EXTRACTING STRINGS FROM MALWARE FOR THE PURPOSE OF CREATING YARA RULES

Jordan Patterson

2024

Goals of this Lab:

- Create yara rules for the non-english files in Oct 2019 link
- Present the methodology with screenshots in a word report
- All rules must be created from memory dump analysis and must be in working order
- Report must specifically identify your file detections
- All screenshots must have explanations

Submission items:

- Word report
- Thorlite Output HTML reports from both Windows and Kali

I began this lab with my Windows VM. I took a snapshot before beginning. I downloaded the required malware, connected to the VPN, ensured that real-time protection was off, and ensured that FTK Imager was open and ready to capture the memory of the system after the malware had been run.

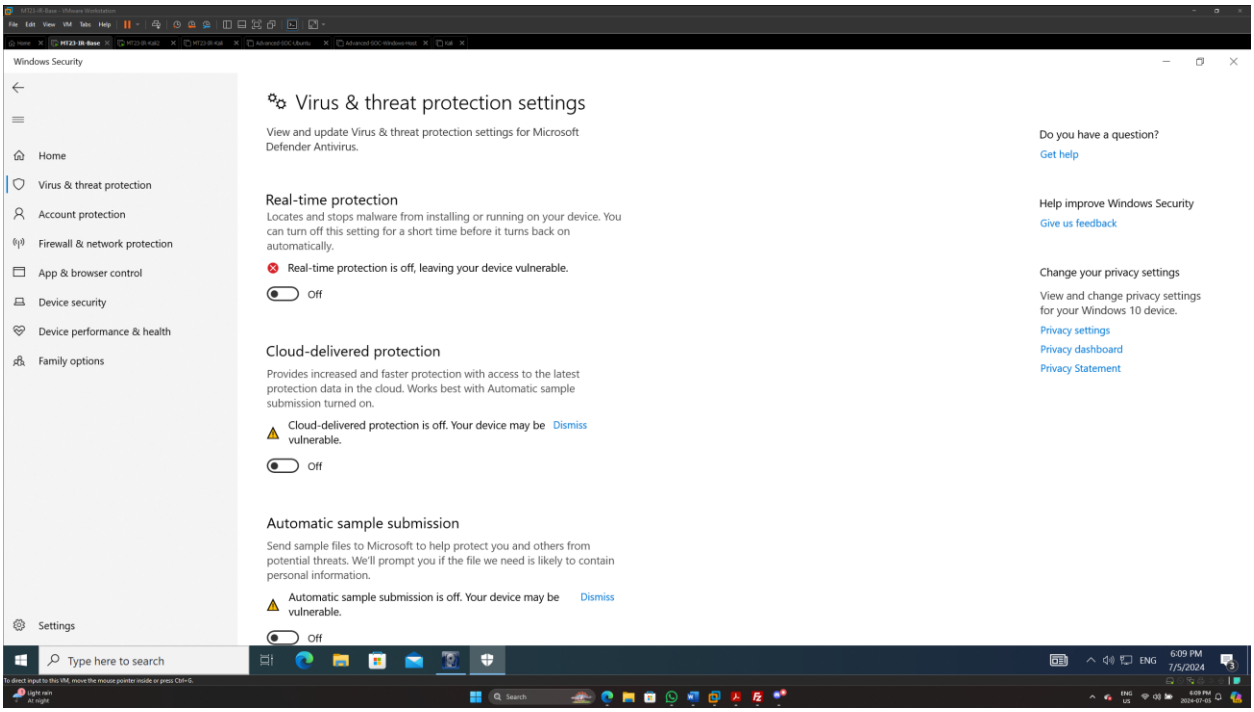


Figure 1 - Screenshot showing part of the preparation process which includes ensure that Real-time protection is off.

After ensuring that my environment was ready and secure, I ran the 3 non-English malware files pictured below as administrator and allowed them to run for a few minutes.

