

Windows Management Instrumentation

What is WMI?

Windows Management Instrumentation (WMI) is the Microsoft implementation of *Web-Based Enterprise Management* (WBEM), which is an industry initiative to develop a standard technology for accessing management information, within an enterprise environment. [\[1\]](#)

WMI Exploitation

Adversaries may abuse Windows Management Instrumentation (WMI) to execute malicious commands and payloads.

The use of *Windows Management Instrumentation* was ranked as the 3rd most prevalent technique observed in 2021, with **15.4%** of Red Canary's customers having been affected by WMI exploitation. [\[2\]](#)

What MITRE ATTACK [\[3\]](#) framework technique number is applied to the WMI exploitation?

- The technique ID assigned to *WMI Exploitation* is **T1047**.

What type of tactic uses this technique?

- Execution

Execution consists of techniques that result in adversary-controlled code running on a local or remote system. Techniques that run malicious code are often paired with techniques from all other tactics to achieve broader goals, like exploring a network or stealing data. For example, an adversary might use a remote access tool to run a PowerShell script that does Remote System Discovery.

- MITRE ATTACK Framework: Execution [\[4\]](#)

- Discovery

Discovery consists of techniques an adversary may use to gain knowledge about the system and internal network. These techniques help adversaries observe the environment and orient themselves before deciding how to act. They also allow adversaries to explore what they can control and what's around their entry point in order to discover how it could benefit their current objective. Native operating system tools are often used toward this post-compromise information-gathering objective.

- MITRE ATTACK Framework: Discovery [\[5\]](#)

- Lateral Movement

Lateral Movement consists of techniques that adversaries use to enter and control remote systems on a network. Following through on their primary objective often requires exploring the network to find their target and subsequently gaining access to it. Reaching their objective often involves pivoting through multiple systems and accounts to gain. Adversaries might install their own remote access tools to accomplish Lateral Movement or use legitimate credentials with native network and operating system tools, which may be stealthier.

- **MITRE ATTACK Framework: Lateral Movement** [\[6\]](#)

WMI is primarily used as part of the Execution tactic, however certain commands used in this stage can be used to perform Reconnaissance via account enumeration or executing processes remotely as a form of Lateral Movement.

WMI Techniques

Why do malicious actors use WMI?

Malicious actors may use *WMI* due to it being a Windows feature that is often used by administrators for automation, system configuration and to execute scripts. Malicious activity can be hidden within "normal" process activity and the use of the *WMI* can help evade detection.

What can WMI be used for?

An adversary may use *Windows Management Instrumentation* to achieve the following:

- Lateral Movement by remotely executing processes on the target machine. [\[7\]](#)
- Perform Reconnaissance by enumerating processes or groups.
- Modify Systems by deleting *Volume Shadow Copies* during a Ransomware attack in order to hinder recovery. [\[8\]](#)
- Achieve Persistence using the technique *Windows Management Instrumentation Event Subscription (T1546.003)* by executing malicious code that is triggered by a *WMI* Event Subscription. [\[9\]](#)

What are some common components of WMI?

- `wmic.exe` - Command Line utility (Client)
- `Get-WMIObject` - PowerShell cmdlet (Client)
- `wmiprvse.exe` - WMI Provider Host (Server)

Where are the files located or commonly used from?

- `C:\Windows\System32\wbem`
- `C:\Windows\ServicePackFiles\i386`

If the files are not located here, they *could* be suspicious. Investigate further to try determine if the process is legitimate. [\[10\]](#)

Can you name any significant Groups that leverage WMI for malicious activity?

Groups are sets of related intrusion activity that are tracked by a common name in the security community. Analysts track clusters of activities using various analytic methodologies and terms such as threat groups, activity groups, threat actors, intrusion sets, and campaigns. Some groups have multiple names associated with similar activities due to various organizations tracking similar activities by different names. Organizations' group definitions may partially overlap with groups designated by other organizations and may disagree on specific activity.

- **MITRE ATTACK Framework: Groups** [\[11\]](#)

This technique has been leveraged by some large cybercrime organizations, state actors and in significant breaches over the past number of years.

Group	Description	Exploit Used
Cobalt Strike	Cobalt Strike is a commercial, full-featured, remote access tool that bills itself as "adversary simulation software designed to execute targeted attacks and emulate the post-exploitation actions of advanced threat actors".	Cobalt Strike can use WMI to deliver a payload to a remote host.
Emotet	Emotet is a modular malware variant which is primarily used as a downloader for other malware variants such as TrickBot and IcedID.	Emotet has used WMI to execute <code>powershell.exe</code> .
REvil	REvil is a ransomware family that has been linked to the GOLD SOUTHFIELD group and operated as ransomware-as-a-service (RaaS) since at least April 2019.	REvil can use WMI to monitor for and kill specific processes listed in its configuration file.
WannaCry	WannaCry is ransomware that was first seen in a global attack during May 2017, which affected more than 150 countries.	WannaCry utilizes <code>wmic</code> to delete shadow copies.

What can you do to mitigate against WMI exploitation?

Please research mitigations and provide the type and a short description of the mitigation techniques.

ID	Mitigation	Description
M1040	Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to block processes created by WMI commands from running. Note: many legitimate tools and applications utilize WMI for command execution.
M1038	Execution Prevention	Use application control configured to block execution of <code>wmic.exe</code> if it is not required for a given system or network to prevent potential misuse by adversaries. For example, in Windows 10 and Windows Server 2016 and above, Windows Defender Application Control (WDAC) policy rules may be applied to block the <code>wmic.exe</code> application and to prevent abuse.
M1026	Privileged Account Management	Prevent credential overlap across systems of administrator and privileged accounts.
M1018	User Account Management	By default, only administrators are allowed to connect remotely using WMI. Restrict other users who are allowed to connect, or disallow all users to connect remotely to WMI.

How can this type of attack be detected?

To detect the *WMI* being used maliciously we must actively monitor for the following:

- Monitor Network Traffic for *WMI* Connections.
- Monitor for *WMI* use in environments where it is not expected.
- Monitor Process Creation for suspicious processes.
- Monitor Command-Line Arguments for suspicious commands such as account enumeration attempts.

ID	Data Source	Data Component
DS0017	Command	Command Execution
DS0029	Network Traffic	Network Connection Creation
DS0009	Process	Process Creation

Performing regular compromise assessments within an environment is also very beneficial to the organization and can also help with detecting threats, both past and present.

Compromise assessments are high-level investigations where skilled teams utilize advanced tools to dig more deeply into their environment to identify ongoing or past attacker activity in addition to identifying existing weaknesses in controls and practices.

- **CrowdStrike** [\[12\]](#)

These tests are usually performed by vulnerability scanners, and will assess the company's infrastructure. The scans will usually incorporate searching for known *Indicators of Compromise* (IOC) from recently investigated attacks.

