Detecting Initial Access: HTML Smuggling and ISO Images — Part 2

mergene.medium.com/detecting-initial-access-html-smuggling-and-iso-images-part-2-f8dd600430e2

Mehmet Ergene June 1, 2021

In the previous blog in this series, we extracted behavioral TTPs, prepared the attack emulation, and executed it.

Detecting Initial Access: HTML Smuggling and ISO Images — Part 1

This blog is a two-part series focusing on TTP extraction, Attack Emulation(Purple Teaming), Log Analysis, Threat…

mergene.medium.com

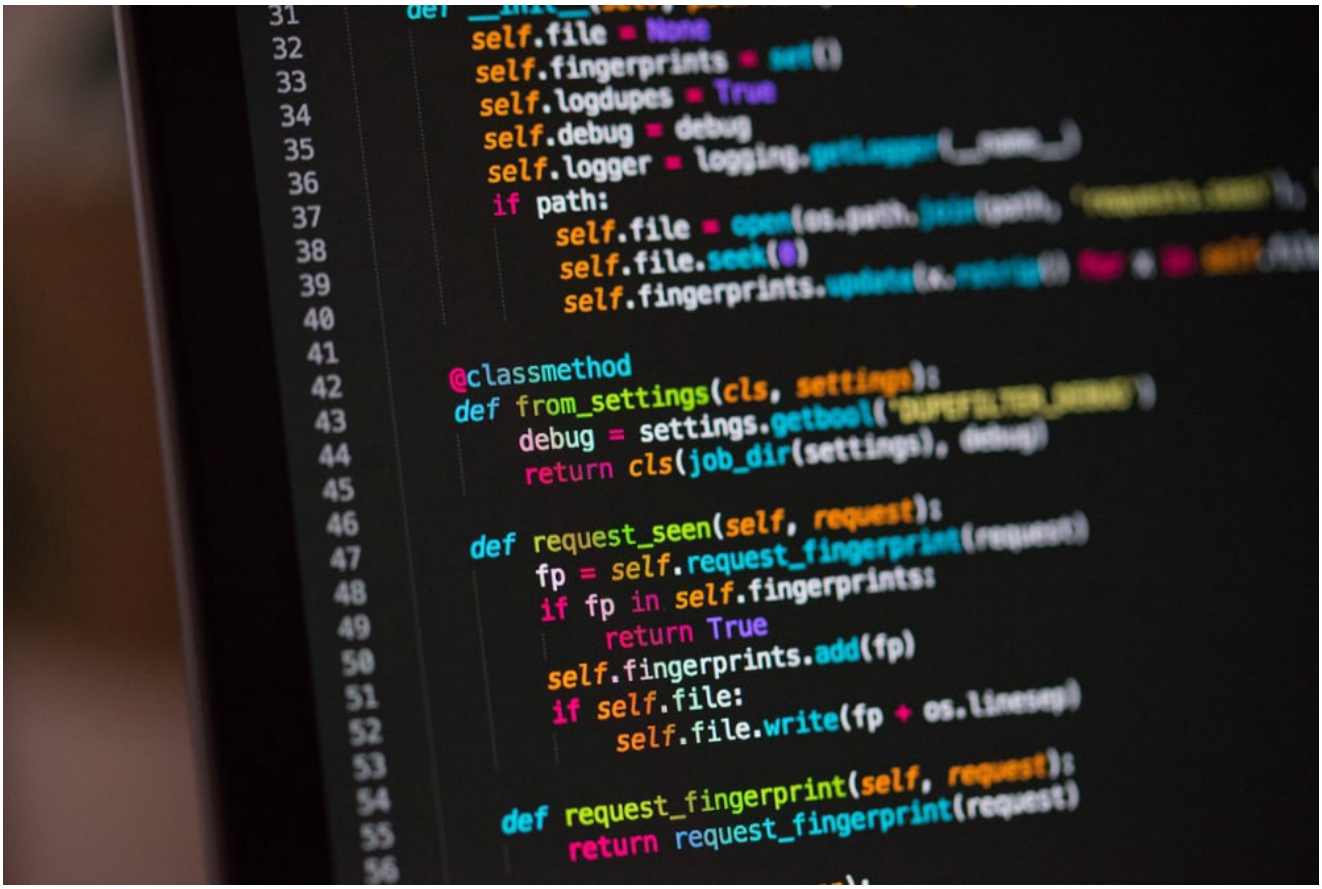

Photo by on

It's time for analyzing the logs, validating/modifying the hypotheses that we generated after reading the report(or generating new ones), generating detection strategies, and developing detections.