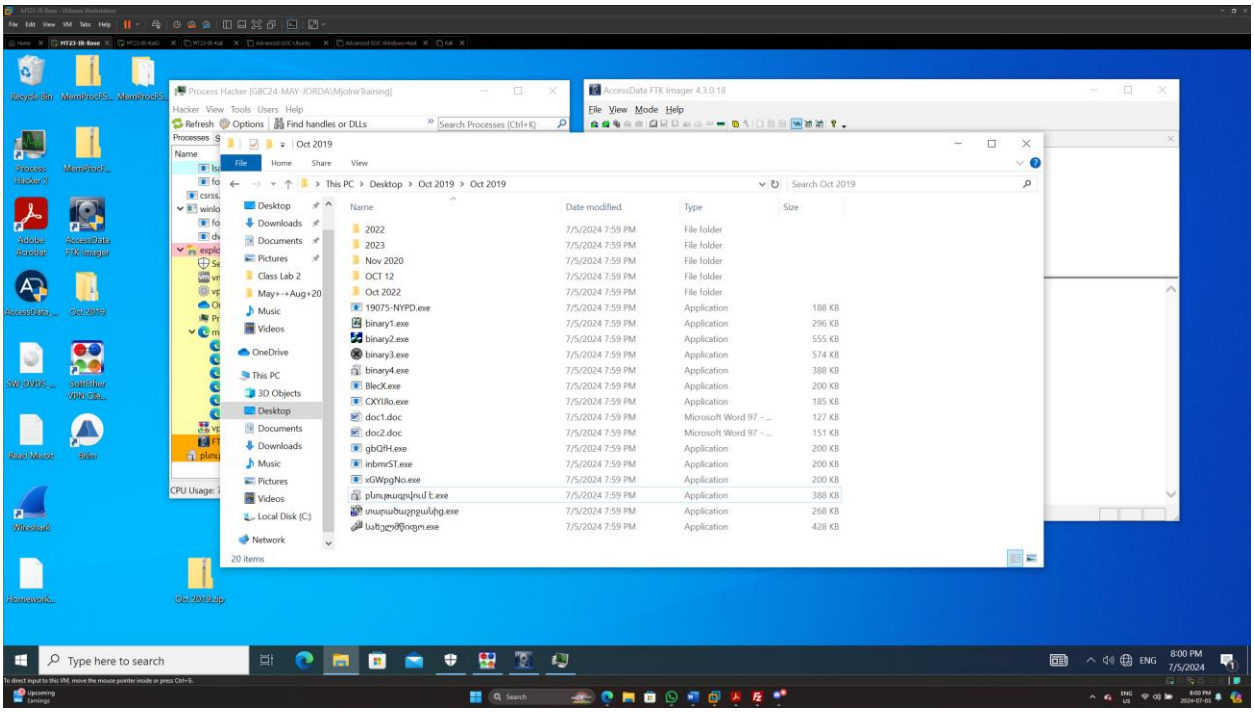


Figure 2 - Screenshot showing non-English malware

After several minutes of run-time, I captured the memory using FTK Imager

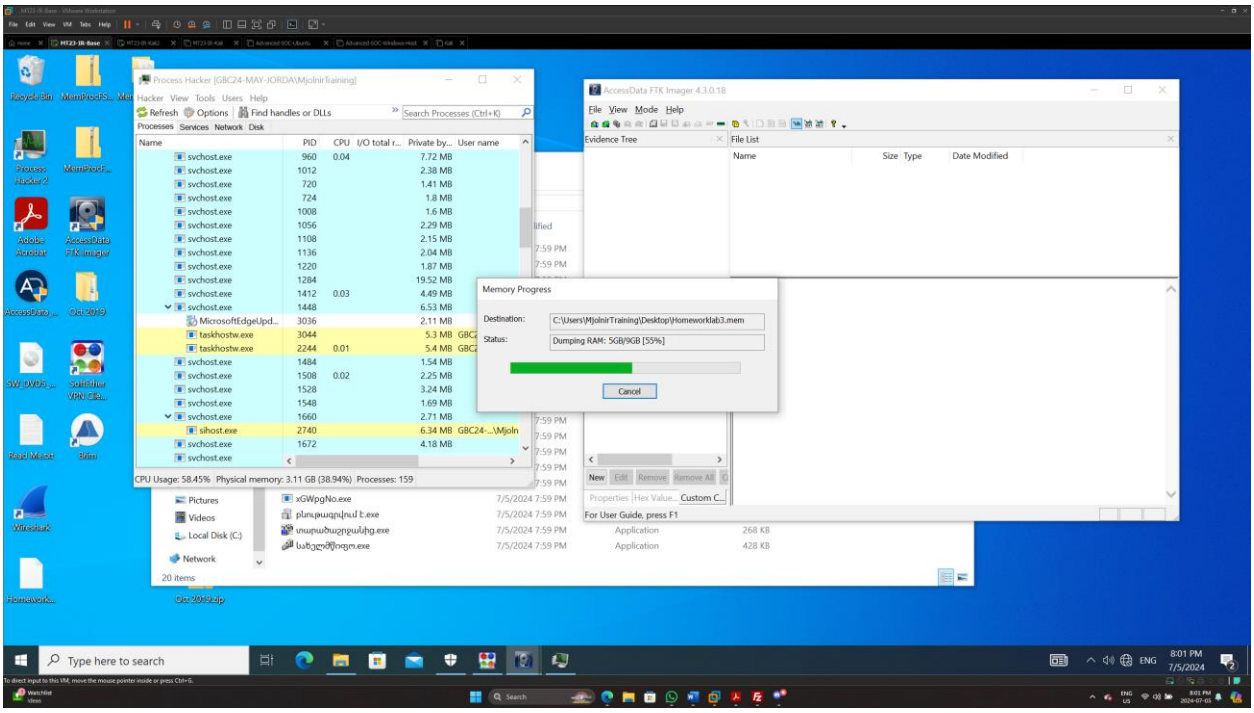


Figure 3 - FTK Imager capturing the systems memory while malware is running

After the memory was captured I moved it safely off of the VM, powered off the VM, and reverted my snapshot it its clean state.

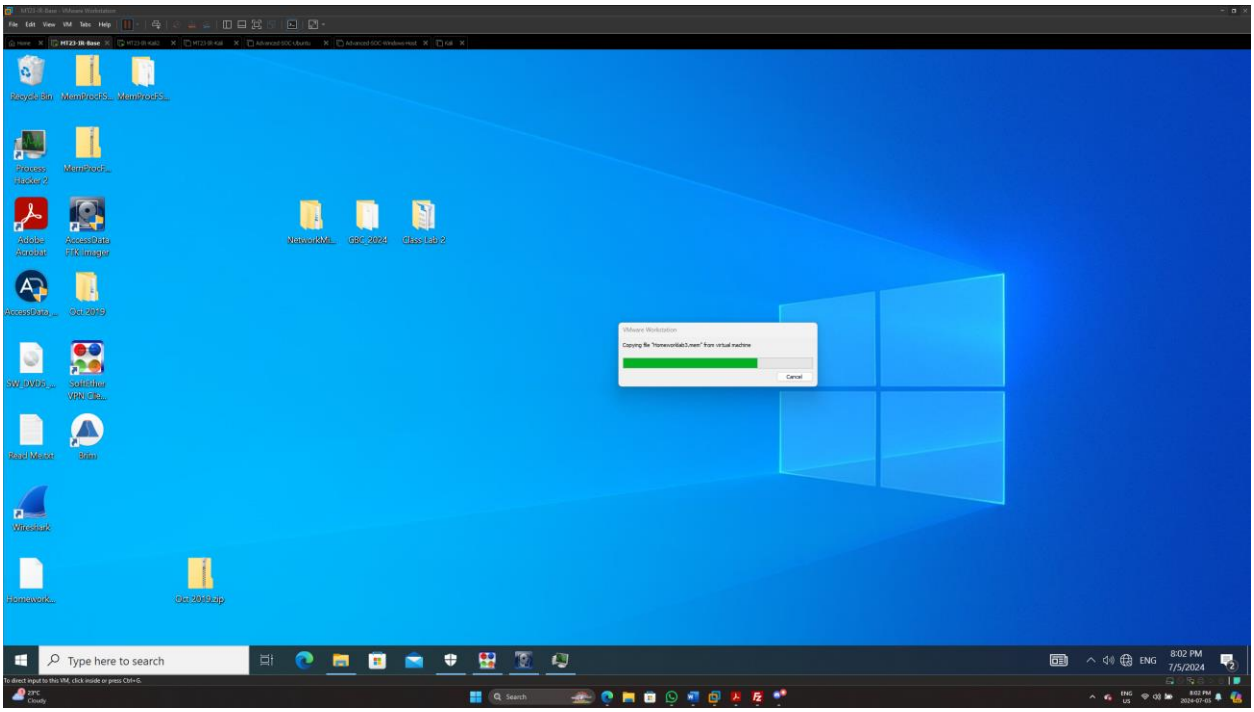


Figure 4 - Moving .mem file off of VM

It is now time to analyse the memory dump that was extracted in my previous step. For this I will transfer it to my Kali VM.

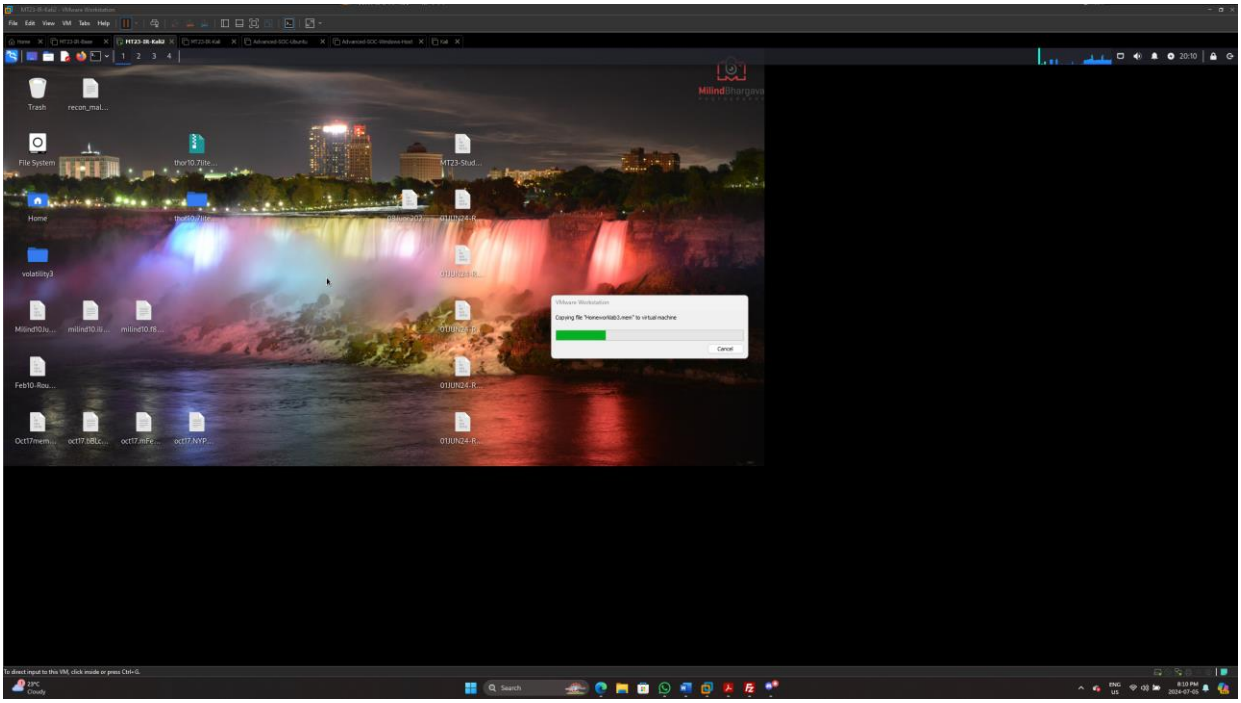


Figure 5 - Moving .mem file to Kali VM for analysis

Use the following command to run Volatility3's timeliner command on the memory dump and send its output to a separate file.

sudo python3 vol.py -f /home/mjolnirtraining/Desktop/Homeworklab3.mem timeliner.Timeliner > homeworklab3_timeliner.jordan