An Indicator of Compromise (IOC) is a piece of digital forensics that suggests that an endpoint or network may have been breached. Just as with physical evidence, these digital clues help information security professionals identify malicious activity or security threats, such as data breaches, insider threats or malware attacks.

- **CrowdStrike** [\[13\]](#)

Indicators of Compromise includes:

- Files Hashes
- IP Addresses
- Sign in Activity from unexpected countries.
- Large volumes of sign in requests.

Log Collection

Listed below are log events to track:

- Windows Event ID 4688: Process Creation
- Sysmon Event IDs 19, 20, and 21: WmiEvents
- Windows Event ID 5861: Microsoft-Windows-WMI-Activity/Operational
- Antimalware Scan Interface (AMSI) telemetry

WMI Exploitation Demonstration

In this section, we will demonstrate some of the tactics that can be performed with WMI and then to view the logs to get an idea for what you should look for.

To help with this section, please open the GitHub link for the *Atomic Red Team* atomics page for WMI.

- <https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1047/T1047.md>

T1047

From the Atomic Red Team GitHub for the technique *T1047: Windows Management Instrumentation* shows that there are 10 tests built into the Atomic Red Team toolset.

It may not be possible to run all the tests, however we will run a couple so that you can view any relevant log information

Step 1: Open Client Machine

- Open the Windows 10 machine connected to the Detection Lab configuration.
- Open PowerShell.

Step 2: Confirm that Invoke-AtomicTest is Installed

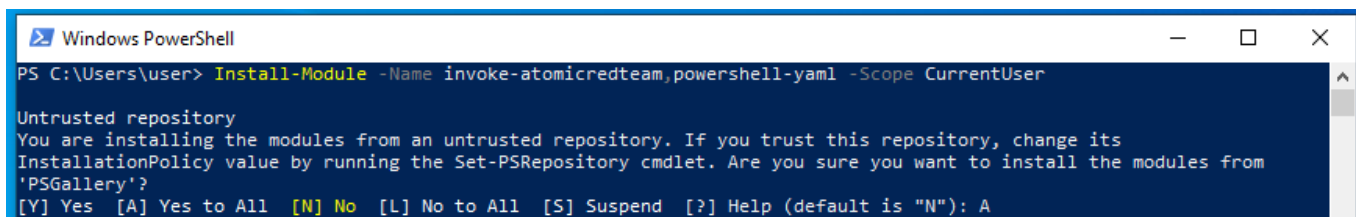
- Confirm that the `Invoke-AtomicTest` cmdlet is installed correctly. This command will install this module.

```
Install-Module -Name invoke-atomicredteam,powershell-yaml -Scope CurrentUser
```

- Type `A` to confirm installing the Module.
- If the module is already installed, you will not be prompted to accept.

Further Reading about the installation process:

- <https://github.com/redcanaryco/invoke-atomicredteam/wiki/Installing-Atomic-Red-Team>



```
Windows PowerShell
PS C:\Users\user> Install-Module -Name invoke-atomicredteam,powershell-yaml -Scope CurrentUser

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
```

Step 3: Check the Prerequisites for T1027

- We need to confirm that all the prerequisites for the tests are available and installed correctly.

```
Invoke-AtomicTest T1047 -CheckPrereqs
```

```
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1047 -CheckPrereqs
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
CheckPrereq's for: T1047-1 WMI Reconnaissance Users
Prerequisites met: T1047-1 WMI Reconnaissance Users
CheckPrereq's for: T1047-2 WMI Reconnaissance Processes
Prerequisites met: T1047-2 WMI Reconnaissance Processes
CheckPrereq's for: T1047-3 WMI Reconnaissance Software
Prerequisites met: T1047-3 WMI Reconnaissance Software
CheckPrereq's for: T1047-4 WMI Reconnaissance List Remote Services
Prerequisites met: T1047-4 WMI Reconnaissance List Remote Services
CheckPrereq's for: T1047-5 WMI Execute Local Process
Prerequisites met: T1047-5 WMI Execute Local Process
CheckPrereq's for: T1047-6 WMI Execute Remote Process
Prerequisites met: T1047-6 WMI Execute Remote Process
CheckPrereq's for: T1047-7 Create a Process using WMI Query and an Encoded Command
Prerequisites met: T1047-7 Create a Process using WMI Query and an Encoded Command
CheckPrereq's for: T1047-8 Create a Process using obfuscated Win32_Process
Prerequisites met: T1047-8 Create a Process using obfuscated Win32_Process
CheckPrereq's for: T1047-9 WMI Execute rundll32
Prerequisites not met: T1047-9 WMI Execute rundll32
[*] DLL with function to execute must exist on disk at specified location ($env:TEMP\calc.dll)
Try installing prereq's with the -GetPrereqs switch
CheckPrereq's for: T1047-10 Application uninstall using WMIC
Prerequisites not met: T1047-10 Application uninstall using WMIC
[*] TightVNC must be installed.
Try installing prereq's with the -GetPrereqs switch
PS C:\Users\vagrant>
```

Step 4: Get the Prerequisites for T1047

- Install the resources required to complete the relevant tests.

```
Invoke-AtomicTest T1047 -GetPrereqs
```

```
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1047 -GetPrereqs
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
GetPrereq's for: T1047-1 WMI Reconnaissance Users
No Preqs Defined
GetPrereq's for: T1047-2 WMI Reconnaissance Processes
No Preqs Defined
GetPrereq's for: T1047-3 WMI Reconnaissance Software
No Preqs Defined
GetPrereq's for: T1047-4 WMI Reconnaissance List Remote Services
No Preqs Defined
GetPrereq's for: T1047-5 WMI Execute Local Process
No Preqs Defined
GetPrereq's for: T1047-6 WMI Execute Remote Process
No Preqs Defined
GetPrereq's for: T1047-7 Create a Process using WMI Query and an Encoded Command
No Preqs Defined
GetPrereq's for: T1047-8 Create a Process using obfuscated Win32_Process
No Preqs Defined
GetPrereq's for: T1047-9 WMI Execute rundll32
Attempting to satisfy prereq: DLL with function to execute must exist on disk at specified location ($env:TEMP\calc.dll)
Prereq successfully met: DLL with function to execute must exist on disk at specified location ($env:TEMP\calc.dll)
GetPrereq's for: T1047-10 Application uninstall using WMIC
Attempting to satisfy prereq: TightVNC must be installed.
Prereq successfully met: TightVNC must be installed.
PS C:\Users\vagrant>
```

Step 5: Begin Testing

I will choose a select few tests to demonstrate the commands used to generate the logs. All the tests can be executed at once, however I prefer to do it test-by-test.

