Detecting & Removing an Attacker's WMI Persistence

By David French

Published: 2018-10-09 · Archived: 2026-04-05 17:27:51 UTC

Press enter or click to view image in full size

Windows Management Instrumentation (WMI) Event Subscription is a popular technique to establish persistence on an endpoint. I decided to spend some time playing with Empire's WMI modules and analyzing the artifacts for detection opportunities. I also reviewed the PowerShell commands that can be used to view and remove WMI event subscriptions.

"Windows Management Instrumentation Event Subscription" is MITRE ATT&CK Technique T1084.

Attackers may use the capabilities of WMI to subscribe to an event and execute arbitrary code when that event occurs, providing persistence on a system.

What is WMI?

"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 in an enterprise environment.

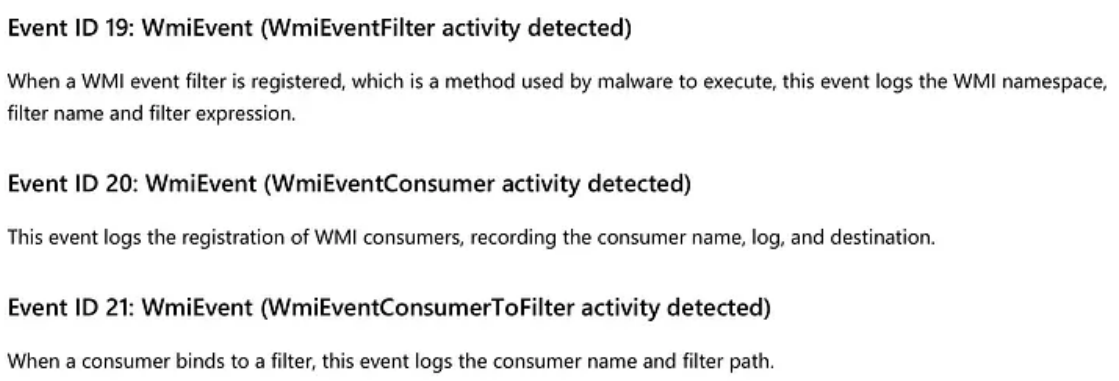
WMI uses the Common Information Model (CIM) industry standard to represent systems, applications, networks, devices, and other managed components."

An **event filter** is a WMI class that describes which events WMI delivers to an **event consumer**. An event filter also describes the conditions under which WMI delivers the events.

Configuring Sysmon Logging

Sysmon can be configured to log `WmiEventFilter`, `WmiEventConsumer`, and `WmiEventConsumerToFilter` activity and enable the detection of WMI abuse.

Press enter or click to view image in full size



Sysmon Event IDs for WMI activity

Roberto Rodriguez's (@Cyb3rWard0g) Sysmon configuration file will capture the above Event IDs.

Get David French's stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

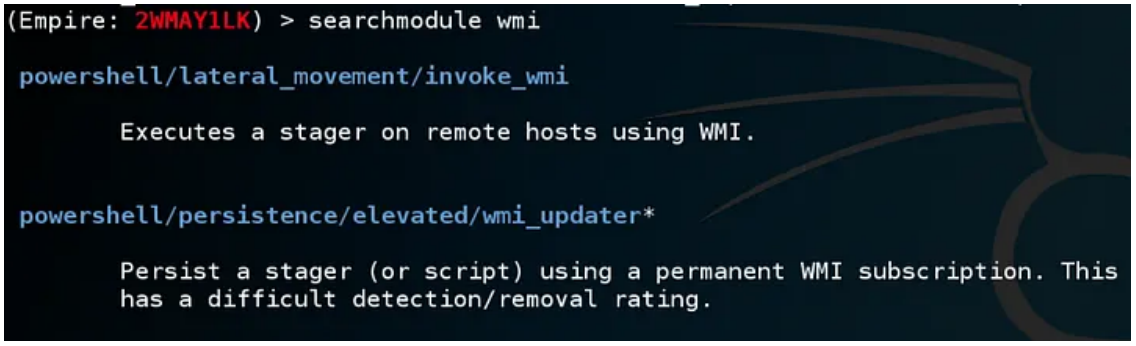
Execute the following command to install Sysmon and apply a configuration file.

```
sysmon.exe -i -c .\config_file.xml
```

