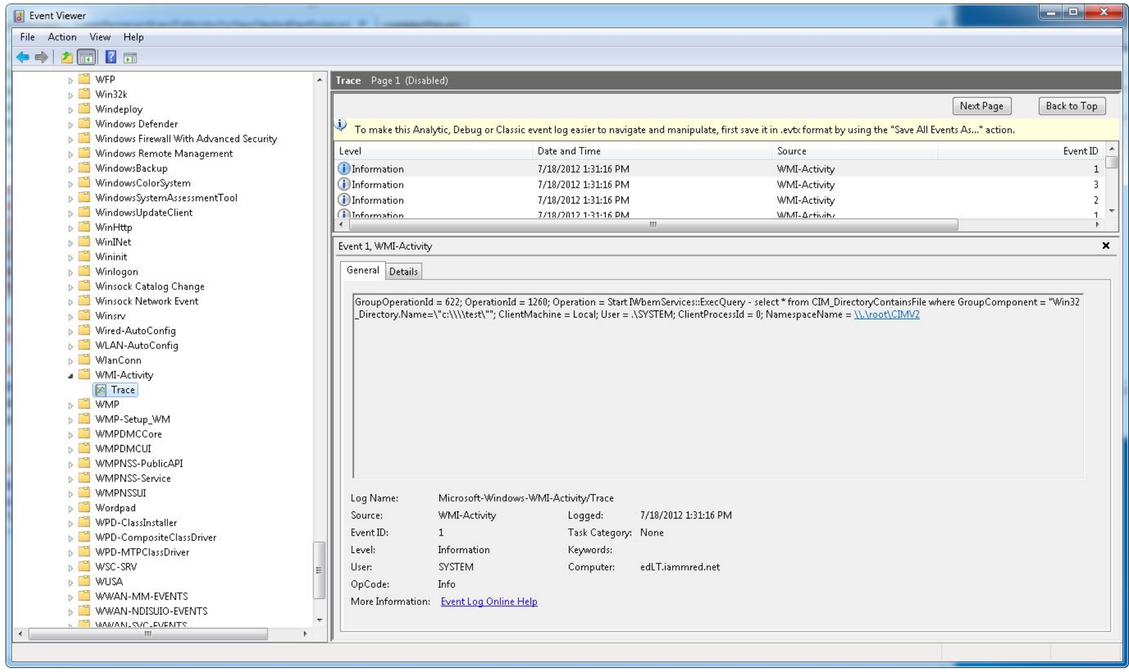
Event logs are a fraction of what is actually produced

- Receives events from ETW
- ETW events that support “Channels” are written to the Event log
- Each Provider has a unique GUID
  - Hyphened

    Microsoft-Windows-WMI-Activity

    Microsoft-Windows-Powershell

    Microsoft-Windows-Sysmon



Trace Page 1 (Disabled)

To make this Analytic, Debug or Classic event log easier to navigate and manipulate, first save it in .evtx format by using the "Save All Events As..." action.

Level	Date and Time	Source	Event ID
Information	7/18/2012 1:31:16 PM	WMI-Activity	1
Information	7/18/2012 1:31:16 PM	WMI-Activity	3
Information	7/18/2012 1:31:16 PM	WMI-Activity	2
Information	7/18/2012 1:31:16 PM	WMI-Activity	1

Event 1, WMI-Activity

```
GroupOperationId = 622; OperationId = 1260; Operation = Start IWbemServices::ExecQuery - select * from CIM_DirectoryContainsFile where GroupComponent = "Win32
_Directory.Name=\"c:\\\\test\""; ClientMachine = Local; User = .\SYSTEM; ClientProcessId = 0; NamespaceName = \\.\root\CIMV2
```

Log Name:	Microsoft-Windows-WMI-Activity/Trace		
Source:	WMI-Activity	Logged:	7/18/2012 1:31:16 PM
Event ID:	1	Task Category:	None
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	edLT.iammred.net
OpCode:	Info		
More Information:	Event Log Online Help		

Benefits of WMI Trace

What's so special about the trace logs

Difficult for Security teams to detect WMI activity through the Event Log on endpoints and the network for

- Remote execution
- Persistence mechanism
- Lateral movement
- etc

Most PowerShell and Sysmon obfuscated WMI based attacks can be seen

Entire log aka… NO FILTERING!