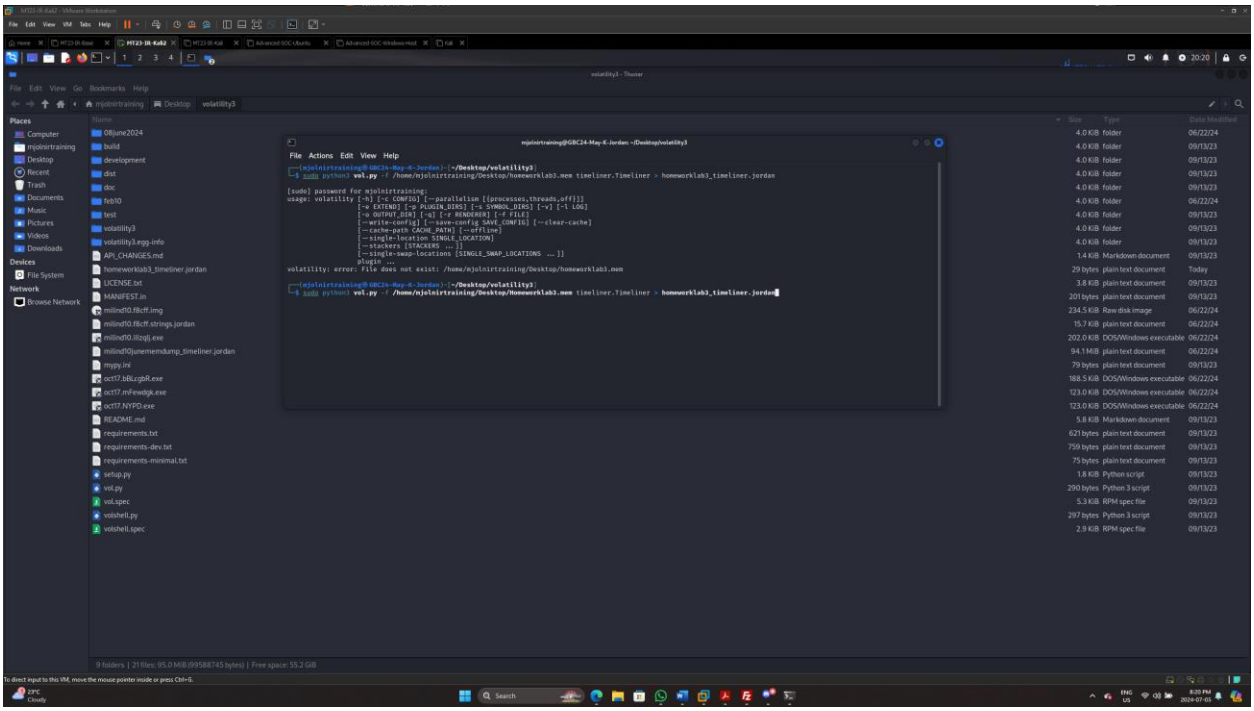


Figure 6 - Timeliner command outputting results to a separate file

I then used the command below to output a list of all of the processes running while the memory was captured. I analyzed the list of processes for anything malicious.

sudo python3 vol.py -f /home/mjolnirtraining/Desktop/Homeworklab3.mem windows.pslist.PsList > homeworklab3_pslist.jordan

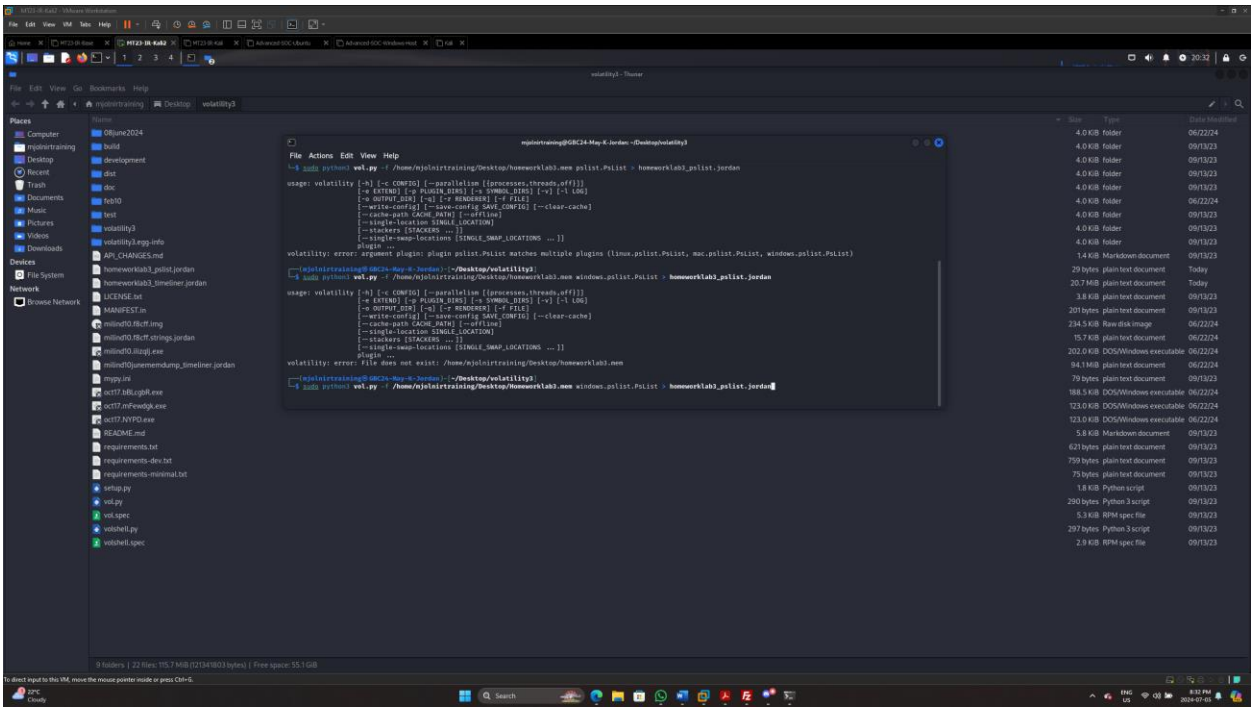
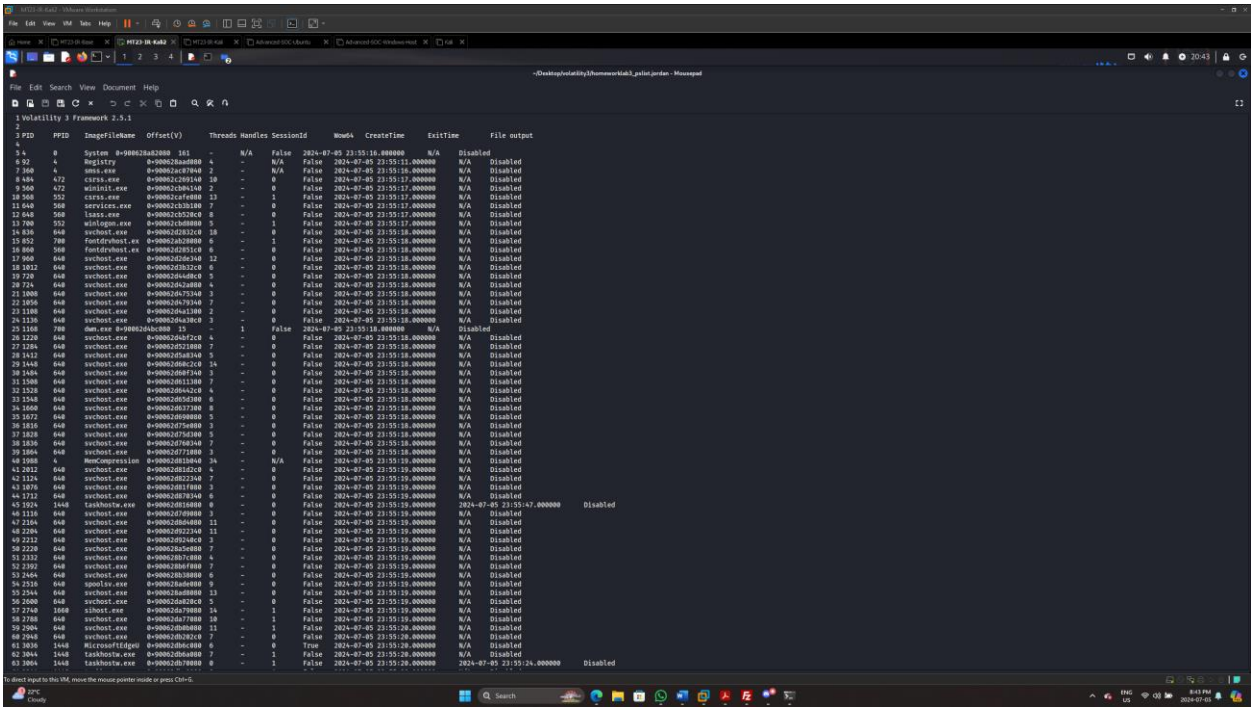