Establish Persistence

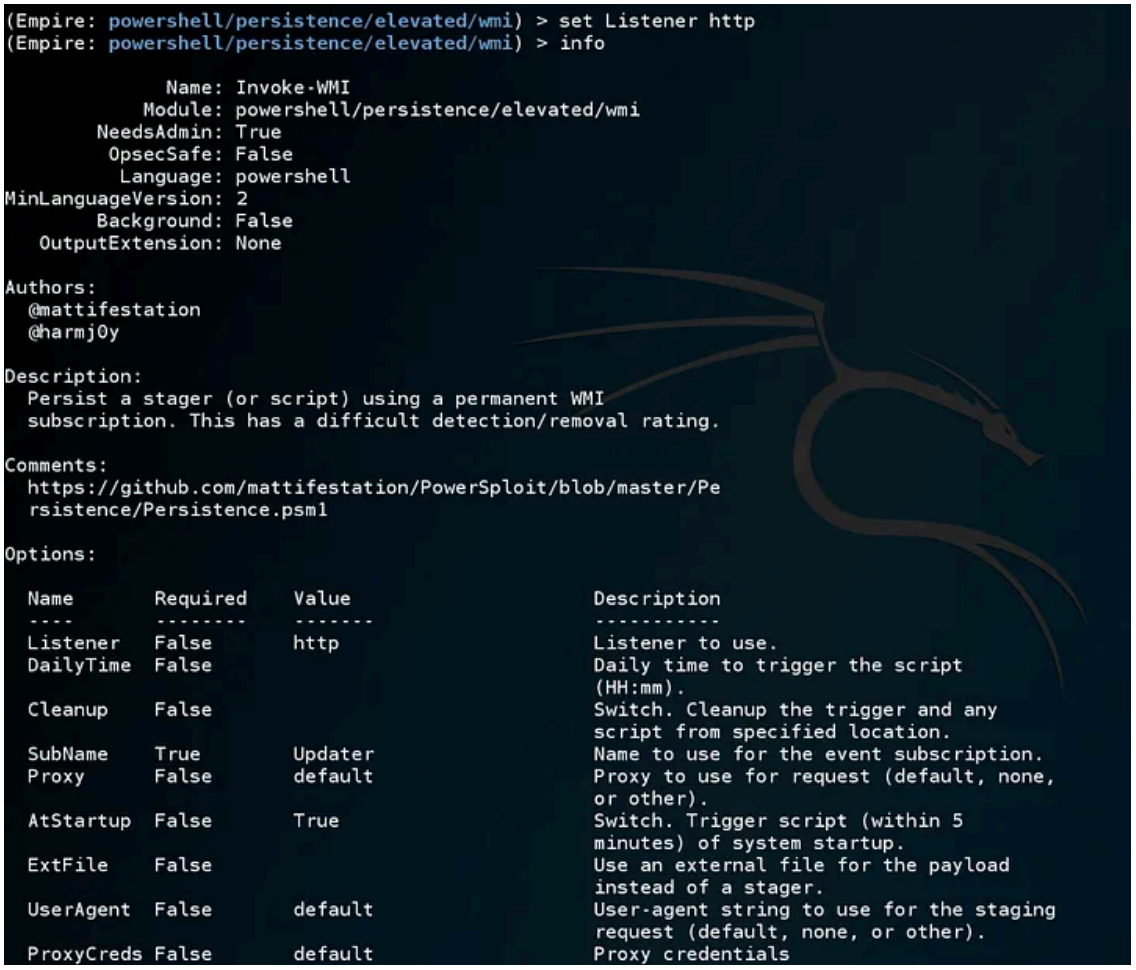
Let's use Empire's `Invoke-WMI` module to create a permanent WMI subscription and persist a stager on the victim endpoint.

Press enter or click to view image in full size



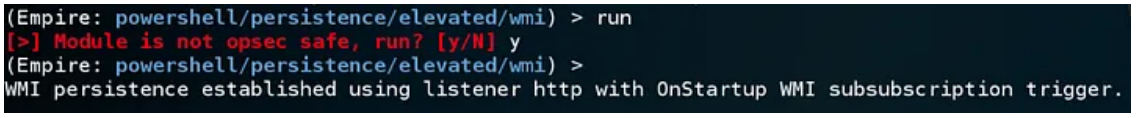
Reviewing Empire's WMI-related modules

Press enter or click to view image in full size



Reviewing the options for Empire's Invoke-WMI module

Press enter or click to view image in full size



Running the module

Detection

Reviewing the Sysmon logs we can see that the Empire module:

