Table of Contents

Detection: Create Remote Thread In Shell Application

Updated Date: 2024-05-21 | ID: 10399c1e-f51e-11eb-b920-acde48001122 | Author: Teoderick Contreras, Splunk | Type: TTP

Product: Splunk Enterprise Security

Description

The following analytic detects suspicious process injection in command shell applications, specifically targeting `cmd.exe` and `powershell.exe`. It leverages Sysmon EventCode 8 to identify the creation of remote threads within these shell processes. This activity is significant because it is a common technique used by malware, such as IcedID, to inject malicious code and execute it within legitimate processes. If confirmed malicious, this behavior could allow an attacker to execute arbitrary code, escalate privileges, or maintain persistence within the environment, posing a severe threat to system security.

Search

```
SPL
`sysmon` EventCode=8 TargetImage IN ("*\\cmd.exe", "*\\powershell*")
| stats count min(_time) as firstTime max(_time) as lastTime by  TargetImage TargetProcessId SourceProcessId  EventCode StartAddress SourceImage dest
|rename SourceImage as process_name
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
| `create_remote_thread_in_shell_application_filter`
```

Data Source

Name	Platform	Sourcetype	Source	Supported App
Sysmon EventID 8	Windows	'xmlwineventlog'	'XmlWinEventLog:Microsoft-Windows-Sysmon/Operational'	N/A

Macros Used

Name	Value
security_content_ctime	`convert timeformat="%Y-%m-%dT%H:%M:%S" ctime($field$)`
create_remote_thread_in_shell_application_filter	`search *`

> `create_remote_thread_in_shell_application_filter` is an empty macro by default. It allows the user to filter out any results (false positives) without editing the SPL.

Annotations

- MITRE ATT&CK + KILL CHAIN PHASES + NIST + CIS - THREAT ACTORS

ID	Technique	Tactic
T1055	Process Injection	Defense Evasion

APT32 APT37 APT41 APT5 COBALT GROUP KIMSUKY PLATINUM SILENCE TA2541 TURLA WIZARD SPIDER

Default Configuration

This detection is configured by default in Splunk Enterprise Security to run with the following settings:

Setting	Value
Disabled	true
Cron Schedule	`0 * * * *`
Earliest Time	`-70m@m`
Latest Time	`-10m@m`
Schedule Window	`auto`
Creates Notable	Yes
Rule Title	`%name%`
Rule Description	`%description%`
Notable Event Fields	user, dest
Creates Risk Event	True

> This configuration file applies to all detections of type TTP. These detections will use Risk Based Alerting and generate Notable Events.

Implementation

To successfully implement this search, you need to be ingesting logs with the process name, parent process, and command-line executions from your endpoints. If you are using Sysmon, you must have at least version 6.0.4 of the Sysmon TA.

Known False Positives

unknown

Associated Analytic Story

- IcedID
- Qakbot
- Warzone RAT

Risk Based Analytics (RBA)

Risk Message	Risk Score	Impact	Confidence
process $process_name$ create a remote thread to shell app process $TargetImage$ in host $dest$	70	70	100

> The Risk Score is calculated by the following formula: Risk Score = (Impact * Confidence/100). Initial Confidence and Impact is set by the analytic author.

References

- https://thedfirreport.com/2021/07/19/icedid-and-cobalt-strike-vs-antivirus/

Detection Testing

Test Type	Status	Dataset	Source	Sourcetype
Validation	✅ Passing	N/A	N/A	N/A
Unit	✅ Passing	Dataset	`XmlWinEventLog:Microsoft-Windows-Sysmon/Operational`	`xmlwineventlog`
Integration	✅ Passing	Dataset	`XmlWinEventLog:Microsoft-Windows-Sysmon/Operational`	`xmlwineventlog`

Replay any dataset to Splunk Enterprise by using our `replay.py` tool or the UI. Alternatively you can replay a dataset into a Splunk Attack Range

---

Source: GitHub | Version: **3**

← Detection: Clop Ra…

Detection: Detect … →