Turn On the Trace Events

Great now what?

Two options for enabling WMI Tracing on endpoints:

- Command line:
  - wevtutil.exe sl Microsoft-Windows-WMI-Activity/Trace /e:true
- Via the Registry:
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Channels\Microsoft-Windows-WMI-Activity/Trac
    e
    DWORD = Enabled (0 or 1)

WMI trace events will be recorded within the

- Event Log file
  - %SystemRoot%\System32\Winevt\Logs\Microsoft-Windows-WMI-Activity%4Trace.etl”. RD = Enabled (0 or 1)

WMI on WMI

WoW… Monitor WMI with WMI?

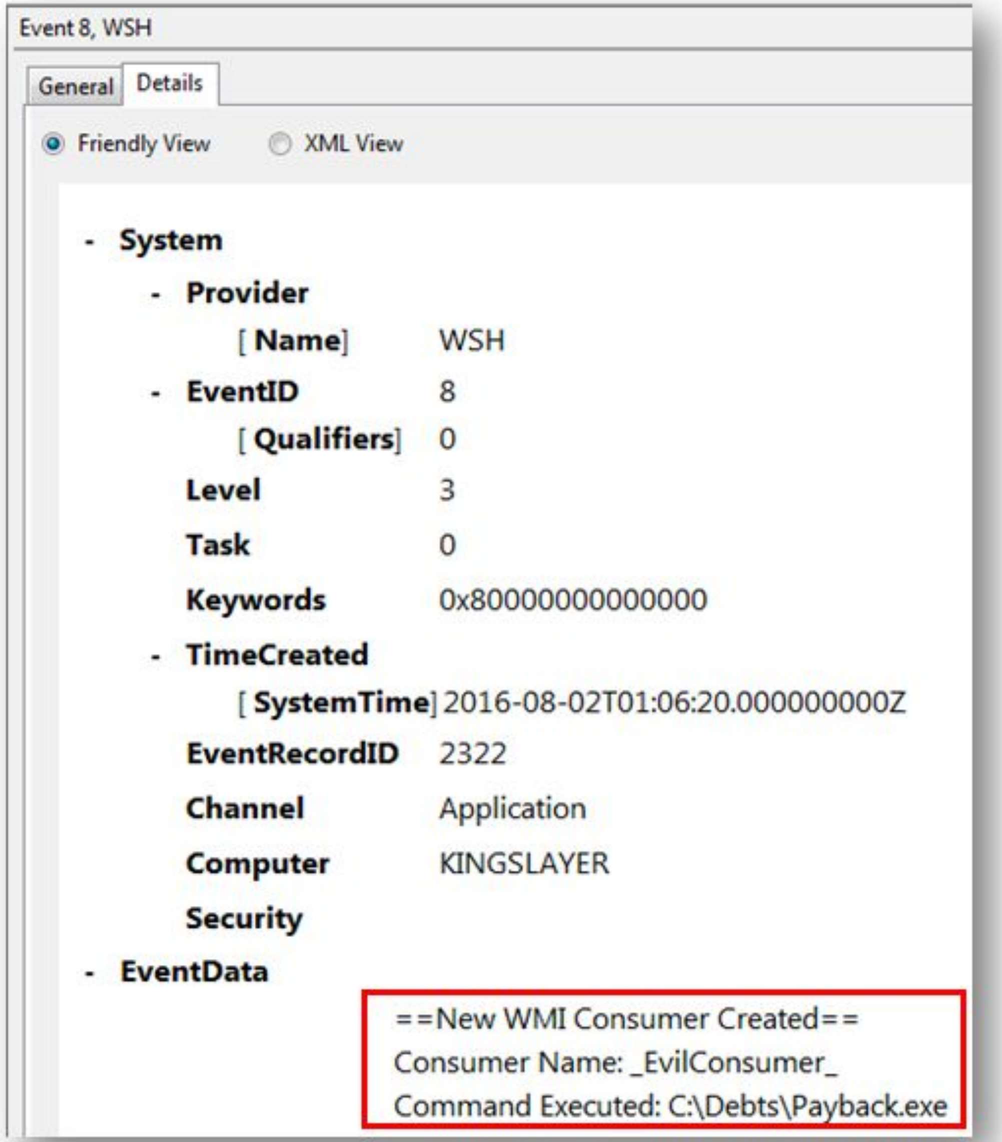
PS Script developed by FireEye

Feed WMI Trace logs from endpoints into SIEM to gain greater visibility

Logs high-fidelity WMI events to the application log

Contains only

- Event Consumer Name
- Event Consumer Command
- Process Call Method
- Process Call Command



```
- System
    - Provider
        [ Name]        WSH
    - EventID          8
        [ Qualifiers]  0
      Level            3
      Task             0
      Keywords         0x80000000000000
    - TimeCreated
        [ SystemTime]  2016-08-02T01:06:20.000000000Z
      EventRecordID    2322
      Channel          Application
      Computer         KINGSLAYER
      Security
- EventData
                       ==New WMI Consumer Created==