Figure 7 - PsList command outputting running processes at time of memory capture to a separate file

There were not too many processes running, so I was able to individually look up each process to see if they were malicious. Unfortunately, none of the processes seemed to be obviously or outwardly malicious. However, there was a huge glaring red flag in my PSList output. The svchost.exe process was running dozens of times, which seemed unusual.

After further research, I found that having several instances of svchost.exe running me be a sign of a malware infection. In our case, its safe to say that it is.

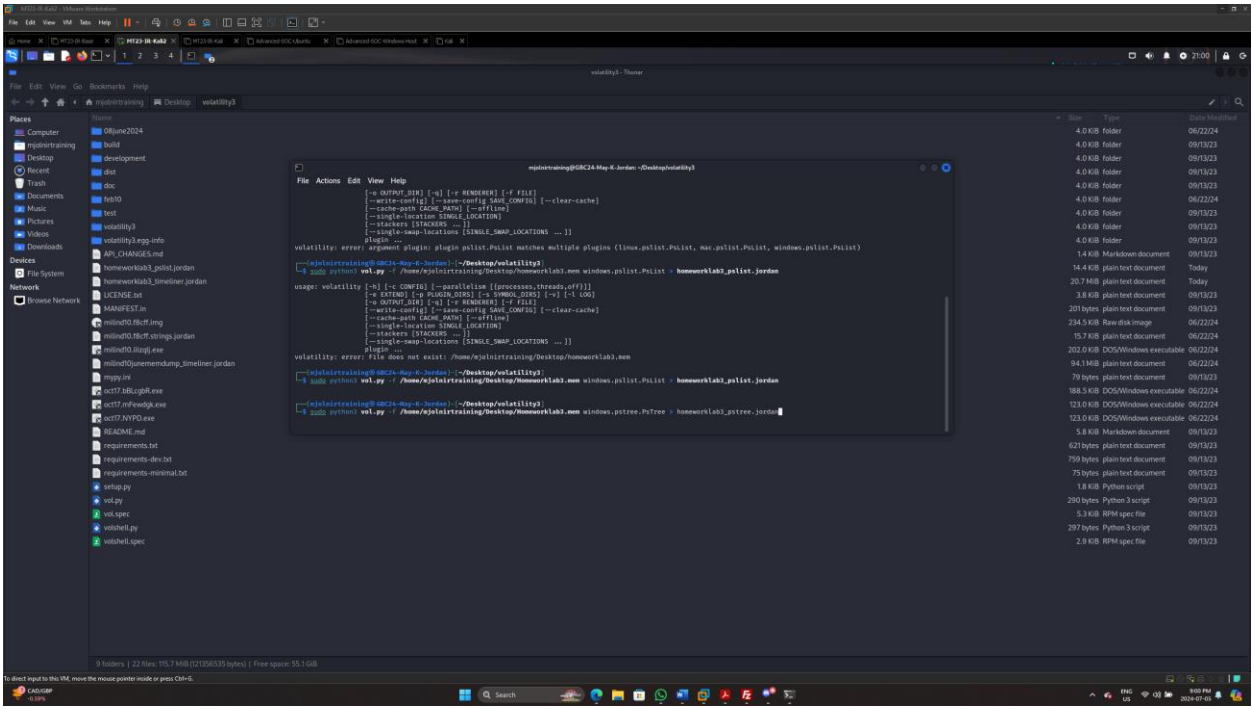


```
Volatility 3 Framework 2.5.1

PID    PPID   ImageFileName  Offset(V)      Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4      0      System         0x900628a82080 161     -       N/A     False   2024-07-05 23:55:16.000000      N/A     Disabled
92     4      Registry       0x900628aad080 4       -       N/A     False   2024-07-05 23:55:11.000000      N/A     Disabled
360    4      smss.exe       0x900622ac0704a 2      -       N/A     False   2024-07-05 23:55:16.000000      N/A     Disabled
444    472    csrss.exe      0x900622c09140 10      -       0       False   2024-07-05 23:55:17.000000      N/A     Disabled
560    472    wininit.exe    0x900622c0b140 2       -       0       False   2024-07-05 23:55:17.000000      N/A     Disabled
568    552    csrss.exe      0x900622cafe080 13     -       1       False   2024-07-05 23:55:17.000000      N/A     Disabled
640    560    services.exe   0x900622c33b180 7      -       0       False   2024-07-05 23:55:17.000000      N/A     Disabled
648    560    lsass.exe      0x900622cb520c0 8      -       0       False   2024-07-05 23:55:17.000000      N/A     Disabled
700    552    winlogon.exe   0x900622cbd0080 3      -       1       False   2024-07-05 23:55:17.000000      N/A     Disabled
836    640    svchost.exe    0x900622d2832c0 18     -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
852    700    fontdrvhost.ex 0x900622eb20080 6      -       1       False   2024-07-05 23:55:18.000000      N/A     Disabled
860    560    fontdrvhost.ex 0x900622d2831c0 6      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
960    640    svchost.exe    0x900622d2de340 12     -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1012   640    svchost.exe    0x900622d3b32c0 6      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
720    640    svchost.exe    0x900622d4e48c0 5      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
724    640    svchost.exe    0x900622d42a080 4      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1008   640    svchost.exe    0x900622d573140 3      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1056   640    svchost.exe    0x900622d479340 7      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1108   640    svchost.exe    0x900622d4a1380 2      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1136   640    svchost.exe    0x900622d4a30c0 3      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1168   780    dwm.exe 0x900622d4bc080 15     -       1       False   2024-07-05 23:55:18.000000      N/A     Disabled
1220   640    svchost.exe    0x900622d4bf2c0 4      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1284   640    svchost.exe    0x900622d523080 7      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1412   640    svchost.exe    0x900622d5a8340 5      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1446   640    svchost.exe    0x900622d6dc2c0 14     -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1484   640    svchost.exe    0x900622d6bf340 3      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1506   640    svchost.exe    0x900622d611380 7      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1526   640    svchost.exe    0x900622d6442c0 4      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1548   640    svchost.exe    0x900622d65d300 6      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1660   640    svchost.exe    0x900622d637380 8      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1672   640    svchost.exe    0x900622d698080 5      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1816   640    svchost.exe    0x900622d75e080 3      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1828   640    svchost.exe    0x900622d75d300 5      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1836   640    svchost.exe    0x900622d76d340 7      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1864   640    svchost.exe    0x900622d771080 3      -       0       False   2024-07-05 23:55:18.000000      N/A     Disabled
1908   4      MemCompression 0x900622d830040 34     -       N/A     False   2024-07-05 23:55:19.000000      N/A     Disabled
2012   640    svchost.exe    0x900622d81d2c0 4      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
1124   640    svchost.exe    0x900622d822340 7      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
1876   640    svchost.exe    0x900622d817080 3      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
1712   640    svchost.exe    0x900622d878340 6      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
1924   1448   taskhostw.exe  0x900622d816080 0      -       0       False   2024-07-05 23:55:19.000000      2024-07-05 23:55:47.000000      Disabled
1116   640    svchost.exe    0x900622d76f080 3      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2104   640    svchost.exe    0x900622d8d6080 11     -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2204   640    svchost.exe    0x900622d922340 11     -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2212   640    svchost.exe    0x900622d924dc0 3      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2220   640    svchost.exe    0x900628a5e080 7      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2332   640    svchost.exe    0x900628b7c080 4      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2392   640    svchost.exe    0x900628b6f080 7      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2464   640    svchost.exe    0x900628b58080 6      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2536   640    spoolsv.exe    0x900628a6e080 9      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2544   640    svchost.exe    0x900628a88080 13     -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2600   640    svchost.exe    0x900628a828c0 5      -       0       False   2024-07-05 23:55:19.000000      N/A     Disabled
2740   1660   sihost.exe     0x900628a70080 14     -       1       False   2024-07-05 23:55:19.000000      N/A     Disabled
2788   640    svchost.exe    0x900628a77080 10     -       1       False   2024-07-05 23:55:19.000000      N/A     Disabled
2904   640    svchost.exe    0x900628d80080 11     -       1       False   2024-07-05 23:55:20.000000      N/A     Disabled
2948   640    svchost.exe    0x900628d282c0 7      -       0       False   2024-07-05 23:55:20.000000      N/A     Disabled
3036   1448   MicrosoftEdgeU 0x900628d8bc080 6     -       0       True    2024-07-05 23:55:20.000000      N/A     Disabled
3044   1448   taskhostw.exe  0x900628d6a080 7      -       1       False   2024-07-05 23:55:20.000000      N/A     Disabled
3064   1448   taskhostw.exe  0x900628d70080 0      -       1       False   2024-07-05 23:55:20.000000      2024-07-05 23:55:24.000000      Disabled
```