Some tests are designed for Linux or Mac. Ensure that you are attempting to demonstrate the Windows Tests.

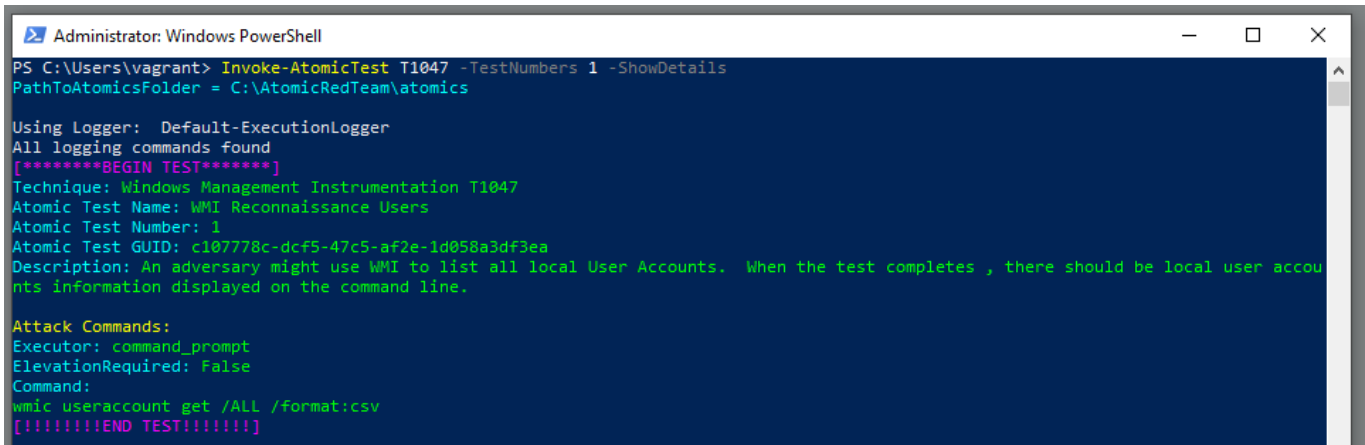
Test #1 - WMI Reconnaissance Users

This test shows how code may be encoded, in the hopes of avoiding detection. The code is then executed. Successful execution of this test should display 'Hey, Atomic!'.

Show Test Details

- Firstly, use the `-ShowDetails` switch to print the details of the specific test to the screen.

```
Invoke-AtomicTest T1047 -TestNumbers 1 -ShowDetails
```



```
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1047 -TestNumbers 1 -ShowDetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

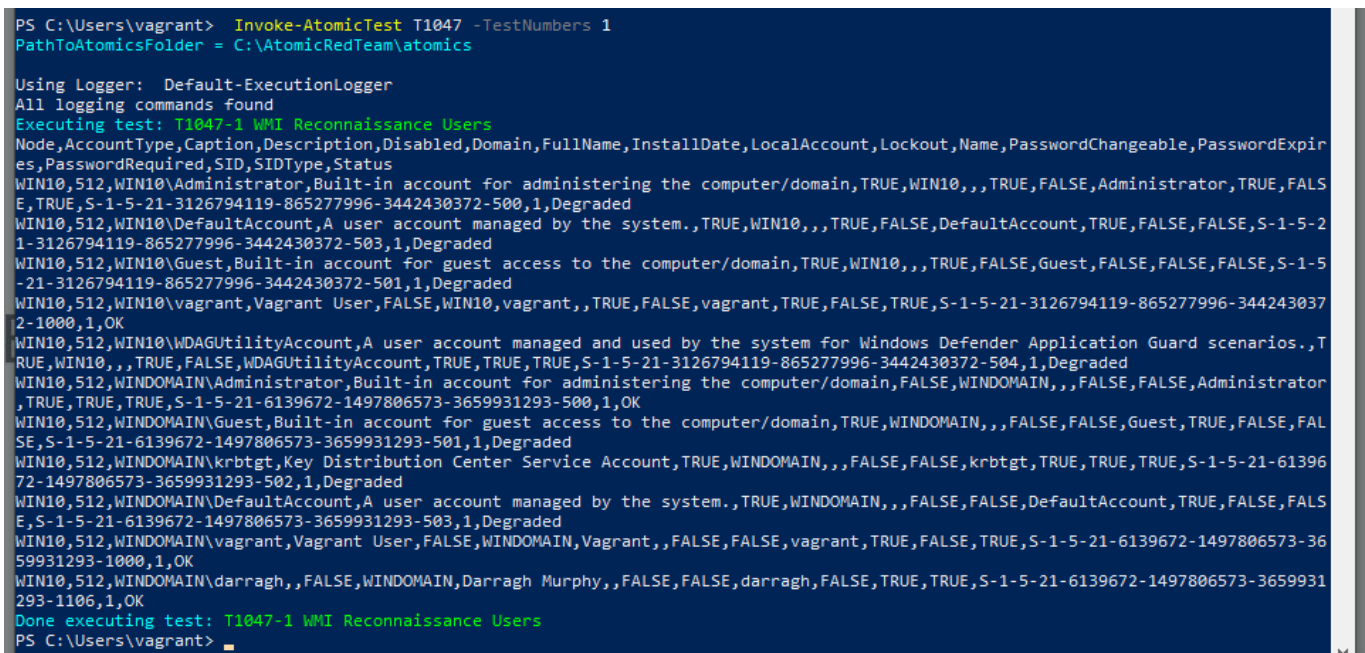
Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Windows Management Instrumentation T1047
Atomic Test Name: WMI Reconnaissance Users
Atomic Test Number: 1
Atomic Test GUID: c107778c-dcf5-47c5-af2e-1d058a3df3ea
Description: An adversary might use WMI to list all local User Accounts. When the test completes , there should be local user accounts information displayed on the command line.

Attack Commands:
Executor: command_prompt
ElevationRequired: False
Command:
wmic useraccount get /ALL /format:csv
[!!!!!!!END TEST!!!!!!!]
```

