Blue Team Labs Online: Anakus

The following writeup is for Anakus on Blue Team Labs Online, it's an easy lab that involves analysing a malicious binary.

Scenario: Loda Sukana, that's me! My big brother Desi Sukana—who's currently based in Australia and working at Tesserent as a Senior DFIR Analyst—sparked my interest in cybersecurity! Yasss, I know, I know, everyone is trying to break into the field, the market is rough, companies are not hiring.... blah, blah, blah! Enough of the doom and gloom, I am going to break in another way—interestingly, hehe.

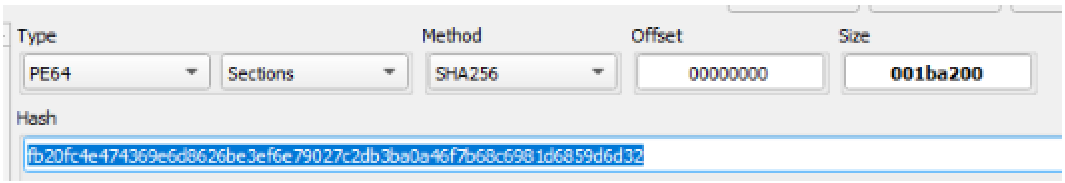
My brother has managed to pull some strings, which got me a referral to a Malware Analyst role at his company. They respect him a lot over there, but he still doesn't trust me... his little sister. He thinks I am going to bring shame to the family name "Sukana" if I don't perform well on this OA. Shhh, I have not told him this: I have been practicing my reverse engineering skills this whole time, haha.

Ok, I got off the phone with the recruiter; his name was Mr. Gula. Not going to cap, but he sounded so hot!!! Oh, gosh, sorry! I should erase that from my journal lol. Back to the task: it looks like the OA will take place virtually, and I will be given a VM, with plenty of tools to analyze "Threat logs" and "Malware(s)".

All right, final entry in my journal before I get started! I may not be as talented as my brother Desi, but I am determined to pass this OA and gain his trust. I know everything changed when Dad died—regardless, I will not bring shame to the Sukana family! Watch me, Desi.

Q1) Using Detect it Easy, what is the SHA256 hash of the malware in question, and what language was it written in? Feel free to utilize VirusTotal to garner more information (Format: SHA256, Language)

The malware in question is the dll file not the exe, to generate the hash using DIE, all you need to do is select the advanced checkbox and then go to Hash > Method = SHA256:



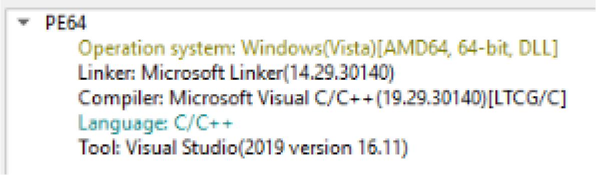
Type: PE64, Sections

Method: SHA256

Offset: 00000000

Size: 001ba200

Hash: fb20fc4e474369e6d8626be3ef6e79027c2db3ba0a46f7b68c6981d6859d6d32

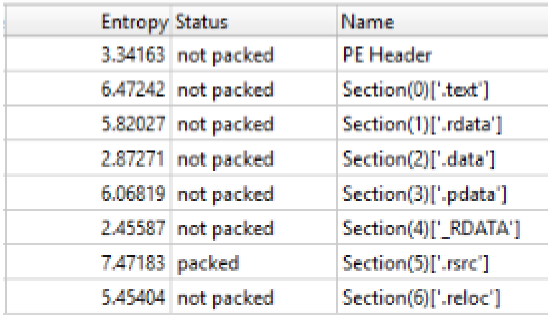


```
PE64
    Operation system: Windows(Vista)[AMD64, 64-bit, DLL]
    Linker: Microsoft Linker(14.29.30140)
    Compiler: Microsoft Visual C/C++(19.29.30140)[LTCG/C]
    Language: C/C++
    Tool: Visual Studio(2019 version 16.11)
```