Figure 8 - List of processes that were running at the time of memory capture. svchost.exe appears dozens of times

I then ran the PsTree command using the command noted below. I did this in the hopes that it may give me a better or clearer picture of what the parents or child processes of these svchost.exe's may be. This could potentially help us identify our malware.

sudo python3 vol.py -f /home/mjolnirtraining/Desktop/Homeworklab3.mem windows.pstree.PsTree > homeworklab3_pstree.jordan



While the PsTree command did give more results/information I was still unable to identify which processes were actually the malware.

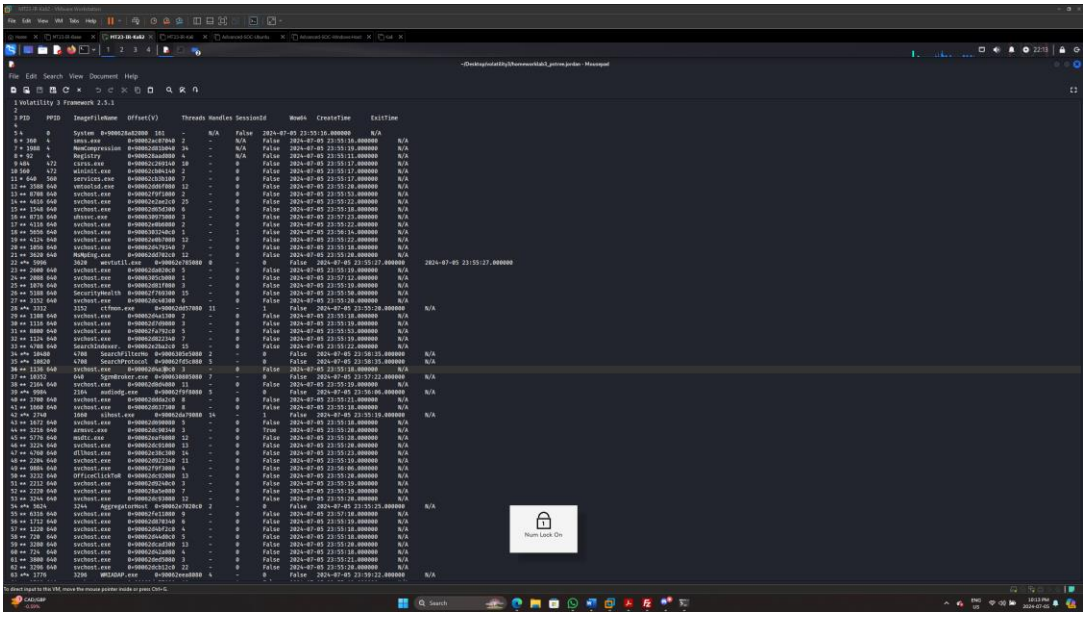
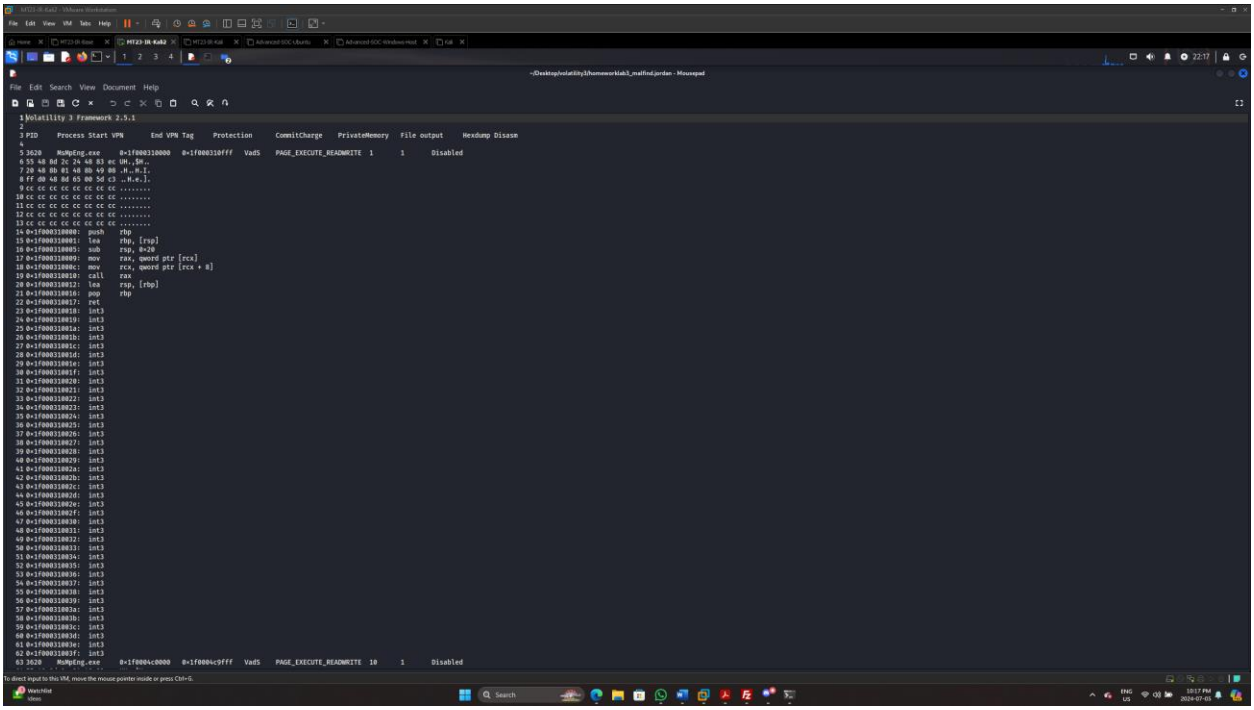


Figure 9 - My PsTree output

I then attempted to find the malware by running the malfind command. Malfind is used to find hidden/injected code. I figure that maybe, because the malwares were hiding themselves as different processes, I may be able to identify them this way. Unfortunately, the output was too long, and I was too untrained to find anything. I reviewed the output, but nothing popped out to me as potential malicious code.

The command used to generate this was:

sudo python3 vol.py -f /home/mjolnirtraining/Desktop/Homeworklab3.mem windows.malfind > homeworklab3_malfind.jordan



Ath this point of the lab I was stuck because I could not identify which PID's belonged to the malware which means that I could then not go and extract its strings to create Yara Rules.

I decided to manually retrieve the strings and required information directly from the malware using process hacker so that the lab could continue. I went back to my Windows VM and re-ran the malware one at a time. In process hacker, if you right click on the process, select properties, and then memory, you can view and save all of the strings for that exe/process.