Execute Test

- Next, we will run the test.

```
Invoke-AtomicTest T1047 -TestNumbers 1
```



```
PS C:\Users\vagrant> Invoke-AtomicTest T1047 -TestNumbers 1
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
Executing test: T1047-1 WMI Reconnaissance Users
Node,AccountType,Caption,Description,Disabled,Domain,FullName,InstallDate,LocalAccount,Lockout,Name>PasswordChangeable>PasswordExpires>PasswordRequired,SID,SIDType,Status
WIN10,512,WIN10\Administrator,Built-in account for administering the computer/domain,TRUE,WIN10,,TRUE,FALSE,Administrator,TRUE,FALSE,TRUE,S-1-5-21-3126794119-865277996-3442430372-500,1,Degraded
WIN10,512,WIN10\DefaultAccount,A user account managed by the system.,TRUE,WIN10,,TRUE,FALSE,DefaultAccount,TRUE,FALSE,FALSE,S-1-5-21-3126794119-865277996-3442430372-503,1,Degraded
WIN10,512,WIN10\Guest,Built-in account for guest access to the computer/domain,TRUE,WIN10,,TRUE,FALSE,Guest,FALSE,FALSE,FALSE,S-1-5-21-3126794119-865277996-3442430372-501,1,Degraded
WIN10,512,WIN10\vagrant,Vagrant User,FALSE,WIN10,vagrant,,TRUE,FALSE,vagrant,TRUE,FALSE,TRUE,S-1-5-21-3126794119-865277996-3442430372-1000,1,OK
WIN10,512,WIN10\WDAGUtilityAccount,A user account managed and used by the system for Windows Defender Application Guard scenarios.,TRUE,WIN10,,TRUE,FALSE,WDAGUtilityAccount,TRUE,TRUE,TRUE,S-1-5-21-3126794119-865277996-3442430372-504,1,Degraded
WIN10,512,WINDOMAIN\Administrator,Built-in account for administering the computer/domain,FALSE,WINDOMAIN,,FALSE,FALSE,Administrator,TRUE,TRUE,TRUE,S-1-5-21-6139672-1497806573-3659931293-500,1,OK
WIN10,512,WINDOMAIN\Guest,Built-in account for guest access to the computer/domain,TRUE,WINDOMAIN,,FALSE,FALSE,Guest,TRUE,FALSE,FALSE,S-1-5-21-6139672-1497806573-3659931293-501,1,Degraded
WIN10,512,WINDOMAIN\krbtgt,Key Distribution Center Service Account,TRUE,WINDOMAIN,,FALSE,FALSE,krbtgt,TRUE,TRUE,TRUE,S-1-5-21-6139672-1497806573-3659931293-502,1,Degraded
WIN10,512,WINDOMAIN\DefaultAccount,A user account managed by the system.,TRUE,WINDOMAIN,,FALSE,FALSE,DefaultAccount,TRUE,FALSE,FALSE,S-1-5-21-6139672-1497806573-3659931293-503,1,Degraded
WIN10,512,WINDOMAIN\vagrant,Vagrant User,FALSE,WINDOMAIN,Vagrant,,FALSE,FALSE,vagrant,TRUE,FALSE,TRUE,S-1-5-21-6139672-1497806573-3659931293-1000,1,OK
WIN10,512,WINDOMAIN\darragh,,FALSE,WINDOMAIN,Darragh Murphy,,FALSE,FALSE,darragh,FALSE,TRUE,TRUE,S-1-5-21-6139672-1497806573-3659931293-1106,1,OK
Done executing test: T1047-1 WMI Reconnaissance Users
PS C:\Users\vagrant>
```


Logs

Next, open up the *Splunk - Search & Reporting* instance and begin searching for the log data surrounding the inputted commands.

- Windows Event Process Creation Event (4688): `index="wineventlog" EventCode=4688 ComputerName="win10.windowmain.local" process_parent_name="cmd.exe"`

i	Time	Event
>	4/23/22 8:14:27.000 PM	<pre>84/23/2022 08:14:27 PM LogName=Security EventCode=4688 EventType=0 ComputerName=win10.windowmain.local SourceName=Microsoft-Windows-Security-Auditing Type=Information RecordNumber=71244 Keywords=Audit Success TaskCategory=Process Creation OpCode=Info Message=A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 Account Name: vagrant Account Domain: WIN10 Logon ID: 0x668CEA Target Subject: Security ID: S-1-0-0 Account Name: - Account Domain: - Logon ID: 0x0 Process Information: New Process ID: 0x500 New Process Name: C:\Windows\System32\wbem\WMIC.exe Token Elevation Type: %L1936 Mandatory Label: S-1-16-12288 Creator Process ID: 0x1378 Creator Process Name: C:\Windows\System32\cmd.exe Process Command Line: wmic useraccount get /ALL /format:csv Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy. Type 1 is a full token with no privileges removed or groups disabled. A full token is only used if User Account Control is disabled or if the user is the built-in Administrator account or a service account. Type 2 is an elevated token with no privileges removed or groups disabled. An elevated token is used when User Account Control is enabled and the user chooses to start the program using Run as administrator. An elevated token is also used when an application is configured to always require administrative privilege or to always require maximum privilege, and the user is a member of the administrators group. Type 3 is a limited token with administrative privileges removed and administrative groups disabled. The limited token is used when User Account Control is enabled, the application does not require administrative privilege, and the user does not choose to start the program using Run as administrator. Account_Domain = WIN10 Account_Domain = Account_Name = vagrant Account_Name = ComputerName = win10.windowmain.local Error_Code = EventCode = 4688 EventType = 0 Keywords = Audit Success LogName = Security Logon_ID = 0x668CEA Logon_ID = 0x0 Message = A new process has been created. Creator Subject Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 Security_ID = S-1-0-0 SourceName = Microsoft-Windows-Security-Auditing. TaskCategory = Process Creation Type = Information action = success app = winunknown body = A new process has been created. Creator Subject Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 category = Process Creation dest_nt_host = win10.windowmain.local dvc = win10.windowmain.local dvc_nt_host = win10.windowmain.local event_description = Process Creation event_id = 71244 event_type = 0 eventtype = windows_process_new_execute process start eventtype = wineventlog_security os windows eventtype = wineventlog_windows os windows eventtype = winsec security host = win10.windowmain.local host_fqdn = win10.windowmain.local host_name = win10.windowmain.local id = 71244 index = wineventlog linecount = 41 member_dn = vagrant - member_sid = S-1-5-21-3126794119-865277996-3442430372-1000 S-1-0-0 name = A new process has been created object = WinEventLog process_id = 0x500 product = Windows punct = / severity = informational session_id = 0x0 severity = informational severity_id = 0 signature = A new process has been created signature_id = 4688 source = WinEventLogSecurity source_type = WinEventLog splunk_server = logger status = success subject = A new process has been created ta_windows_action = failure tag = execute tag = os tag = process tag = security tag = start tag = windows user = user_domain = WIN10 user_logon_id = 0x668CEA user_logon_id = 0x0 user_name = vagrant user_sid = S-1-5-21-3126794119-865277996-3442430372-1000 vendor = Microsoft</pre>