Analyzing the Logs

I analyzed the Microsoft Defender for Endpoint logs, but you can check Sysmon or your EDR logs. Although there can be other events generated during the attack, below are the most important ones for me to generate or validate hypotheses:

1. Mounting an ISO image generates the below Registry event:

RegistryKey	RegistryValueType	RegistryValueName
HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices	Binary	\??\Volume{7d5ec64e-c07c-11eb-8f0d-000c29415f47}
HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices	Binary	\DosDevices\F:

2. Opening the mounted image generates a file creation event:

Timestamp	ActionType	FileName	FolderPath
5/29/2021 22:18:38	FileCreated	NV.iso.lnk	C:\Users\[redacted]\AppData\Roaming\Microsoft\Windows\Recent\NV.iso.lnk
5/29/2021 22:21:15	FileCreated	nv.iso.lnk	C:\Users\[redacted]\AppData\Roaming\Microsoft\Windows\Recent\nv.iso.lnk

I mounted the ISO twice.

3. Double-clicking the shortcut file generates the below process execution events:

ActionType	FileName	FolderPath	ProcessCommandLine	InitiatingProcessFileName	InitiatingProcessCommandLine
ProcessCreated	rundll32.exe	C:\Windows\System32\rundll32.exe	"rundll32.exe" advpack.dll,RegisterOCX BOOM.exe	explorer.exe	Explorer.EXE
ProcessCreated	BOOM.exe	F:\BOOM.exe	BOOM.exe /RegServer	rundll32.exe	"rundll32.exe" advpack.dll,RegisterOCX BOOM.exe
ProcessCreated	conhost.exe	C:\Windows\System32\conhost.exe	conhost.exe 0xffffffff -ForceV1	BOOM.exe	BOOM.exe /RegServer

The process was executed under the mounted drive

4. Execution of the payload(BOOM.exe) generates a network connection event:

Timestamp	ActionType	RemoteIP	RemotePort	InitiatingProcessFileName	InitiatingProcessFolderPath	InitiatingProcessParentFileName
5/29/2021 22:36:59	ConnectionSuccess	192.168.1.102	80	BOOM.exe	f:\boom.exe	rundll32.exe

Mounted drive

Validating/Modifying the Hypotheses

I already generated some hypotheses after reading the report. Alternatively, you can generate your hypotheses after the emulation and analysis of the logs.

I intentionally skipped some hypotheses as they were fragile. For example:

- Outlook creating an HTML file (can be bypassed with a .zip)
- rundll32 execution (can be replaced by another technique)

To me, generating as few hypotheses as possible with enough coverage to detect the attack is important(less is more + Pareto). If I can cover almost all possibilities with a few hypotheses, it will make my life easy.

Going back to the attack, since we have the folder name, we can use it for finalizing our hypotheses.

Final hypotheses

1. ISO file creation is highly suspicious on non-IT users' computers
2. Process execution under a mounted drive can be highly suspicious
3. Network connection from a process that runs under a mounted drive can be highly suspicious

We need to check if these hypotheses generate hide fidelity results. To check the 2. and 3. hypotheses, we need to correlate the Registry event with process creation and network connection events. This is possible if you can query logs and generate new fields.

After analyzing the logs historically, I saw all three hypotheses were valid and would generate high fidelity results.

Creating Detections

I've created 3 queries for Microsoft Defender for Endpoint, 3 queries for Azure Sentinel (Sysmon) and published them in my GitHub repo. You can use the same logic on your own tool.

Cyb3r-Monk/Threat-Hunting-and-Detection

Repository for threat hunting and detection queries, tools, etc. - Cyb3r-Monk/Threat-Hunting-and-Detection

github.com

Conclusion

In this post, I used a different approach for TTP extraction without fully using the MITRE ATT&CK framework and wanted to show alternative ways of detecting attacks. I also wanted to show detecting Initial Access is still possible. I hope you stop assuming the breach and start hunting/detecting initial access as an additional effort in your threat management program.

I'll keep posting blogs about initial access detection. Stay tuned…