I saved the strings for all of the malwares for use in the creation of Yara rules.

For the sake of the exercise I will be referring to the malwares as Malware 1, 2 , and 3. The name of the malware corresponding to each new name can be seen above the photos below.

სახელმწიფო.exe – Malware1

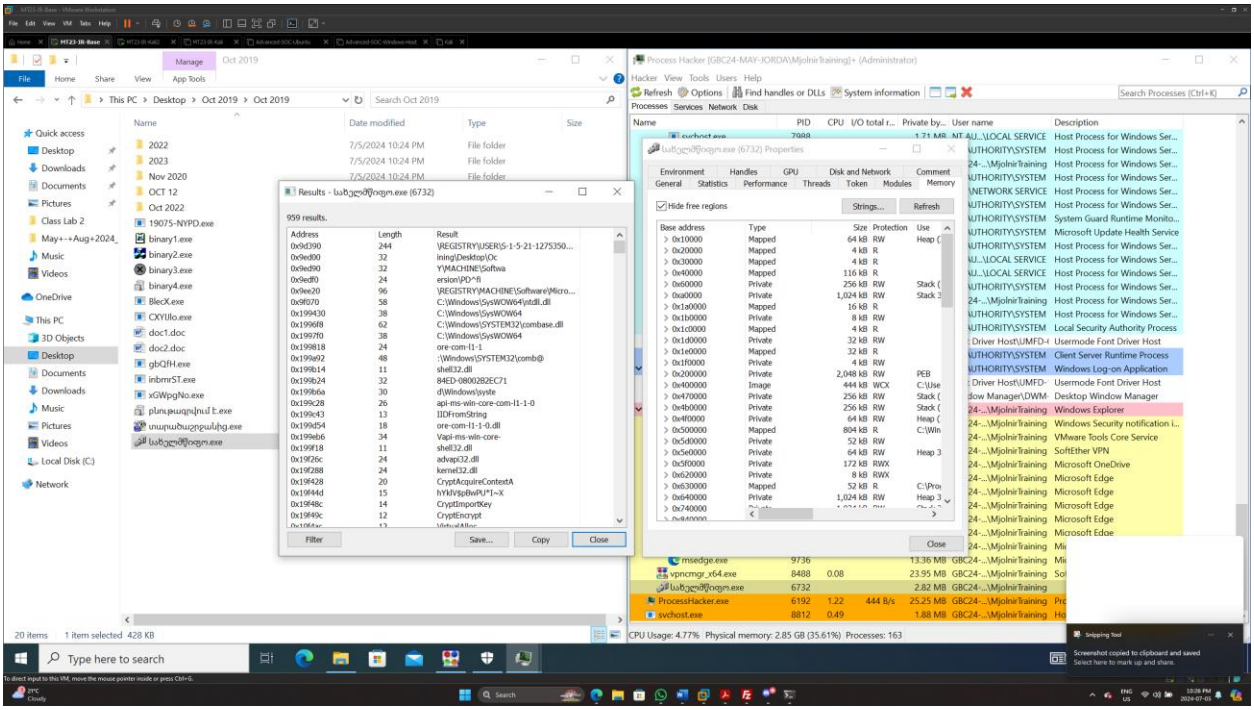


Figure 10 - Capturing the strings for Malware 1

տարածաշրջանից.exe – Malware 2

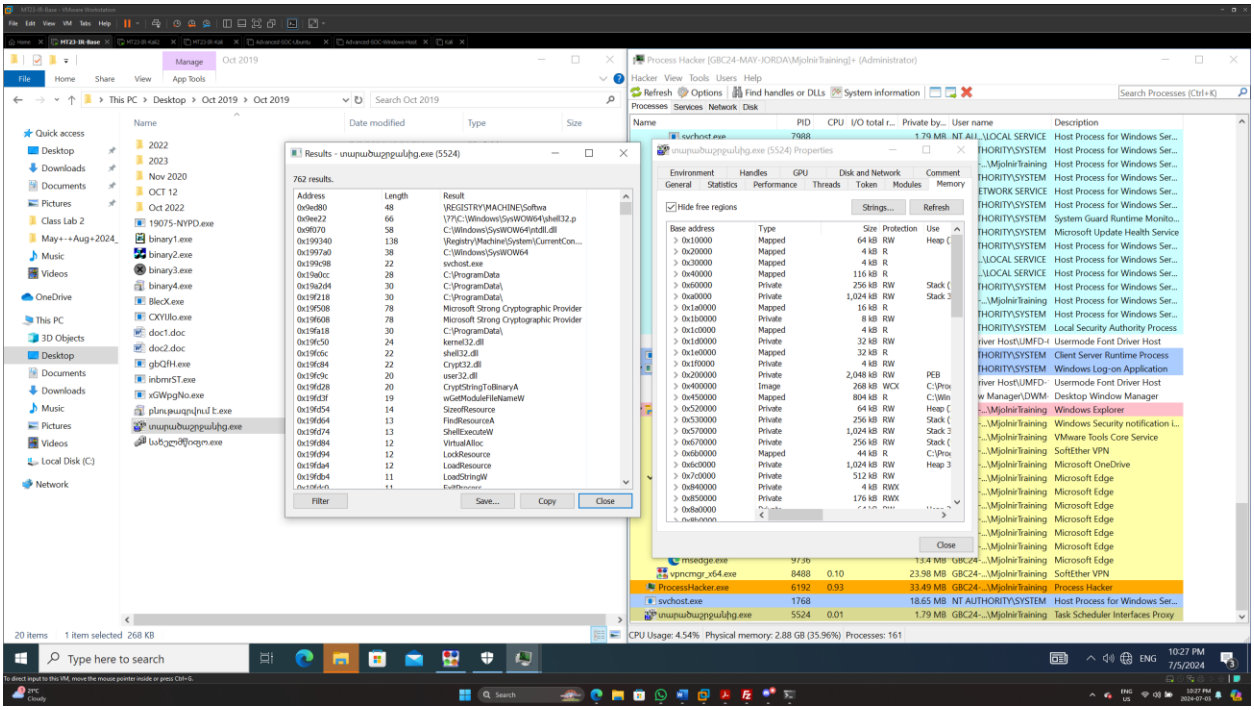


Figure 11 - Capturing the strings for Malware 2

բնութագրվում է.exe – Malware 3

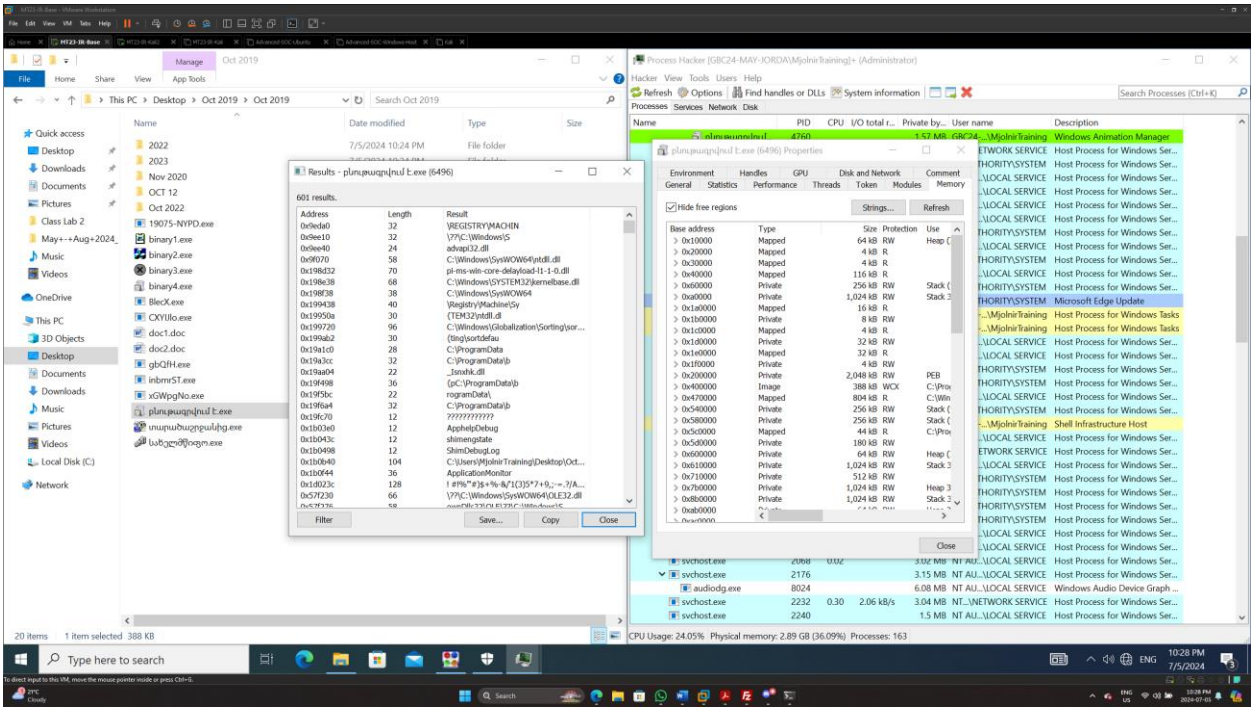


Figure 12 - Capturing the strings for Malware 3

Now that I have all of the strings for my 3 malwares, I brought them over to my Kali VM so that I could work on creating Yara rules.

I created a Yara rule for each malware, they are the following:

Malware 1:

```
rule Malware1
	{
	meta:
		Description = "Detects the execution of malware 1"
		Name = "Jordan Patterson"
	strings:
		$s1 = "w2rsBoBCSep" ascii wide
		$s2 = "j8rjhir00Yp+S2Hg" ascii wide
		$s3 = "4E5N5T5w5}5" ascii wide
	condition:
	3 of them
	}
```

Malware 2:

```
rule Malware2
	{
	meta:
		Description = "Detects the execution of malware 2"
		Name = "Jordan Patterson"
	strings:
		$s1 = "RLjlVwZiGg5l" ascii wide
		$s2 = "PLQLPLUFfOl8GyklD7P" ascii wide
		$s3 = "cLhxeLhxeBTYeXdpRUGpGgWL4wJl" ascii wide
	condition:
```

```
        3 of them
        }
```

Malware 3:

```
rule Malware3
        {
        meta:
                Description = "Detects the execution of malware 3"
                Name = "Jordan Patterson"