                       Consumer Name: _EvilConsumer_
                       Command Executed: C:\Debts\Payback.exe
```

Ingesting the Data into Splunk

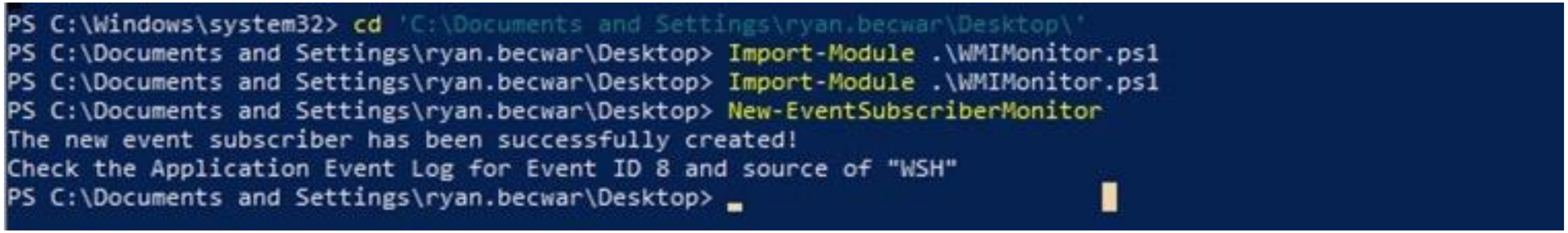
But how do we ingest this into Splunk?

WMI/Trace events are debug logs and the format is in binary

- Splunk currently does not like the format

Run a PowerShell script on the endpoint

- https://gist.github.com/mattifestation/aff0cb8bf66c7f6ef44a#file-example_wmi_detection_eventlogalert-ps1



```
PS C:\Windows\system32> cd 'C:\Documents and Settings\ryan.becwar\Desktop\'
PS C:\Documents and Settings\ryan.becwar\Desktop> Import-Module .\WMIMonitor.ps1
PS C:\Documents and Settings\ryan.becwar\Desktop> Import-Module .\WMIMonitor.ps1
PS C:\Documents and Settings\ryan.becwar\Desktop> New-EventSubscriberMonitor
The new event subscriber has been successfully created!
Check the Application Event Log for Event ID 8 and source of "WSH"
PS C:\Documents and Settings\ryan.becwar\Desktop> _
```

Ingestion Roadblocks

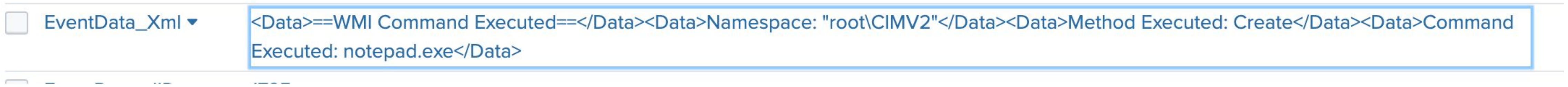
Wait, something's missing…

Some environments may leverage WMI invoked process creations for system administration

- Establish white list/black list

Why am I only seeing just a generalized name for the events and not the entire command that was executed?

