ProcNet - Walkthrough

Wednesday, August 21, 2024 10:55 AM

Story:
With the rising utilization of open-source C2 frameworks by threat actors, our red team has simulated the functionalities of one such widely employed framework.
The objective of this exercise is to aid blue teams in strengthening their defenses against these specific threats.
We have been provided with PCAP files and APIs collected during the event, which will serve as valuable resources.
Using the API Monitor: We are well-acquainted with opening PCAP and .EVTX files, but what are .apmx64 ? The .apmx64 file extension is associated with API Monitor, a software used to monitor and control API calls made by applications and services.
To commence your analysis, follow the steps provided below: Download the API Monitor Navigate to "Files" and click on "Open" to view captured data from the file: "Employee.apmx64" or "DC01.apmx64"
After opening the file, the "Monitoring Process" window will populate with a list of processes.
Expand the view by clicking the '+' symbol to reveal the modules and threads associated with each process.
The API calls can be observed in the "Summary" window.
To focus our analysis on a specific module, click on the different DLLs loaded by the processes. TIP: When conducting analysis, it is advisable to begin by examining the API calls made by the process itself, rather than focusing solely on DLLs.
For instance, if I intend to analyze the API calls of a process named csgo.exe, I will initially expand the view by clicking the '+' symbol.
Then, I will narrow down my analysis specifically to 'csgo.exe' by selecting it, and I can further analyze other DLLs as needed.

Task1: To which IP address and port number is the malware attempting to establish a connection ?

I opened Sysmon logs of the employee using 'Event Log Explorer' and filtered for event ID '3', which relates to network connections.
While navigating through the logs, I discovered that the process 'csgo.exe', located in the Downloads folder, established several connections to the IP address '3.6.165.8' on port 443.

```
The description for Event ID ( 3 ) in Source ( Microsoft-Windows-Sysmon ) could not
be found.
Either the component that raises this event is not installed on the computer or the
installation is corrupted.You can install or repair the component or try to change
Description Server.

The following information was included with the event:
-
2023-05-22 07:52:07.246
{5080714d-1856-646b-df01-000000000a00}
6148
C:\Users\alonzo.spire\Downloads\csgo.exe
FORELA\alonzo.spire
tcp
true
false
10.10.0.79
-
50017
-
false
3.6.165.8
-
443
-
```

Task2: Now that you are aware of the IP address and port number, what is the JA3 fingerprint of the C2 server ?

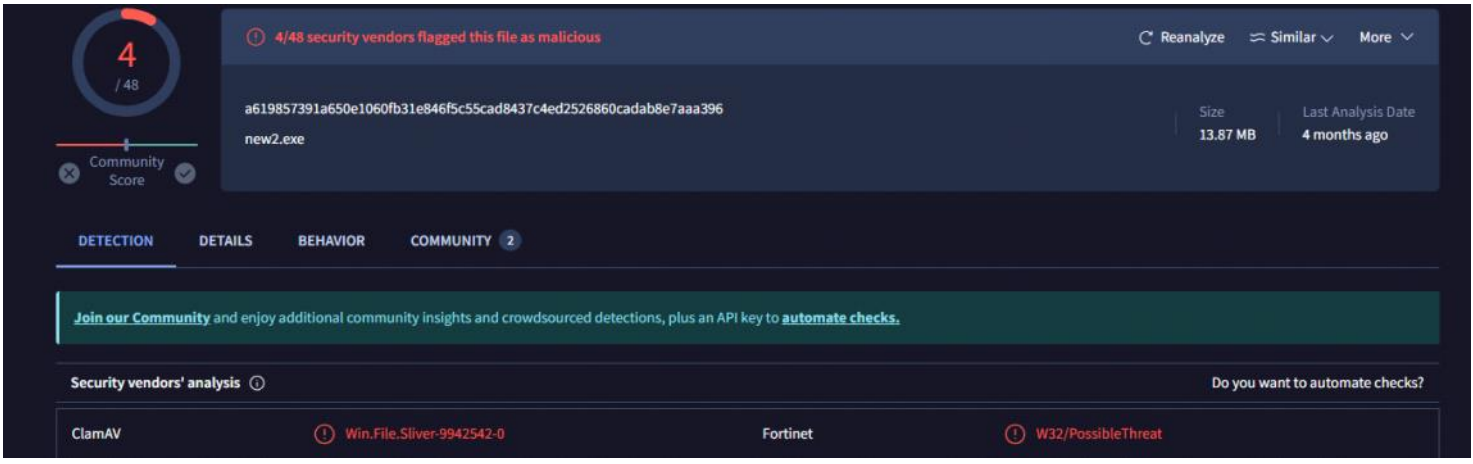
JA3 is a method for creating a unique fingerprint of SSL/TLS clients based on the specific settings and options they use when initiating a TLS handshake.

