Basic PowerShell Execution

Metadata

id	WIN-190410151110
author	Roberto Rodriguez @Cyb3rWard0g
creation date	2019/04/10
platform	Windows
playbook link	

Technical Description

Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Therefore, it is important to understand the basic artifacts left when PowerShell is used in your environment.

Hypothesis

Adversaries might be leveraging PowerShell to execute code within my environment

Analytics

Initialize Analytics Engine

```
from openhunt.mordorutils import *
spark = get_spark()
```

Download & Process Mordor File

```
mordor_file = "https://raw.githubusercontent.com/OTRF/mordor/master/data
registerMordorSQLTable(spark, mordor_file, "mordorTable")
```

```
[+] Processing a Spark DataFrame..
```

```
[+] Processing Data from Winlogbeat version 6..
[+] DataFrame Returned !
```

FP Rate	Log Channel	Description
Medium	['Microsoft-Windows-PowerShell/Operational', 'PowerShell']	Within the classic PowerShell log, event ID 400 indicates when a new PowerShell host process has started. You can filter on powershell.exe as a host application if you want to or leave it without a filter to captuer every single PowerShell host

```
df = spark.sql(
    '''
SELECT `@timestamp`, computer_name, channel
FROM mordorTable
WHERE (channel = "Microsoft-Windows-PowerShell/Operational" OR channel
    AND (event_id = 400 OR event_id = 4103)
    '''
)
df.show(10,False)
```

```
+----------------------+--------------+----------------------------
|@timestamp            |computer_name  |channel
+----------------------+--------------+----------------------------
|2019-05-18 14:20:49.575|HR001.shire.com|Windows PowerShell
|2019-05-18 14:20:50.108|HR001.shire.com|Microsoft-Windows-PowerShell/
|2019-05-18 14:20:50.963|HR001.shire.com|Microsoft-Windows-PowerShell/
|2019-05-18 14:20:50.984|HR001.shire.com|Microsoft-Windows-PowerShell/
|2019-05-18 14:20:50.989|HR001.shire.com|Microsoft-Windows-PowerShell/
|2019-05-18 14:20:51.038|HR001.shire.com|Microsoft-Windows-PowerShell/
|2019-05-18 14:20:51.287|HR001.shire.com|Microsoft-Windows-PowerShell/
|2019-05-18 14:20:51.306|HR001.shire.com|Microsoft-Windows-PowerShell/
|2019-05-18 14:20:51.341|HR001.shire.com|Microsoft-Windows-PowerShell/
|2019-05-18 14:20:51.589|HR001.shire.com|Microsoft-Windows-PowerShell/
+----------------------+--------------+----------------------------
only showing top 10 rows
```

Analytic II

FP Rate	Log Channel	Description
High	['Security']	Looking for non-interactive powershell session might be a sign of PowerShell being executed by another application in the background

```
df = spark.sql(
    '''
SELECT `@timestamp`, computer_name, NewProcessName, ParentProcessName
FROM mordorTable
WHERE channel = "Security"
    AND event_id = 4688
    AND NewProcessName LIKE "%powershell.exe"
    AND NOT ParentProcessName LIKE "%explorer.exe"
    '''
)
df.show(10,False)
```

```
+----------------------+--------------+--------------------------+
|2019-05-18 14:20:46.325|HR001.shire.com|C:\Windows\System32\WindowsPo
```

Analytic III

FP Rate	Log Channel	Description
High	['Microsoft-Windows-Sysmon/Operational']	Looking for non-interactive powershell session might be a sign of PowerShell being executed by another application in the background

```
df = spark.sql(
    '''
SELECT `@timestamp`, computer_name, Image, ParentImage
FROM mordorTable
WHERE channel = "Microsoft-Windows-Sysmon/Operational"
    AND event_id = 1
    AND Image LIKE "%powershell.exe"
    AND NOT ParentImage LIKE "%explorer.exe"
    '''
)
df.show(10,False)
```

```
+----------------------+--------------+--------------------------+
|@timestamp            |computer_name  |Image                     |
+----------------------+--------------+--------------------------+
|2019-05-18 14:20:46.353|HR001.shire.com|C:\Windows\System32\WindowsPo
+----------------------+--------------+--------------------------+
```

Analytic IV

FP Rate	Log Channel	Description
Medium	['Microsoft-Windows-Sysmon/Operational']	Monitor for processes loading PowerShell DLL *system.management.automation*

```
df = spark.sql(
    '''
SELECT `@timestamp`, computer_name, Image, ImageLoaded
FROM mordorTable
WHERE channel = "Microsoft-Windows-Sysmon/Operational"
    AND event_id = 7
    AND (lower(Description) = "system.management.automation" OR lower(Ir
    '''
)
df.show(10,False)
```

```
+----------------------+--------------+--------------------------+
|@timestamp            |computer_name  |Image                     |
+----------------------+--------------+--------------------------+
|2019-05-18 14:20:48.649|HR001.shire.com|C:\Windows\System32\WindowsPo
+----------------------+--------------+--------------------------+
```

Analytic V

	Sysmon/Operational’]	another interesting way to find PowerShell execution

```
df = spark.sql(
    '''
SELECT `@timestamp`, computer_name, Image, PipeName
FROM mordorTable
WHERE channel = "Microsoft-Windows-Sysmon/Operational"
    AND event_id = 17
    AND lower(PipeName) LIKE "\\\\pshost%"
    '''
)
df.show(10,False)
```

```
+----------------------+--------------+-------------------------
|@timestamp            |computer_name  |Image
+----------------------+--------------+-------------------------
|2019-05-18 14:20:49.334|HR001.shire.com|C:\Windows\System32\WindowsPo
+----------------------+--------------+-------------------------
```

Analytic VI

FP Rate	Log Channel	Description
Medium	[‘Microsoft-Windows-Sysmon/Operational’]	The “PowerShell Named Pipe IPC” event will indicate the name of the PowerShell AppDomain that started. Sign of PowerShell execution

```
df = spark.sql(
    '''
SELECT `@timestamp`, computer_name, message
FROM mordorTable
WHERE channel = "Microsoft-Windows-PowerShell/Operational"
    AND event_id = 53504
    '''
)
df.show(10,False)
```

```
+----------------------+--------------+-------------------------
|@timestamp            |computer_name  |message
+----------------------+--------------+-------------------------
|2019-05-18 14:20:49.47|HR001.shire.com|Windows PowerShell has started
+----------------------+--------------+-------------------------
```

Detection Blindspots

Hunter Notes

- Explore the data produced in your environment with the analytics above and document what normal looks like from a PowerShell perspective.
- If execution of PowerShell happens all the time in your environment, I suggest to categorize the data you collect by business unit to build profiles and be able to filter out potential noise.

Hunt Output

Category	Type	Name
signature	SIGMA	sysmon_powershell_execution_moduleload
signature	SIGMA	sysmon_powershell_execution_pipe
signature	SIGMA	sysmon_non_interactive_powershell_execution
signature	SIGMA	win_non_interactive_powershell

References

- https://github.com/darkoperator/Presentations/blob/master/PSConfEU%202019%20Tracking%20PowerShell
- https://posts.specterops.io/abusing-powershell-desired-state-configuration-for-lateral-movement-ca42ddbe6f06

<< WMI Win32_Process Class and Create Method for Remote Execution

Service Creation >>

By Roberto Rodriguez @Cyb3rWard0g