        strings:
                $s1 = "j64J3KdhevVJ" ascii wide
                $s2 = "QCqKQ6pr7iJIej" ascii wide
                $s3 = "Q6mA0tmXPJNhevVJ" ascii wide
        condition:
        3 of them
        }
```

Each rule initially contained the non-English names of the malware but the Yara rules failed to execute because of them. For this reason, I removed the non-English names from the rules completely.

I chose the strings to identify each malware based on their uniqueness. I felt that they were strings that were unlikely to be found in any other executable.

Once my rules were finalized, I placed them in the custom signatures folder of Thor-Lite.

Figure 13 - Custom Signatures folder containing my Yara Rules

Because this linux system does not contain the malware or any memory dumps that contain traces of these malwares or strings, I placed copies of the strings files I got from Process hacker into the volatility3 folder and told Thor-Lite to scan that folder. This should simulate a file existing that contains those strings.

I scanned the Linux system with Thor-Lite using the following command:

sudo ./thor-lite-linux-64 --path /home/mjolnirtraining/Desktop/volatility3 --quick –customonly

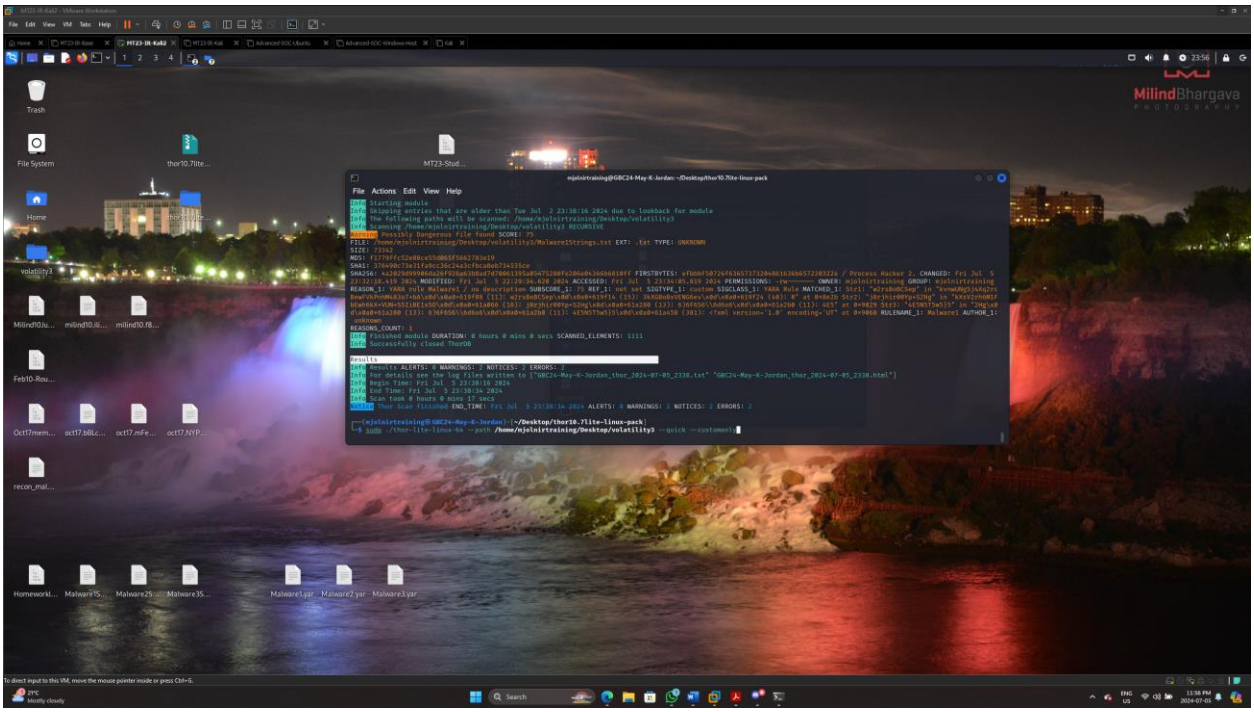


Figure 14 - About to execute the command to scan using my Yara rules

The scan yielded several warnings and zero errors. All files containing the strings were successfully identified by Thor-Lite based on my Yara rules. The official HTML report will be uploaded with my Word report as instructed.