Therefore, the answer is fb20fc4e474369e6d8626be3ef6e79027c2db3ba0a46f7b68c6981d6859d6d32, C/C++

Q2) The total entropy value in Detect it Easy gives us a general indication of the randomness across the entire file, but the presence of a highly entropic-packed section indicates a portion of the file containing data that has been compressed—packed. Usually, an entropy above 7.2 is considered malicious, what is the name of this section in question? (Format: .xxxx)

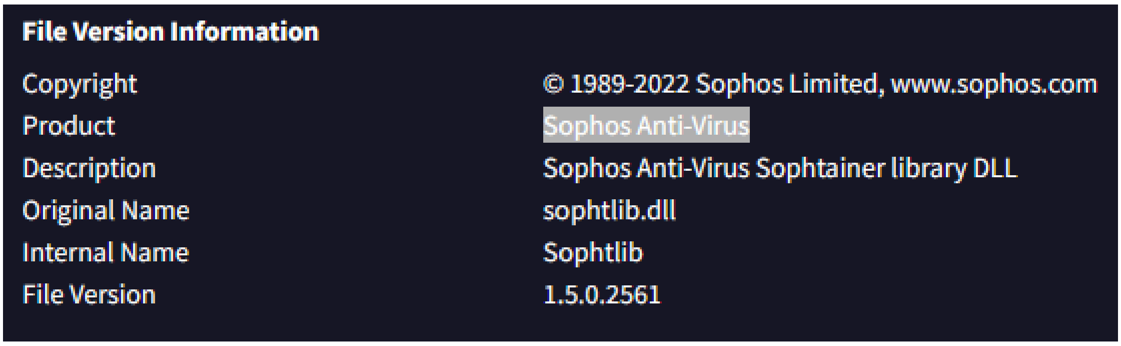
If you click on the Entropy button, we can see that the resource (.rsrc) section has an entropy of 7.47183 which is extremely suspicious and highly indicative of a packed second stage:



Entropy	Status	Name
3.34163	not packed	PE Header
6.47242	not packed	Section(0)['.text']
5.82027	not packed	Section(1)['.rdata']
2.87271	not packed	Section(2)['.data']
6.06819	not packed	Section(3)['.pdata']
2.45587	not packed	Section(4)['_RDATA']
7.47183	packed	Section(5)['.rsrc']
5.45404	not packed	Section(6)['.reloc']

Q3) Given the file's entropy level, reputation on VirusTotal, and weird characteristics, it is clear that is malicious. However, attackers will sometimes use the names of security companies in their malware to bypass detection. In this case, what product name is this malware impersonating? (Format: Product Name)

If you look at the details section in VirusTotal and check out the File Version Information, we can see that it is impersonating Sophos Anti-Virus:

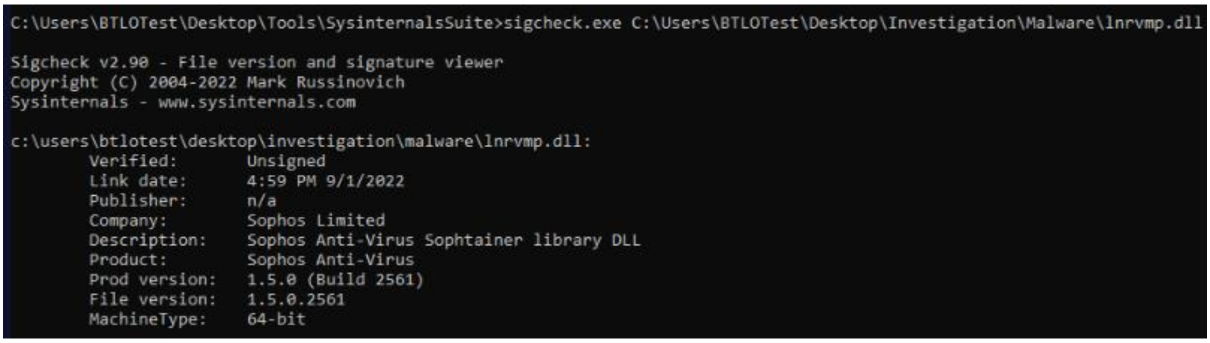


File Version Information

Copyright	© 1989-2022 Sophos Limited, www.sophos.com
Product	Sophos Anti-Virus
Description	Sophos Anti-Virus Sophtainer library DLL
Original Name	sophtlib.dll
Internal Name	Sophtlib
File Version	1.5.0.2561

Q4) Using SigCheck, let's see if the file has been signed with a code-signing certificate—proving its validity. Include the "Verified" status and "Signing date" also known as Link Date (Format: Status, X:XX PM X/X/XXXX)

SigCheck is a tool part of the sysinternals suite. All you need to do is execute the script and provide the filename of the binary you want to check:



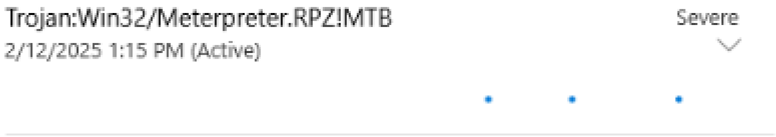
```
C:\Users\BTLOTest\Desktop\Tools\SysinternalsSuite>sigcheck.exe C:\Users\BTLOTest\Desktop\Investigation\Malware\lnrvmp.dll

Sigcheck v2.90 - File version and signature viewer
Copyright (C) 2004-2022 Mark Russinovich
Sysinternals - www.sysinternals.com

c:\users\btlotest\desktop\investigation\malware\lnrvmp.dll:
        Verified:       Unsigned
        Link date:      4:59 PM 9/1/2022
        Publisher:      n/a
        Company:        Sophos Limited
        Description:    Sophos Anti-Virus Sophtainer library DLL
        Product:        Sophos Anti-Virus
        Prod version:   1.5.0 (Build 2561)
        File version:   1.5.0.2561
        MachineType:    64-bit
```

With this information, the answer is Unsigned,4:59 PM 9/1/2022

Q5) Let's look at the alleged malware "hataker.exe". Judging from its hash, it is not apparent on most malware platforms—albeit, it does not exclude its maliciousness. Turn on Windows Defender and run the malware until it picks it up. What is the name given to this alleged, malicious trojan-type program? (Format: Trojan:full name)



Trojan:Win32/Meterpreter.RPZ!MTB
2/12/2025 1:15 PM (Active)
Severe

Trojan:Win32/Mterpreter.RPZ!MTB

Q6) Interestingly, now knowing the trojan type for the "hataker.exe", we can safely assume the attacker was planning to initiate a connection from the victim's endpoint to its command and control server. What is this method called? (Format: Xxxxxxx Xxxxx)

This method is called a reverse shell.

Q7) Sticking to the same context, let's dissect the malware further. What dynamic domain is the malware "hataker.exe" using to establish a connection back to the attacker's system? (Format: Domain name)

If you look at the strings in memory using a tool like Process Explorer, we can see a reference to 0.tcp.ngrok.io:

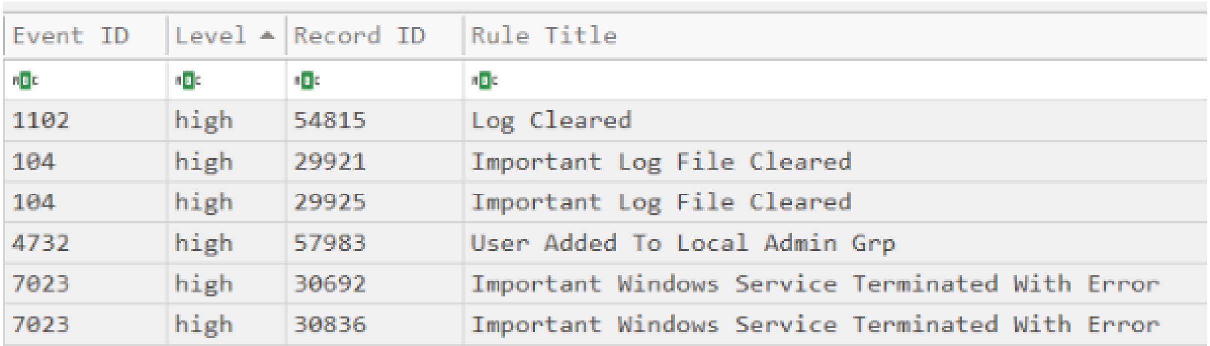
0.tcp.ngrok.io

0.tcp.ngrok.io

You can also launch Wireshark and look for resolved addresses.

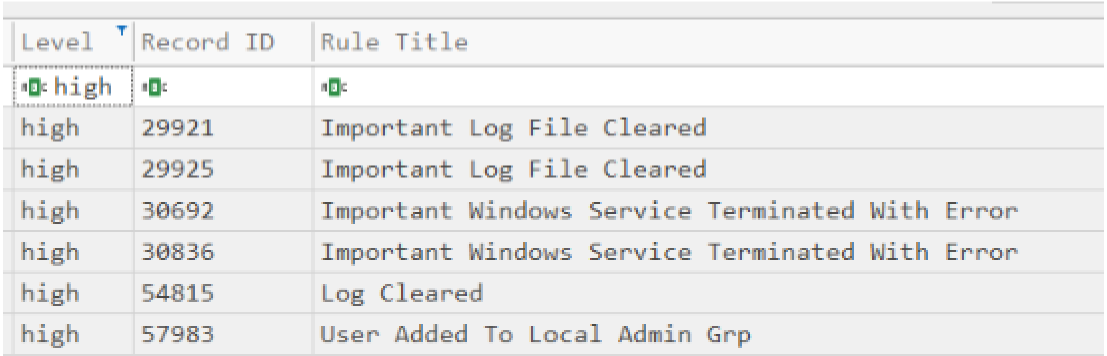
Q8) Using Timeline Explorer, look at the "Threat Logs" from the incident, how many high-risk alerts were tracked? (Format: Count)

There are 6 high-risk alerts within the log file:



Event ID	Level ▲	Record ID	Rule Title
1102	high	54815	Log Cleared
104	high	29921	Important Log File Cleared
104	high	29925	Important Log File Cleared
4732	high	57983	User Added To Local Admin Grp
7023	high	30692	Important Windows Service Terminated With Error
7023	high	30836	Important Windows Service Terminated With Error

Q9) What are the two “Rule Titles” with the highest count under the high-risk alerts level group—in respective order? (Format: Rule Title 1, Rule Title 2)



Level	Record ID	Rule Title
high		
high	29921	Important Log File Cleared
high	29925	Important Log File Cleared
high	30692	Important Windows Service Terminated With Error
high	30836	Important Windows Service Terminated With Error
high	54815	Log Cleared
high	57983	User Added To Local Admin Grp