- To address this question, I opened the PCAP file of the employee and filtered the C2 IP and port, found the TLSv1.3 handshake.
  In the packet you able to find the answer:

```
Version: TLS 1.2 (0x0303)
Random: d7d2912caad08fa9addcb6f565e70dfcb67f10b479e7db629ec978c89149087a
Session ID Length: 32
Session ID: a2db3c3ca78dfc52898ae8f5c425e47423426a3ca8b94e4de4a3ad2595cb27b1
Cipher Suites Length: 38
> Cipher Suites (19 suites)
Compression Methods Length: 1
> Compression Methods (1 method)
Extensions Length: 119
> Extension: status_request (len=5)
> Extension: supported_groups (len=10)
> Extension: ec_point_formats (len=2)
> Extension: signature_algorithms (len=26)
> Extension: renegotiation_info (len=1)
> Extension: signed_certificate_timestamp (len=0)
> Extension: supported_versions (len=5) TLS 1.3, TLS 1.2
> Extension: key_share (len=38) x25519
[JA4: t13i190800_9dc949149365_97f8aa674fd9]
[JA4_r: t13i190800_000a,002f,0035,009c,009d,1301,1302,1303,c009,c00a,c012,c013,c014,c02b,c02c,c02f,c030,cca8
[JA3 Fullstring: 771,49195-49199-49196-49200-52393-52392-49161-49171-49162-49172-156-157-47-53-49170-10-4865
[JA3: 19e29534fd49dd27d09234e639c4057e]
```

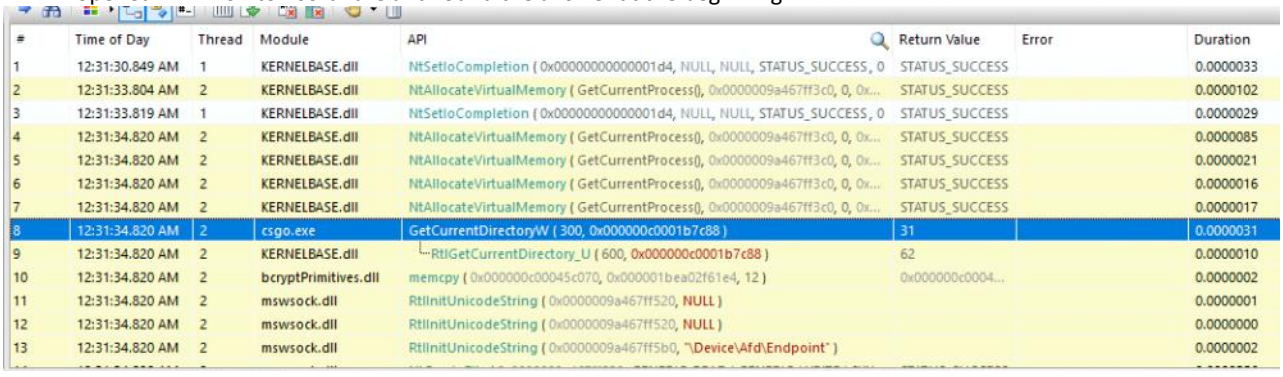
Task3: What is the name of the C2 framework being utilized by the red team ?

- Using Wireshark, I filtered the IP associated with the C2 server and identified a GET request linked to the 'csgo.exe' executable, which was tied to the C2.
  I then extracted all the TCP streams and uploaded them to VirusTotal, where I discovered that the software was related to the 'Sliver' C2 framework.



Task4: Which WIN32 API provided the red team with the current directory information ?