- Forked the PS script to include the entire event!
  - https://gist.github.com/rbecwar/2413500e5f8d71e01580c29879d114cf



EventData_Xml

```
<Data>==WMI Command Executed==</Data><Data>Namespace: "root\CIMV2"</Data><Data>Method Executed: Create</Data><Data>Command Executed: notepad.exe</Data>
```

Event Validation

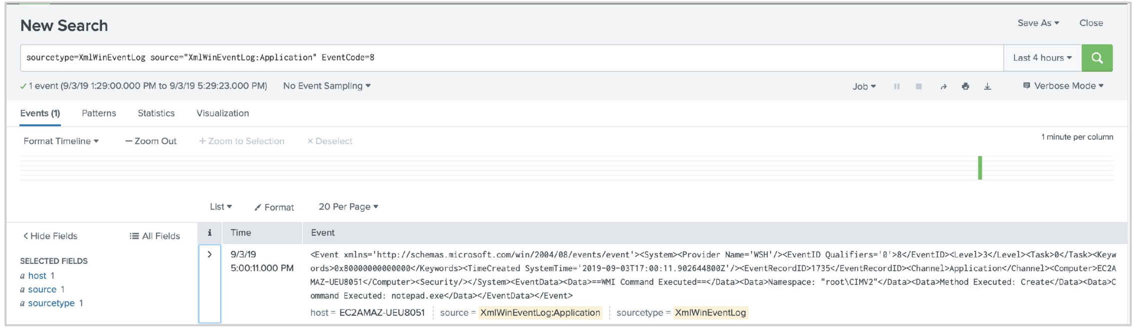
Can we test this?

Detecting malicious WMI trace events

- Malicious events are rolled into one simplistic alert
- sourcetype= WinEventLog:Application Source=WSH EventCode=8
- sourcetype=XmlWinEventLog source="XmlWinEventLog:Application" EventCode=8

PowerShell test script to simulate the attacks

- https://gist.github.com/mattifestation/fa2e3cea76f70b1e2267#file-wmi_attack_detection-ps1-L183


New Search

```
sourcetype=XmlWinEventLog source="XmlWinEventLog:Application" EventCode=8
```

Last 4 hours

✓ 1 event (9/3/19 1:29:00.000 PM to 9/3/19 5:29:23.000 PM)   No Event Sampling   Verbose Mode

Events (1)   Patterns   Statistics   Visualization

Format Timeline   Zoom Out   Zoom to Selection   Deselect   1 minute per column

List   Format   20 Per Page

SELECTED FIELDS
- a host 1
- a source 1
- a sourcetype 1

i	Time	Event
>	9/3/19 5:00:11.000 PM	`<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='WSH'/><EventID Qualifiers='0'>8</EventID><Level>3</Level><Task>0</Task><Keywords>0x80000000000000</Keywords><TimeCreated SystemTime='2019-09-03T17:00:11.902644800Z'/><EventRecordID>1735</EventRecordID><Channel>Application</Channel><Computer>EC2AMAZ-UEU8051</Computer><Security/></System><EventData><Data>==WMI Command Executed==</Data><Data>Namespace: "root\CIMV2"</Data><Data>Method Executed: Create</Data><Data>Command Executed: notepad.exe</Data></EventData></Event>` host = EC2AMAZ-UEU8051 source = XmlWinEventLog:Application sourcetype = XmlWinEventLog

References

Rico Valdez .conf18 “The Hacker’s Chocolate to their PowerShell Peanutbutter”

- https://static.rainfocus.com/splunk/splunkconf18/sess/1523578675195001VubJ/finalPDF/SEC1833_WMITheHackerChocolate_Final_15385980427120012ky0.pdf

FireEye blogs

- https://www.fireeye.com/blog/threat-research/2016/08/wmi_vs_wmi_monitor.html
- https://www.fireeye.com/blog/threat-research/2018/06/bring-your-own-land-novel-red-teaming-technique.html

MITRE ATT&CK

- https://attack.mitre.org/techniques/T1047/

GitHub repo

- https://gist.github.com/rbecwar/2413500e5f8d71e01580c29879d114cf/revisions

Matt Graeber Derbycon 2019

- http://www.irongeek.com/i.php?page=videos/derbycon9/1-05-how-do-i-detect-technique-x-in-windows-applied-methodology-to-definitively-answer-this-question-matt-graeber

Key Takeaways

WMI Trace Events

1. Fairly simple deployment
2. Better visibility for WMI events
3. Great for detecting obfuscated WMI attacks Not a silver bullet!

splunk> .conf19

.conf19

splunk>

Thank You!

Go to the .conf19 mobile app to

RATE THIS SESSION

Q&A

Ryan Becwar | Sales Engineer

splunk> .conf19