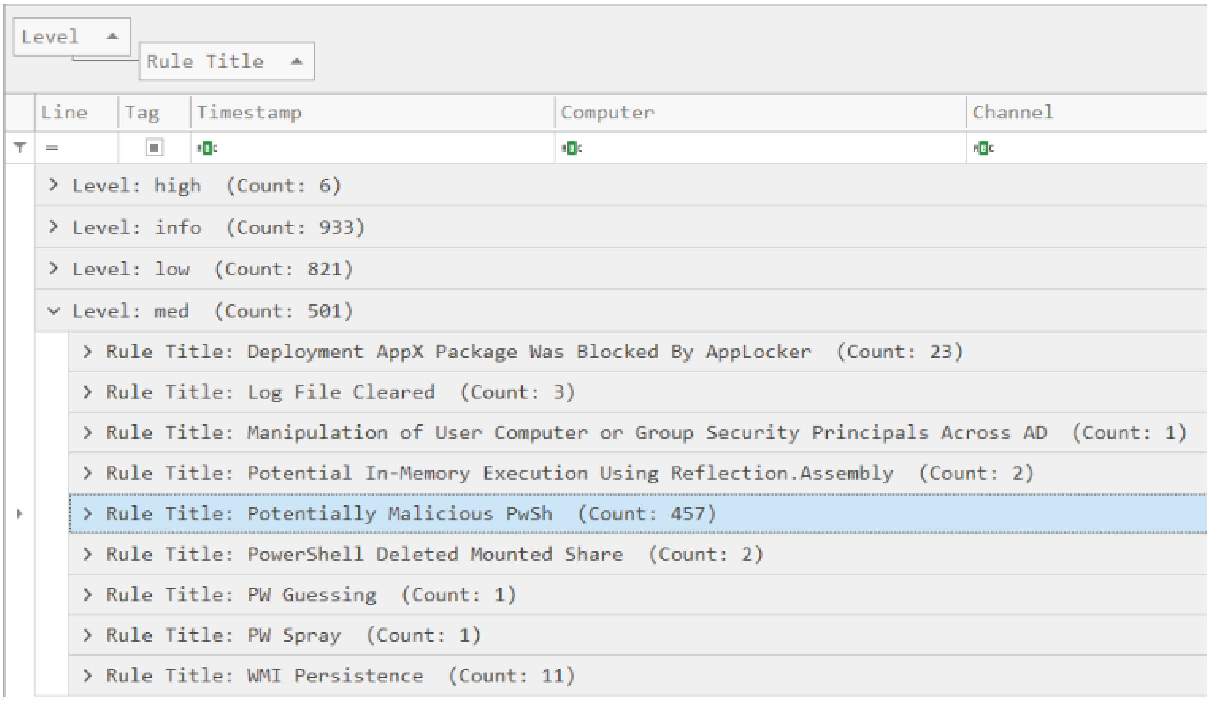
important log file cleared,impotrnat windows service terminared with error

Q10) Examine the last “Rule Title” for the high-risk alerts level group: what MITRE ID does this correspond to and what is the TgtGrp in question? (Format: TechniqueID, TgtGrp)

The last Rule Title is “User Added To Local Admin Grp”. If you search this up and append MITRE, you will see a reference to Account Manipulation (T1098). Therefore, the answer is: T1098,administrators

Q11) Looking at the medium-risk level alerts, what is the “Rule Title” and count of the popular alert group? (Format: Rule Title, Count)

To find the answer, we can group by what we are concerned with. In this case, its medium-risk alerts:



```
Level ▲
    Rule Title ▲
Line | Tag | Timestamp | Computer | Channel
> Level: high  (Count: 6)
> Level: info  (Count: 933)
> Level: low  (Count: 821)
v Level: med  (Count: 501)
    > Rule Title: Deployment AppX Package Was Blocked By AppLocker  (Count: 23)
    > Rule Title: Log File Cleared  (Count: 3)
    > Rule Title: Manipulation of User Computer or Group Security Principals Across AD  (Count: 1)
    > Rule Title: Potential In-Memory Execution Using Reflection.Assembly  (Count: 2)
    > Rule Title: Potentially Malicious PwSh  (Count: 457)
    > Rule Title: PowerShell Deleted Mounted Share  (Count: 2)
    > Rule Title: PW Guessing  (Count: 1)
    > Rule Title: PW Spray  (Count: 1)
    > Rule Title: WMI Persistence  (Count: 11)
```

Potentially Malicious PwSh, 457

Q12) What function was used for the password spray attack? Lists its MITRE ID for this type of attack as well (Format: powershell-function, TXXXX.xxx)

Invoke-LocalPasswordSpray,T11110.003