- Sysmon Process Creation Event: `index=sysmon ComputerName="win10.windowmain.local" CommandLine="\"cmd.exe\" /c \"wmic useraccount get /ALL /format:csv\""`

i	Time	Event
>	4/23/22 8:21:43.000 PM	<pre>84/23/2022 08:21:43 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=1 EventType=4 ComputerName=win10.windowmain.local User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=54652 Keywords=None TaskCategory=Process Create (rule: ProcessCreate) OpCode=Info Message=Process Create: RuleName: technique_id=T1059,technique_name=Command-Line Interface UtcTime: 2022-04-23 20:21:43.914 ProcessGuid: {2913fec3-5fd7-6264-bf0d-00000000700} ProcessId: 2132 Image: C:\Windows\System32\cmd.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Windows Command Processor Product: Microsoft® Windows® Operating System Company: Microsoft Corporation OriginalFileName: Cmd.Exe CommandLine: "cmd.exe" /c "wmic useraccount get /ALL /format:csv" CurrentDirectory: C:\Users\vagrant\AppData\Local\Temp\ User: WIN10\vagrant LogonGuid: {2913fec3-c1b5-6262-ea8c-660000000000} LogonId: 0x668CEA TerminalSessionId: 1 IntegrityLevel: High Hashes: SHA1=1A0BD49490F893E52281B06A2024AA508203D5_MD5=9D5944231... Image = C:\Windows\System32\cmd.exe IntegrityLevel = High ParentProcessGuid: {2913fec3-c234-6262-7307-00000000700} ParentImage = C:\Windows\System32\WindowsPowerShell\1.0\powershell.exe ParentProcessId: 860 ParentImage = C:\Windows\System32\WindowsPowerShell\1.0\powershell.exe ParentImage: C:\Windows\System32\WindowsPowerShell\1.0\powershell.exe ParentCommandLine: "C:\Windows\System32\WindowsPowerShell\1.0\powershell.exe" ParentUser: WIN10\vagrant CommandLine = "cmd.exe" /c "wmic useraccount get /ALL /format:csv" Company = Microsoft Corporation ComputerName = win10.windowmain.local CurrentDirectory = C:\Users\vagrant\AppData\Local\Temp\ Description = Windows Command Processor EventCode = 1 EventType = 4 FileVersion = 10.0.18362.1 (WinBuild.160101.0800) Hashes = SHA1=1A0BD49490F893E52281B06A2024AA508203D5_MD5=9D5944231... Image = C:\Windows\System32\cmd.exe IntegrityLevel = High Keywords = None LogName = Microsoft-Windows-Sysmon/Operational LogonGuid = {2913fec3-c1b5-6262-ea8c-660000000000} LogonId = 0x668CEA Message = Process Create: RuleName: technique_id=T1059,technique_name=Command-L... OpCode = Info OriginalFileName = Cmd.Exe ParentCommandLine = "C:\Windows\System32\WindowsPowerShell\1.0\powershell.exe" ParentImage = C:\Windows\System32\WindowsPowerShell\1.0\powershell.exe ParentProcessGuid {2913fec3-c234-6262-7307-00000000700} ParentProcessId = 860 ParentUser = WIN10\vagrant ProcessGuid = {2913fec3-5fd7-6264-bf0d-00000000700} ProcessId = 2132 Product = Microsoft® Windows® Operating System RecordNumber = 54652 RuleName = technique_id=T1059,technique_name=Command-Line Interface Sid = S-1-5-18 SidType = 0 SourceName = Microsoft-Windows-Sysmon TaskCategory = Process Create (rule: ProcessCreate) TerminalSessionId = 1 Type = Information User = NOT_TRANSLATED User = WIN10\vagrant UtcTime = 2022-04-23 20:21:43.914 host = win10.windowmain.local index = sysmon linecount = 38 punct = / severity = informational session_id = 0x0 source = WinEventLogSysmon source_type = XmiWinEventLog:Microsoft-Windows-Sysmon/Operational splunk_server = logger</pre>

What information do you think may be relevant to determine what occurred on the device?

- Use the fields on the left of the Splunk Search to help filter the search results.
- Some of the relevant fields will be as follows:
 - Host
 - Command Line
 - Process Name
 - Parent Process
 - Process Paths
 - IP Addresses
 - File Hash

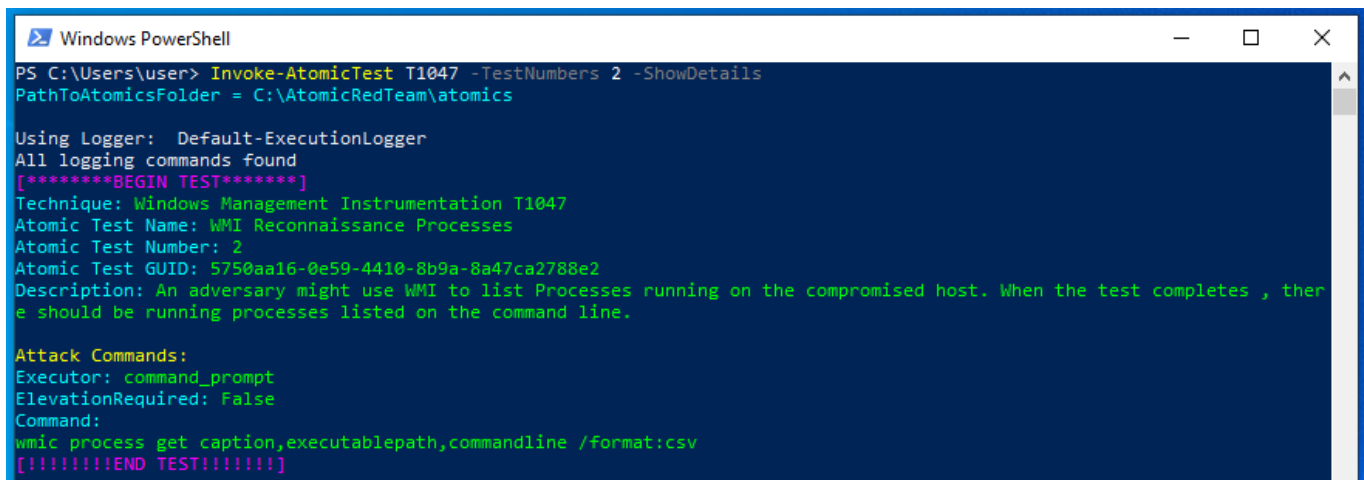
Test #2 - WMI Reconnaissance Processes

This test shows how code may be encoded, in the hopes of avoiding detection. The code is then executed. Successful execution of this test should display 'Hey, Atomic!'.