- I opened 'API Monitor' software and found the answer at the beginning:



#	Time of Day	Thread	Module	API	Return Value	Error	Duration
1	12:31:30.849 AM	1	KERNELBASE.dll	NtSetIoCompletion (0x00000000000001d4, NULL, NULL, STATUS_SUCCESS, 0	STATUS_SUCCESS		0.0000033
2	12:31:33.804 AM	2	KERNELBASE.dll	NtAllocateVirtualMemory (GetCurrentProcess(), 0x0000009a467ff3c0, 0, 0x...	STATUS_SUCCESS		0.0000102
3	12:31:33.819 AM	1	KERNELBASE.dll	NtSetIoCompletion (0x00000000000001d4, NULL, NULL, STATUS_SUCCESS, 0	STATUS_SUCCESS		0.0000029
4	12:31:34.820 AM	2	KERNELBASE.dll	NtAllocateVirtualMemory (GetCurrentProcess(), 0x0000009a467ff3c0, 0, 0x...	STATUS_SUCCESS		0.0000085
5	12:31:34.820 AM	2	KERNELBASE.dll	NtAllocateVirtualMemory (GetCurrentProcess(), 0x0000009a467ff3c0, 0, 0x...	STATUS_SUCCESS		0.0000021
6	12:31:34.820 AM	2	KERNELBASE.dll	NtAllocateVirtualMemory (GetCurrentProcess(), 0x0000009a467ff3c0, 0, 0x...	STATUS_SUCCESS		0.0000016
7	12:31:34.820 AM	2	KERNELBASE.dll	NtAllocateVirtualMemory (GetCurrentProcess(), 0x0000009a467ff3c0, 0, 0x...	STATUS_SUCCESS		0.0000017
8	12:31:34.820 AM	2	csgo.exe	GetCurrentDirectoryW (300, 0x000000c0001b7c88)	31		0.0000031
9	12:31:34.820 AM	2	KERNELBASE.dll	RtlGetCurrentDirectory_U (600, 0x000000c0001b7c88)	62		0.0000010
10	12:31:34.820 AM	2	bcryptPrimitives.dll	memcpy (0x000000c00045c070, 0x000001bea02f61e4, 12)	0x000000c0004...		0.0000002
11	12:31:34.820 AM	2	mswsock.dll	RtlInitUnicodeString (0x0000009a467ff520, NULL)			0.0000001
12	12:31:34.820 AM	2	mswsock.dll	RtlInitUnicodeString (0x0000009a467ff520, NULL)			0.0000000
13	12:31:34.820 AM	2	mswsock.dll	RtlInitUnicodeString (0x0000009a467ff5b0, "\Device\Afd\Endpoint")			0.0000002

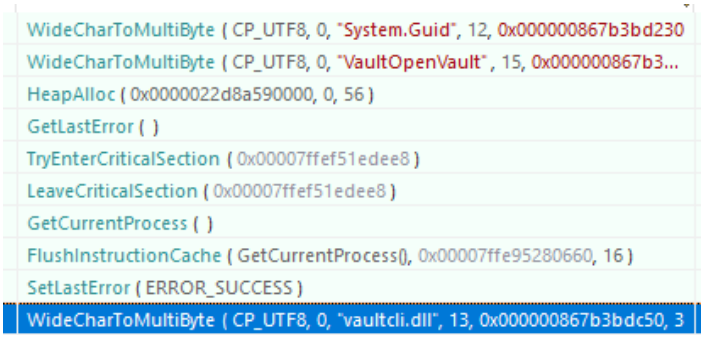
Task5: Now that we have identified the C2 framework utilized by the red team, which C2 command is responsible for opening notepad.exe by default and loading the .NET CLR into it ?

- To tackle this question, I initially attempted to find the answer using API Monitor software, but was unsuccessful. After reading about the Sliver C2 framework, I discovered that it commonly performs DLL side-loading or injection. I found the answer in an article by CyberReason, which discusses how Sliver C2 is leveraged by many threat actors.
  You can read more about it here: CyberReason Article on Sliver C2.|

  Answer:
  Execute-Assembly

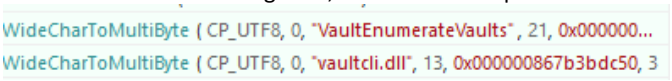
Task6: What is the name of the module (DLL) that was loaded to gain access to Windows Vault ?

