Challenge: REvil Lab

Platform: CyberDefenders

Category: Threat Hunting

Difficulty: Easy

Tools Used: ELK

Summary: This lab involves investigating a host compromised by REvil ransomware. You are required to identify the ransomware note dropped, the ransomware process, and recovery disruption techniques among other things. Whilst this lab was quite easy if you have done threat hunting challenges before, it was still extremely enjoyable.

Scenario: You are a Threat Hunter working for a cybersecurity consulting firm. One of your clients has been recently affected by a ransomware attack that caused the encryption of multiple of their employees' machines. The affected users have reported encountering a ransom note on their desktop and a changed desktop background. You are tasked with using Splunk SIEM containing Sysmon event logs of one of the encrypted machines to extract as much information as possible.

To begin your investigation, can you identify the filename of the note that the ransomware left behind?

Ransomware often leaves behind readme files that contain information about what to do next (who to contact, wallet address to send crypto to, etc). Fortunately, this environment had Sysmon enabled, therefore, we can query for event ID 11 (file create) and look for files that contain the string “readme”:

- `event.provider : "Microsoft-Windows-Sysmon" AND event.code : 11 AND winlog.event_data.TargetFilename : *readme*`

Here we can find 21 results for the host WIN-2FOSVI0LSCF, all of which are the same field name “5uizv5660t-readme.txt” in multiple folders, which is consistent with ransomware notes:

@timestamp	winlog.event_data.Image	winlog.event_data.TargetFilename	agent.name
Sep 7, 2023 @ 16:10:14.827	C:\Users\Administrator\Downloads\facebook assistant.exe	C:\Users\Public\Videos\5uizv5660t-readme.txt	WIN-2FOSVI0LSCF
Sep 7, 2023 @ 16:10:14.826	C:\Users\Administrator\Downloads\facebook assistant.exe	C:\Users\Public\Music\5uizv5660t-readme.txt	WIN-2FOSVI0LSCF
Sep 7, 2023 @ 16:10:14.826	C:\Users\Administrator\Downloads\facebook assistant.exe	C:\Users\Public\Pictures\5uizv5660t-readme.txt	WIN-2FOSVI0LSCF
Sep 7, 2023 @ 16:10:14.825	C:\Users\Administrator\Downloads\facebook assistant.exe	C:\Users\Public\Libraries\5uizv5660t-readme.txt	WIN-2FOSVI0LSCF

You can also see the image responsible for creating these files, which is likely the ransomware binary.

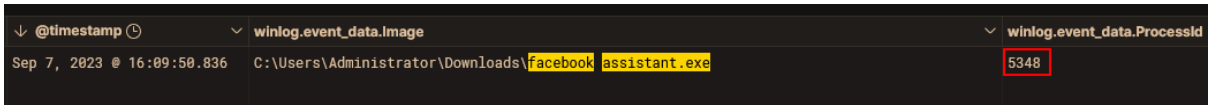
Answer: 5uizv5660t-readme.txt

After identifying the ransom note, the next step is to pinpoint the source. What's the process ID of the ransomware that's likely involved

From the previous question, we identified that a binary called “facebook assistant.exe” was responsible for creating the readme files. We can pivot off this by querying for process creation events (Event ID 1) for this image:

- `event.provider : "Microsoft-Windows-Sysmon" AND event.code : 1 AND winlog.event_data.Image : *facebook assistant.exe*`

If you look at the winlog.event_data.Processid field, you can find the PID for this process:



@timestamp	winlog.event_data.Image	winlog.event_data.ProcessId
Sep 7, 2023 @ 16:09:50.836	C:\Users\Administrator\Downloads\facebook assistant.exe	5348

Answer: 5348

Having determined the ransomware's process ID, the next logical step is to locate its origin. Where can we find the ransomware's executable file?

As seen in the previous image, you can see that this binary is located in the Administrator’s Downloads folder.

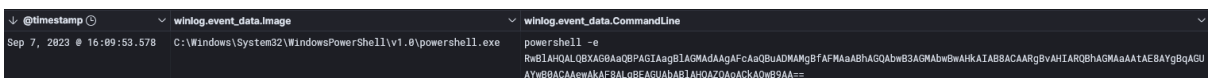
Answer: C:\Users\Administrator\Downloads\facebook assistant.exe

Now that you've pinpointed the ransomware's executable location, let's dig deeper. It's a common tactic for ransomware to disrupt system recovery methods. Can you identify the command that was used for this purpose?