1. Registered a WMI event filter
2. Registered a WMI event consumer
3. Bound the event consumer to the event filter

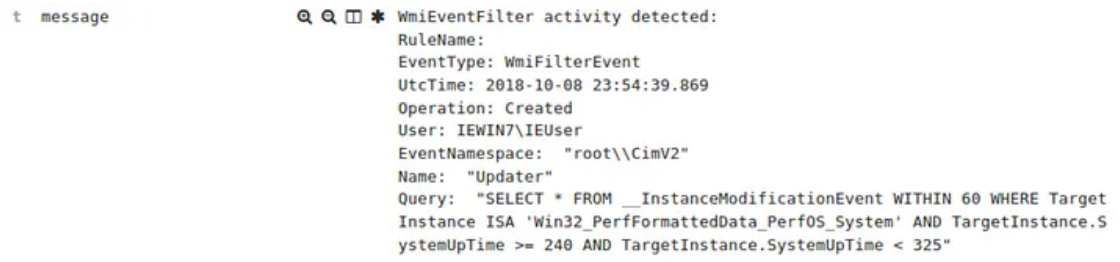
Press enter or click to view image in full size

Time	event_id	task
October 8th 2018, 18:54:39.869	19	WmiEventFilter activity detected (rule: WmiEvent)
October 8th 2018, 18:54:39.884	20	WmiEventConsumer activity detected (rule: WmiEvent)
October 8th 2018, 18:54:56.212	21	WmiEventConsumerToFilter activity detected (rule: WmiEvent)

Sysmon events logged after Empire Invoke-WMI module execution

The WMI event filter sets the conditions for the stager to execute, which includes references to the system's uptime.

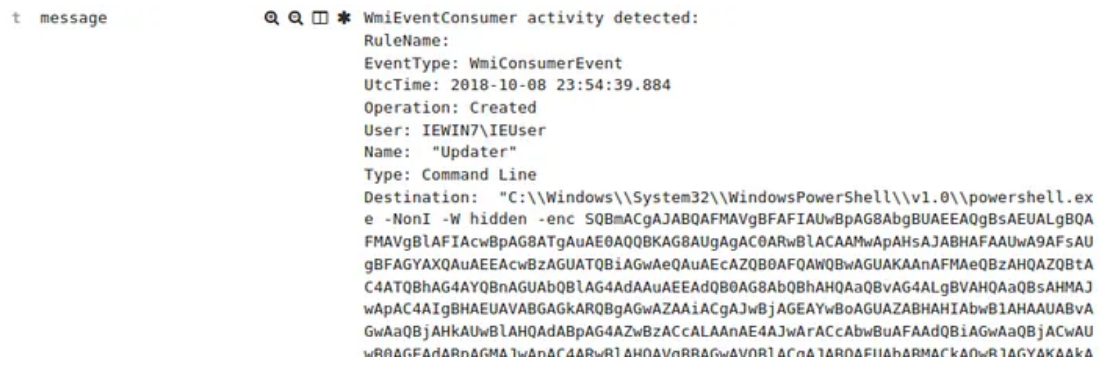
Press enter or click to view image in full size



Sysmon Event ID 19: WmiEvent (WmiEventFilter activity detected)

The WMI event consumer contains the Empire stager in Base64-encoded form and is registered with the innocuous name, `Updater` when its default settings are used.

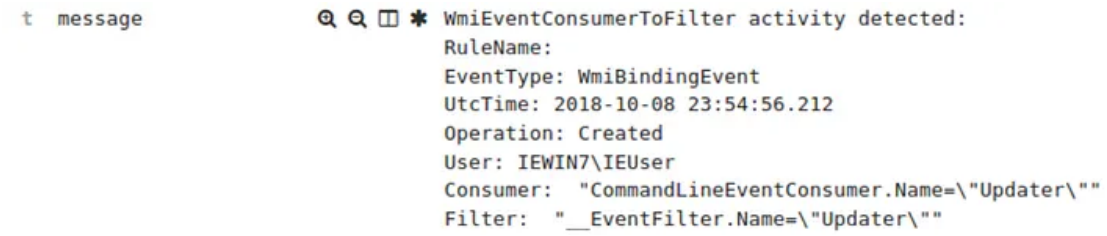
Press enter or click to view image in full size



Sysmon Event ID 20: WmiEvent (WmiEventConsumer activity detected)

The WMI event consumer `CommandLineEventConsumer.Name=\"Updater\"` is bound to the event filter `__EventFilter.Name=\"Updater\"`