- Via API Monitor I trace the API calls and searched 'vault' and found the 'dll'



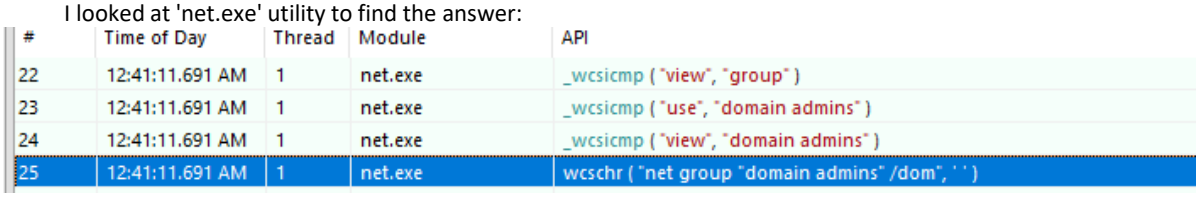
Task7: After loading the mentioned module, there were a series of WIN32 APIs loaded. Which specific Win32 API is responsible for enumerating vaults ?

- After further investigation, I identified a suspicious API call that raised enumeration process:



Task8: Which command did the attacker execute to identify domain admins ?

- To address this question, I used API monitor again.
  I looked at 'net.exe' utility to find the answer:



#	Time of Day	Thread	Module	API
22	12:41:11.691 AM	1	net.exe	_wcsicmp ("view", "group")
23	12:41:11.691 AM	1	net.exe	_wcsicmp ("use", "domain admins")
24	12:41:11.691 AM	1	net.exe	_wcsicmp ("view", "domain admins")
25	12:41:11.691 AM	1	net.exe	wcschr ("net group "domain admins" /dom", ' ')

Task9: The red team has provided us with a hint that they utilized one of the tools from "ARMORY" for lateral movement to DC01. What is the name of the tool ?

- While analyzing the DLLs loaded by the 'csgo.exe' process, I discovered that 'wminet_utils.dll' was among them, indicating the use of WMI. Investigating further, I identified the API call 'ExecQueryWMI', which is associated with WMI operations. Upon searching for related strings, I confirmed that the tool used was SharpWMI.

  Answer: SharpWMI

Task10:Which command was executed by the red team to extract/dump the contents of NTDS.DIT ?

- To address this question, I opened 'API Montior' of the DC and found the suspicious process was 'fifa24.exe', and the suspicious process used PowerShell.exe and Cmd.exe.
  I accessed the Sysmon logs of the DC and searched Cmd.exe via event ID 1.
  And found the suspicious command:

The description for Event ID (1) in Source (Microsoft-Windows-Sysmon) could not be found.
Either the component that raises this event is not installed on the computer or the installation is corrupted.You can install or repair the component or try to change Description Server.

The following information was included with the event:
-
2023-05-22 07:55:26.425
{524bc579-1fee-646b-5601-000000000900}
7332
C:\Windows\System32\cmd.exe
10.0.14393.0 (rs1_release.160715-1616)
Windows Command Processor
Microsoft® Windows® Operating System
Microsoft Corporation
Cmd.Exe
cmd /c ntdsutil "ac in ntds" ifm "cr fu %%TEMP%%\H00i0Z000.dat" q q
C:\Windows\system32\
NT AUTHORITY\SYSTEM
{524bc579-15c0-646a-e703-000000000000}
0x3e7
0
System
SHA1=99AE9C73E9BEE6F9C76D6F4093A9882DF06832CF,MD5
=F4F684066175B77E0C3A000549D2922C,SHA256=
935C1861DF1F4018D698E8B65ABFA02D7E9037D8F68CA3C2065B6CA165D44AD2,IMPHASH
=3062ED732D4B25D1C64F084DAC97D37A
{524bc579-1d93-646b-5101-000000000900}
1132
C:\Windows\Temp\fifa24.exe
"C:\Windows\TEMP\fifa24.exe"
NT AUTHORITY\SYSTEM

Task11: The red team has obtained the aforementioned dump by compressing it into a ZIP file. Which specific Win32 API is responsible for retrieving the full path of the file to be downloaded?

- **GetFullPathNameW** function take a relative file path and return the full path by combining it with the current directory or any other necessary components.