Let’s start by hunting for processes spawned by “facebook assistant.exe”:

- `event.provider : "Microsoft-Windows-Sysmon" AND event.code : 1 AND winlog.event_data.ParentImage : *facebook assistant.exe*`

This outputs one result, indicating that “facebook assistant.exe” spawned a PowerShell process, which executed a Base64 encoded command:



@timestamp	winlog.event_data.Image	winlog.event_data.CommandLine
Sep 7, 2023 @ 16:09:53.578	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	powershell -e RwBlAHQALQBXAG0AaQBPAGIAagBlAGMAdAAgAFcAaQBuADMAMgBfAFMAaABhAGQAbwB3AGMAbwBwAHkAIAB8ACAARgBvAHIARQBhAGMAaAAtAE8AYgBqAGUAYwB0ACAAewAkAF8ALgBEAGUAbABlAHQAZQAoACkAOwB9AA==

Using CyberChef to decode this command, we can see that it uses WMI to delete all Volume Shadow Copies:

Volume Shadow Copies are backups of files and folders, enabling users to revert to previous versions if a file is corrupted or lost. Ransomware often deletes these copies to prevent recovery efforts.

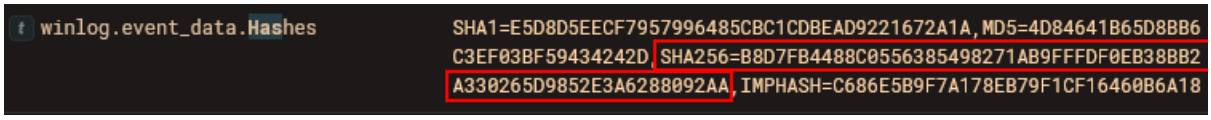
Answer: Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();}

As we trace the ransomware's steps, a deeper verification is needed. Can you provide the sha256 hash of the ransomware's executable to cross-check with known malicious signatures?

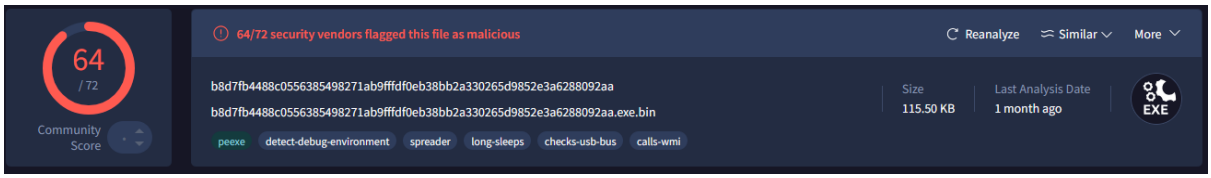
Thankfully, Sysmon records the SHA1, MD5, SHA256, and IMPHASH for each process creation event:

- `event.provider : "Microsoft-Windows-Sysmon" AND event.code : 1 AND winlog.event_data.Image : *facebook assistant.exe*`

In the winlog.event_data.Hashes field, you can find the SHA256 hash of the ransomware's executable:



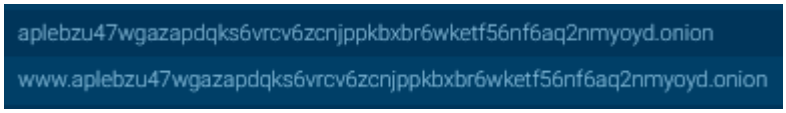
If you submit this hash to VirusTotal, we can see that it receives a high detection rate:



Answer: B8D7FB4488C0556385498271AB9FFFDF0EB38BB2A330265D9852E3A6288092AA

One crucial piece remains: identifying the attacker's communication channel. Can you leverage threat intelligence and known Indicators of Compromise (IoCs) to pinpoint the ransomware author's onion domain?

If you go through AnyRun reports for the ransomware identified previously, you can see it make a DNS request to a .onion site:



Answer: aplebzu47wgazapdqks6vrcv6zcnjppkbxbr6wketf56nf6aq2nmyoyd.onion