Press enter or click to view image in full size



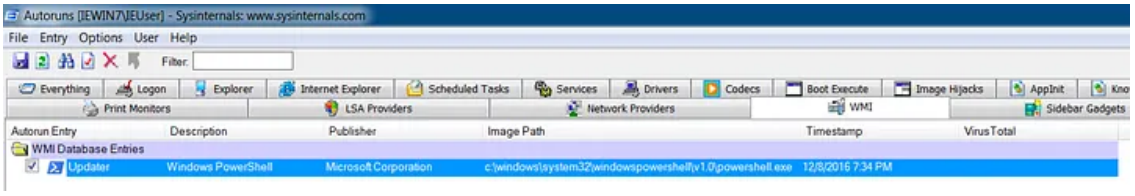
Sysmon Event ID 21: WmiEvent (WmiEventConsumerToFilter activity detected)

Now that the event consumer is bound to the event filter, **IF** the event filter conditions are true **THEN** trigger the event consumer (the stager).

Eradication

The simplest method to remove the entry from the WMI database is to use Autoruns. Launch Autoruns as an administrator and select the `WMI` tab to review WMI-related persistence.

Press enter or click to view image in full size



Using Autoruns to review WMI database entries

Press enter or click to view image in full size

Using Autoruns to review content of the WMI database

Right-click the malicious WMI database entry and select `Delete`.

Alternatively, you can remove the WMI event subscriptions from the command line.

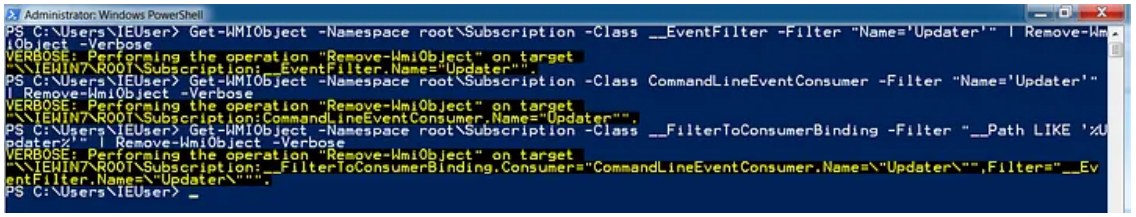
Use `Get-WMIObject` in PowerShell to review the WMI event filter, event consumer, and consumer filter to event filter binding. Thanks to Boe Prox (@proxb) for explaining these commands in detail on his blog.

```
# Reviewing WMI Subscriptions using Get-WMIObject
# Event Filter
Get-WMIObject -Namespace root\Subscription -Class __EventFilter -Filter “Name='Updater'”# Event Cons
Get-WMIObject -Namespace root\Subscription -Class CommandLineEventConsumer -Filter “Name='Updater'”

 # Binding
Get-WMIObject -Namespace root\Subscription -Class __FilterToConsumerBinding -Filter “__Path LIKE '%U
```

Use `Remove-WMIObject` to remove all components of the WMI persistence.

```
# Removing WMI Subscriptions using Remove-WMIObject
# Event Filter
Get-WMIObject -Namespace root\Subscription -Class __EventFilter -Filter “Name=’Updater’” | Remove-Wm
Get-WMIObject -Namespace root\Subscription -Class CommandLineEventConsumer -Filter “Name=’Updater’”

 # Binding
Get-WMIObject -Namespace root\Subscription -Class __FilterToConsumerBinding -Filter “__Path LIKE ‘%U
```

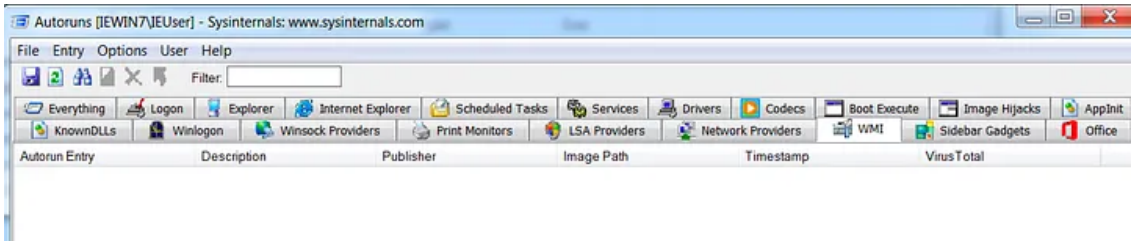
Press enter or click to view image in full size



Removing WMI event subscriptions

Run Autoruns again to verify that the persistence was removed.

Press enter or click to view image in full size



Source: https://medium.com/threatpunter/detecting-removing-wmi-persistence-60ccbb7dff96