Show Test Details

- Firstly, use the `-ShowDetails` switch to print the details of the specific test to the screen.

```
Invoke-AtomicTest T1047 -TestNumbers 2 -ShowDetails
```



```
Windows PowerShell
PS C:\Users\user> Invoke-AtomicTest T1047 -TestNumbers 2 -ShowDetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Windows Management Instrumentation T1047
Atomic Test Name: WMI Reconnaissance Processes
Atomic Test Number: 2
Atomic Test GUID: 5750aa16-0e59-4410-8b9a-8a47ca2788e2
Description: An adversary might use WMI to list Processes running on the compromised host. When the test completes , there should be running processes listed on the command line.

Attack Commands:
Executor: command_prompt
ElevationRequired: False
Command:
wmic process get caption,executablepath,commandline /format:csv
[!!!!!!!END TEST!!!!!!!]
```

Execute Test

- Next, we will run the test.

```
Invoke-AtomicTest T1047 -TestNumbers 2
```

```

Windows PowerShell
PS C:\Users\user> Invoke-AtomicTest T1047 -TestNumbers 2
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
Executing test: T1047-2 WMI Reconnaissance Processes
Node,Caption,CommandLine,ExecutablePath
WIN10,System Idle Process,,
WIN10,System,,
WIN10,Registry,,
WIN10,smss.exe,,
WIN10,csrss.exe,,
WIN10,csrss.exe,,
WIN10,wininit.exe,,
WIN10,winlogon.exe,,
WIN10,services.exe,,
WIN10,lsass.exe,,
WIN10,fontdrvhost.exe,,
WIN10,fontdrvhost.exe,,
WIN10,svchost.exe,,
WIN10,svchost.exe,,
WIN10,dwm.exe,,
WIN10,svchost.exe,,
WIN10,svchost.exe,,
WIN10,svchost.exe,,

```

We can see from the testing, and the screenshot above, that testing was completed successfully. - The running processes were displayed in the console, in CSV format.

Logs

Next, open up the *Splunk - Search & Reporting* instance and begin searching for the log data surrounding the inputted commands.

- Windows Event Process Creation Event (4688): `index="wineventlog" ComputerName="win10.windomain.local" signature_id=4688 Creator_Process_Name="C:\\Windows\\System32\\cmd.exe" process_command_line="wmic process get caption,executablepath,commandline /format:csv"`

i	Time	Event
>	4/23/22 8:23:44.000 PM	<pre> 04/23/2022 08:23:44 PM LogName=Security EventCode=4688 EventType=0 ComputerName=win10.windomain.local SourceName=Microsoft Windows security auditing. Type=Information RecordNumber=71762 Keywords=Audit: Success TaskCategory=Process Creation OpCode=Info Message=A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 Account Name: vagrant Account Domain: WIN10 Logon ID: 0x668CEA Target Subject: Security ID: S-1-0-0 Account Name: - Account Domain: - Logon ID: 0x0 Process Information: New Process ID: 0x18 New Process Name: C:\Windows\System32\wsmi\WMI.exe Token Elevation Type: 331936 Mandatory Label: S-1-16-12288 Creator Process ID: 0x1480 Creator Process Name: C:\Windows\System32\cmd.exe Process Command Line: wmic process get caption,executablepath,commandline /format:csv Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy. Type 1 is a full token with no privileges removed or groups disabled. A full token is only used if User Account Control is disabled or if the user is the built-in Administrator account or a service account. Type 2 is an elevated token with no privileges removed or groups disabled. An elevated token is used when User Account Control is enabled and the user chooses to start the program using Run as administrator. An elevated token is also used when an application is configured to always require administrative privilege or to always require maximum privilege, and the user is a member of the Administrators group. Type 3 is a limited token with administrative privileges removed and administrative groups disabled. The limited token is used when User Account Control is enabled, the application does not require administrative privilege, and the user does not choose to start the program using Run as administrator. Account_Domain=WIN10 Account_Domain=- Account_Name=vagrant Account_Name=- ComputerName=win10.windomain.local Error_Code=- EventCode=4688 EventType=0 Keywords=Audit: Success LogName=Security Logon_ID=0x668CEA Logon_ID=0x0 Message=A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 S-1-0-0 SourceName=Microsoft Windows security auditing TaskCategory=Process Creation Type=Information action=success app=wmicunknown body=A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 S-1-0-0 dest=win10.windomain.local dvc=win10.windomain.local dvc_nt_host=win10.windomain.local event_description=Process Creation event_id=71762 event_type=0 eventtype=windows_process_new_execute process_start eventtype=wineventlog_security os windows eventtype=wineventlog_windows os windows eventtype=winsec security host=win10.windomain.local host_fqdn=win10.windomain.local host_name=win10.windomain.local id=71762 index=win@eventlog inaccount=41 member_dn=vagrant+ member_id=S-1-5-21-3126794119-865277996-3442430372-1000 S-1-0-0 name=A new process has been created object=WinEventLog process_id=0x18 product=Windows parent_id=0 source=win@eventlog source_type=WinEventLog splunk_server=logger status=success subject=A new process has been created tag=execute tag=os tag=process tag=security tag=start tag=windows user=- user_domain=WIN10 user_logon_id=0x668CEA user_logon_id=0x0 user_name=vagrant user_sid=S-1-5-21-3126794119-865277996-3442430372-1000 vendor=Microsoft </pre>

- Sysmon Process Creation Event: `index="sysmon" ComputerName="win10.windomain.local" CommandLine="\"cmd.exe\" /c \"wmic process get caption,executablepath,commandline`

```
/format:csv\""
```

i	Time	Event
>	4/23/22 8:23:44.000 PM	<pre> 04/23/2022 08:23:44 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=1 EventType=4 ComputerName=win10.windomain.local User=NOT_TRANSLATED Sid=5-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=54762 Keywords=None TaskCategory=Process Create (rule: ProcessCreate) OpCode=Info Message=Process Create: RuleName: technique_id=T1059,technique_name=Command-Line Interface UtcTime: 2022-04-23 20:23:44.609 ProcessGuid: (2913fec3-6050-6264-c70d-000000000700) ProcessId: 5248 Image: C:\Windows\System32\cmd.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Windows Command Processor Product: Microsoft® Windows® Operating System Company: Microsoft Corporation OriginalFileName: Cmd.Exe CommandLine: "cmd.exe" /c "wmic process get caption,executablepath,commandline //format:csv" CurrentDirectory: C:\Users\vagrant\AppData\Local\Temp\ User: WIN10\vagrant LogonGuid: (2913fec3-clb5-6262-ea8c-660000000000) LogonId: 0x668CEA TerminalSessionId: 1 IntegrityLevel: High Hashes: SHA1=A1DBD4949DF9E892E52201B06A2D24AA5082B3D5_MD5=9D59442313565C2E0860888BF32B2277_SHA256=00CBE18272966AB628EDFF100E9B4A6A3C85DC0F2A32B2B18721FEA2D9C09A5_IMPHASH=272245E2988E1E4305008B52C4F85E18 ParentProcessGuid: (2913fec3-c234-6262-7307-000000000700) ParentProcessId: 860 ParentImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe ParentCommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" ParentUser: WIN10\vagrant CommandLine = "cmd.exe" /c "wmic process get caption,executablepath,commandline //format:csv" Company = Microsoft Corporation ComputerName = win10.windomain.local CurrentDirectory = C:\Users\vagrant\AppData\Local\Temp\ Description = Windows Command Processor EventCode = 1 EventType = 4 FileVersion = 10.0.18362.1 (WinBuild.160101.0800) Hashes = SHA1=A1DBD4949DF9E892E52201B06A2D24AA5082B3D5_MD5=9D5944231... Image = C:\Windows\System32\cmd.exe IntegrityLevel = High Keywords = None LogName = Microsoft-Windows-Sysmon/Operational LogonGuid = (2913fec3-clb5-6262-ea8c-660000000000) LogonId = 0x668CEA Message = Process Create: RuleName: technique_id=T1059,technique_name=Command-L... OpCode = Info OriginalFileName = Cmd.Exe ParentCommandLine = "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" ParentImage = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe ParentProcessGuid = (2913fec3-c234-6262-7307-000000000700) ParentProcessId = 860 ParentUser = WIN10\vagrant ProcessGuid = (2913fec3-6050-6264-c70d-000000000700) ProcessId = 5248 Product = Microsoft® Windows® Operating System RecordNumber = 54762 RuleName = technique_id=T1059,technique_name=Command-Line Interface Sid = 5-1-5-18 SidType = 0 SourceName = Microsoft-Windows-Sysmon TaskCategory = Process Create (rule: ProcessCreate) TerminalSessionId = 1 Type = Information User = NOT_TRANSLATED User = WIN10\vagrant UtcTime = 2022-04-23 20:23:44.609 host = win10.windomain.local Index = Sysmon Inccount = 38 punct = //.../... source = WinEventLog.Sysmon sourcetype = XmlWinEventLog:Microsoft-Windows-Sysmon/Operational splunk_server = logger </pre>

Test #5 - WMI Execute Local Process

This test shows how code may be encoded, in the hopes of avoiding detection. The code is then executed. Successful execution of this test should display 'Hey, Atomic!'.

Show Test Details

- Firstly, use the -ShowDetails switch to print the details of the specific test to the screen.

```
Invoke-AtomicTest T1047 -TestNumbers 5 -ShowDetails
```

```

Windows PowerShell
PS C:\Users\User> Invoke-AtomicTest T1047 -TestNumbers 5 -ShowDetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Windows Management Instrumentation T1047
Atomic Test Name: WMI Execute Local Process
Atomic Test Number: 5
Atomic Test GUID: b3bdfc91-b33e-4c6d-a5c8-d64bee0276b3
Description: This test uses wmic.exe to execute a process on the local host. When the test completes , a new process will
be started locally .A notepad application will be started when input is left on default.

Attack Commands:
Executor: command_prompt
ElevationRequired: False
Command:
wmic process call create #{process_to_execute}
Command (with inputs):
wmic process call create notepad.exe

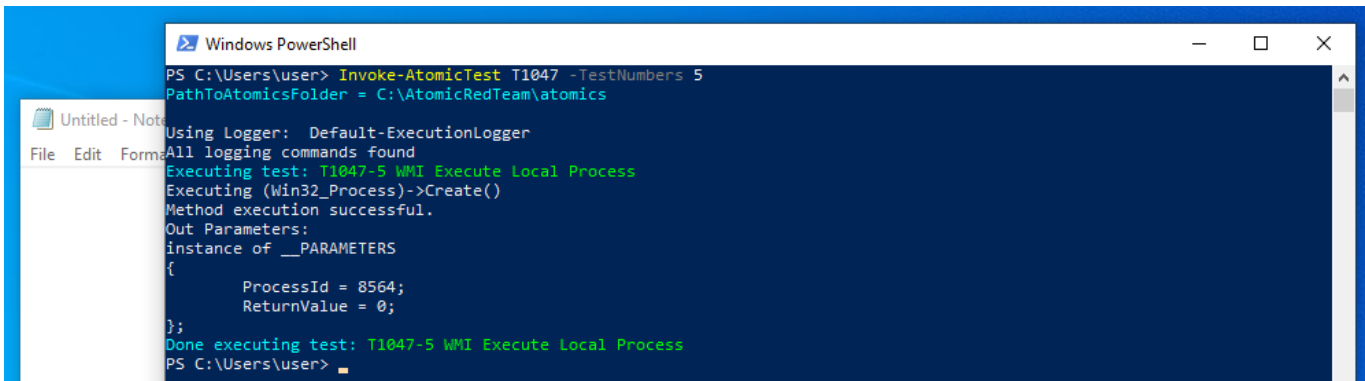
Cleanup Commands:
Command:
wmic process where name='#{process_to_execute}' delete >nul 2>&1
Command (with inputs):
wmic process where name='notepad.exe' delete >nul 2>&1
[!!!!!!!!!!END TEST!!!!!!!!!]

```

Execute Test

- Next, we will run the test.

```
Invoke-AtomicTest T1047 -TestNumbers 5
```



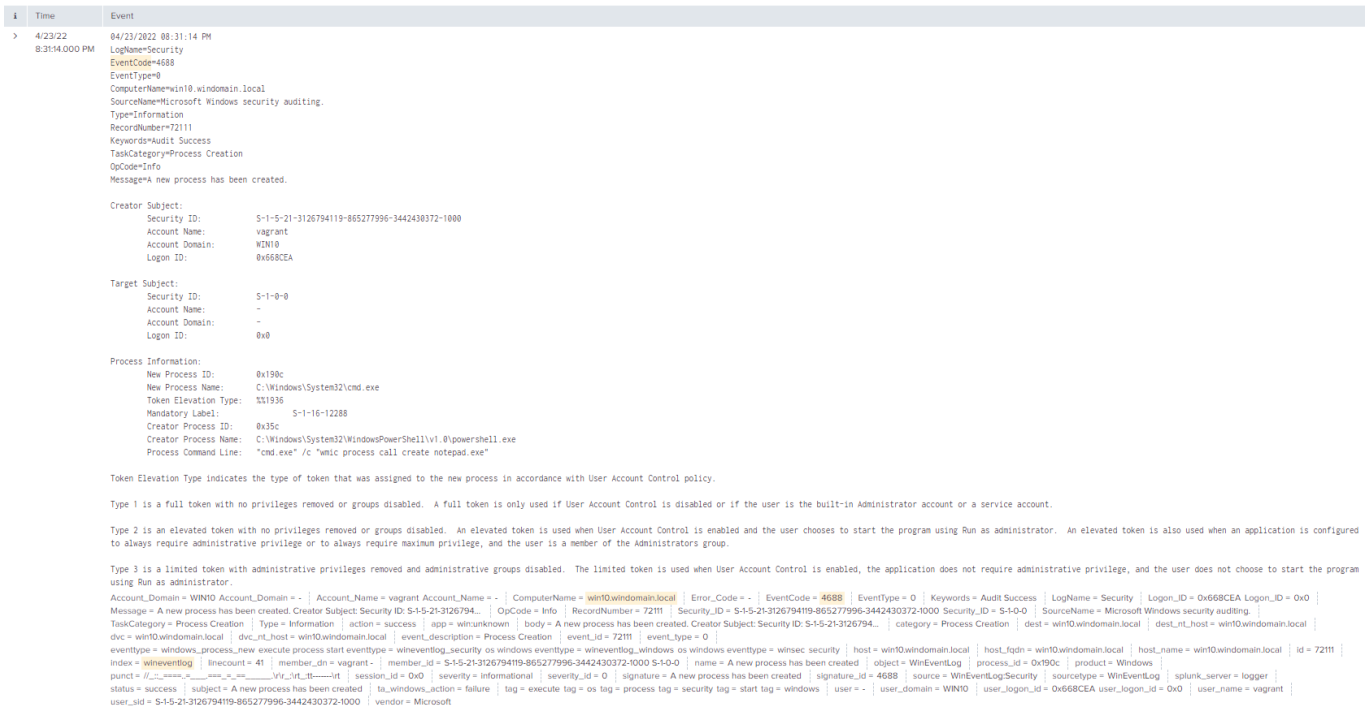
We can see from the testing, and the screenshot above, that testing was completed successfully. - `notepad.exe` opened successfully after this command executed.

Logs

Next, open up the *Splunk - Search & Reporting* instance and begin searching for the log data surrounding the inputted commands.

- Windows Event Process Creation Event (4688): `index="wineventlog"`

```
ComputerName="win10.windomain.local" EventCode=4688 Process_Command_Line="\cmd.exe\" /c \\"wmic process call create notepad.exe\""
```



- Sysmon Process Creation Event: `index="sysmon" ComputerName="win10.windomain.local" CommandLine="\cmd.exe\" /c \\"wmic process call create notepad.exe\""`

i	Time	Event
>	4/23/22 8:31:14.000 PM	<pre> 04/23/2022 08:31:14 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=1 EventType=4 ComputerName=win10.windomain.local User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=55096 Keywords=None TaskCategory=Process Create (rule: ProcessCreate) OpCode=Info Message=Process Create: RuleName= technique_id=T1059,technique_name=Command-Line Interface UtcTime= 2022-04-23 20:31:14.043 ProcessGuid: {2913fec3-6212-6264-d50d-00000000700} ProcessId: 6412 Image: C:\Windows\System32\cmd.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Windows Command Processor Product: Microsoft Windows® Operating System Company: Microsoft Corporation OriginalFileName: Cmd.Exe CommandLine: "cmd.exe" /c "wmic process call create notepad.exe" CurrentDirectory: C:\Users\vagrant\AppData\Local\Temp\ User: WIN10\vagrant LogonGuid: {2913fec3-c1b5-6262-ea8c-660000000000} LogonId: 0x668CEA TerminalSessionId: 1 IntegrityLevel: High Hashes: SHA1=A1D6D4949DF9E892E52201B06A2D24AA5082B3D5_MD5=9D5944231... ParentProcessGuid: {2913fec3-c234-6262-7307-000000000700} ParentProcessId: 860 ParentImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe ParentCommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" ParentUser: WIN10\vagrant CommandLine = "cmd.exe" /c "wmic process call create notepad.exe" Company = Microsoft Corporation ComputerName = win10.windomain.local CurrentDirectory = C:\Users\vagrant\AppData\Local\Temp\ Description = Windows Command Processor EventCode = 1 EventType = 4 FileVersion = 10.0.18362.1 (WinBuild.160101.0800) Hashes = SHA1=A1D6D4949DF9E892E52201B06A2D24AA5082B3D5_MD5=9D5944231... Image = C:\Windows\System32\cmd.exe IntegrityLevel = High Keywords = None LogName = Microsoft-Windows-Sysmon/Operational LogonGuid = {2913fec3-c1b5-6262-ea8c-660000000000} LogonId = 0x668CEA Message = Process Create: RuleName: technique_id=T1059,technique_name=Command-Li... OpCode = Info OriginalFileName = Cmd.Exe ParentCommandLine = "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" ParentImage = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe ParentProcessGuid = {2913fec3-c234-6262-7307-000000000700} ParentProcessId = 860 ParentUser = WIN10\vagrant ProcessGuid = {2913fec3-6212-6264-d50d-000000000700} ProcessId = 6412 Product = Microsoft® Windows® Operating System RecordNumber = 55096 RuleName = technique_id=T1059,technique_name=Command-Line Interface Sid = S-1-5-18 SidType = 0 SourceName = Microsoft-Windows-Sysmon TaskCategory = Process Create (rule: ProcessCreate) TerminalSessionId = 1 Type = Information User = NOT_TRANSLATED User = WIN10\vagrant UtcTime = 2022-04-23 20:31:14.043 host = win10.windomain.local index = sysmon linecount = 38 punct = //... source = WinEventLog\Sysmon sourcetype = XmlWinEventLog\Microsoft-Windows-Sysmon/Operational splunk_server = logger </pre>

Step 6: Clean Up

- Some tests may change items within your environment.
- Run command the following command to clean up any changes made to the system while performing tests.

```
Invoke-AtomicTest T1047 -Cleanup
```

References

1. <https://docs.microsoft.com/en-us/windows/win32/wmisdk/about-wmi>↔
2. https://resource.redcanary.com/rs/003-YRU-314/images/2022_ThreatDetectionReport_RedCanary.pdf↔
3. <https://attack.mitre.org/>↔
4. <https://attack.mitre.org/tactics/TA0002/>↔
5. <https://attack.mitre.org/tactics/TA0007/>↔
6. <https://attack.mitre.org/tactics/TA0008/>↔
7. <https://www.microsoft.com/security/blog/2021/01/20/deep-dive-into-the-solorigate-second-stage-activation-from-sunburst-to-teardrop-and-raindrop/>↔
8. <https://redcanary.com/threat-detection-report/techniques/windows-management-instrumentation/>↔
9. <https://attack.mitre.org/techniques/T1546/003/>↔
10. <https://www.processlibrary.com/en/directory/files/wmic/29005/>↔
11. <https://attack.mitre.org/groups/>↔
12. <https://www.crowdstrike.com/cybersecurity-101/compromise-assessments/>↔
13. <https://www.crowdstrike.com/cybersecurity-101/indicators-of-compromise/>↔