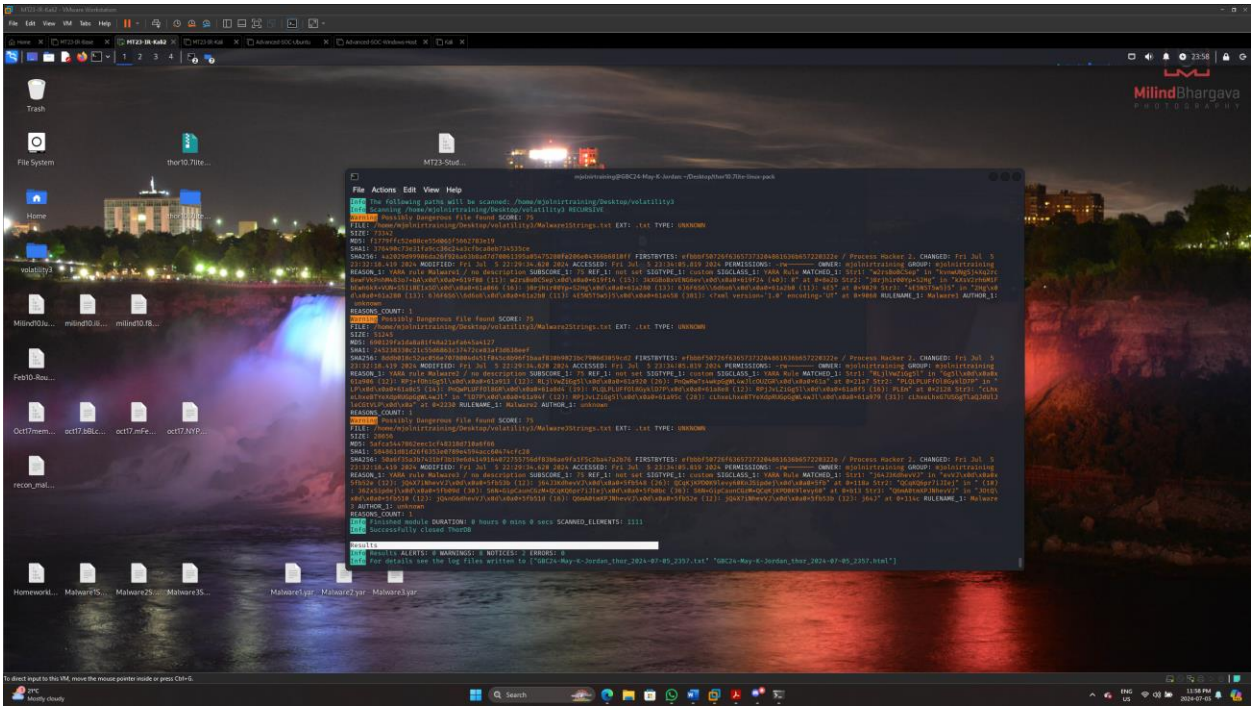


Figure 15 - Thor-Lite scan successfully identified the files containing the strings belonging to the non-English malware's

The next task will be to test my Yara rules on my Windows VM using Thor-Lite. I set up Thor-Lite on my Windows VM and deposited my previously created Yara rules into the custom signatures folder. I also created a directory containing files that contain the strings specified in my Yara rules. I will point Thor-Lite to this directory to simulate detection as malware will not be re-run at this step.

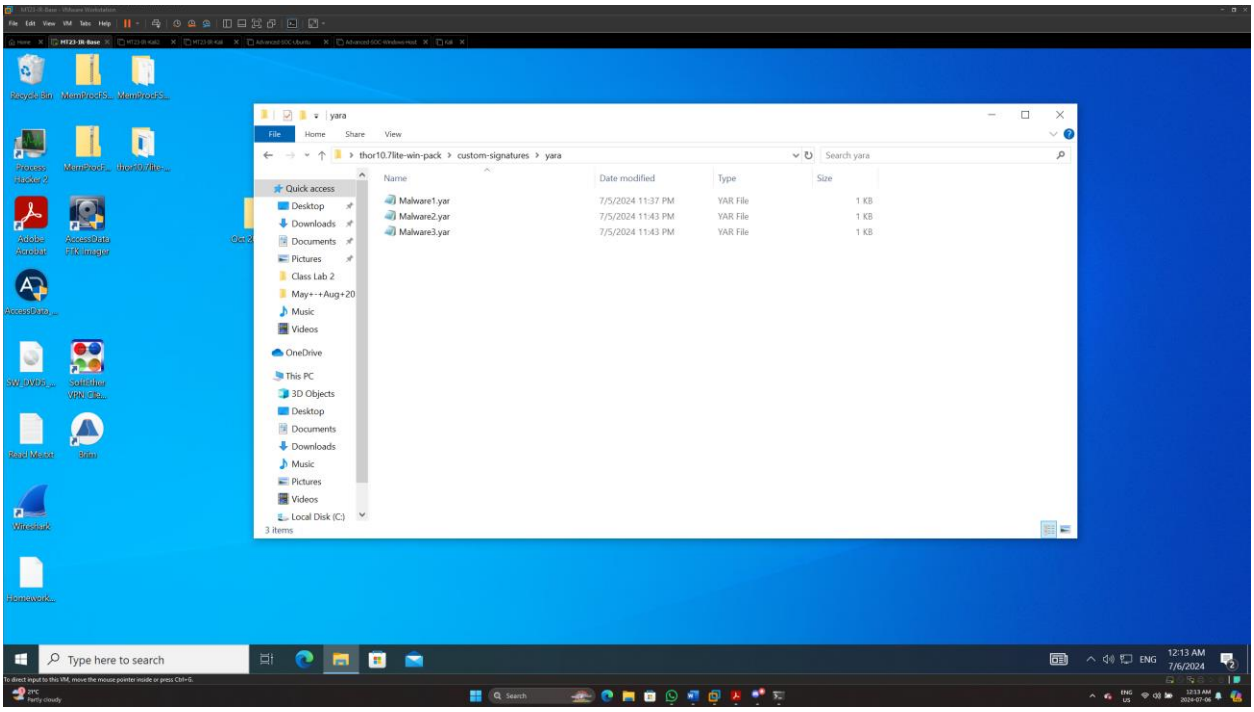


Figure 16 - The Yara rules have been placed in the custom signatures folder on the Windows VM

I direct Thor-lite to only scan the specified directory using my custom Yara rules using the command below.

.\thor64-lite.exe --path C:\Users\MjolnirTraining\Desktop\MalwareStrings --quick --customonly

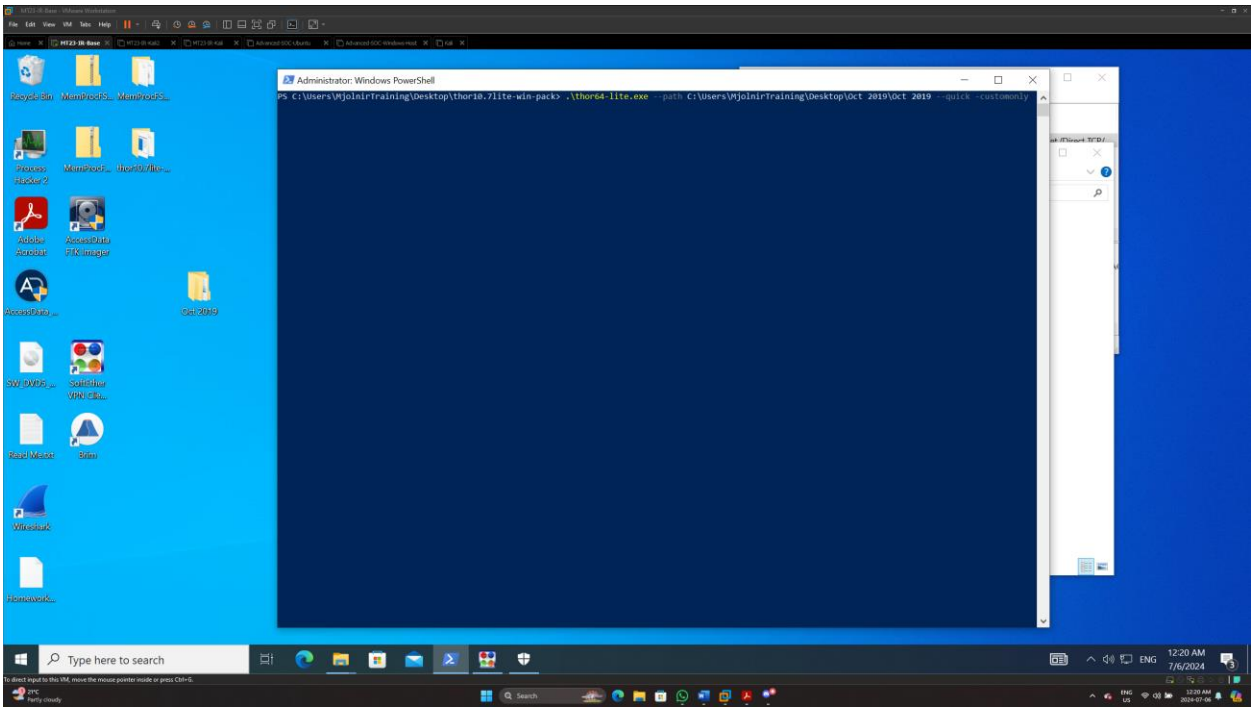


Figure 17 - Thor-Lite about to scan using my Yara rules

Several warnings are raised by this scan. The full HTML report will be uploaded along with this Word report.

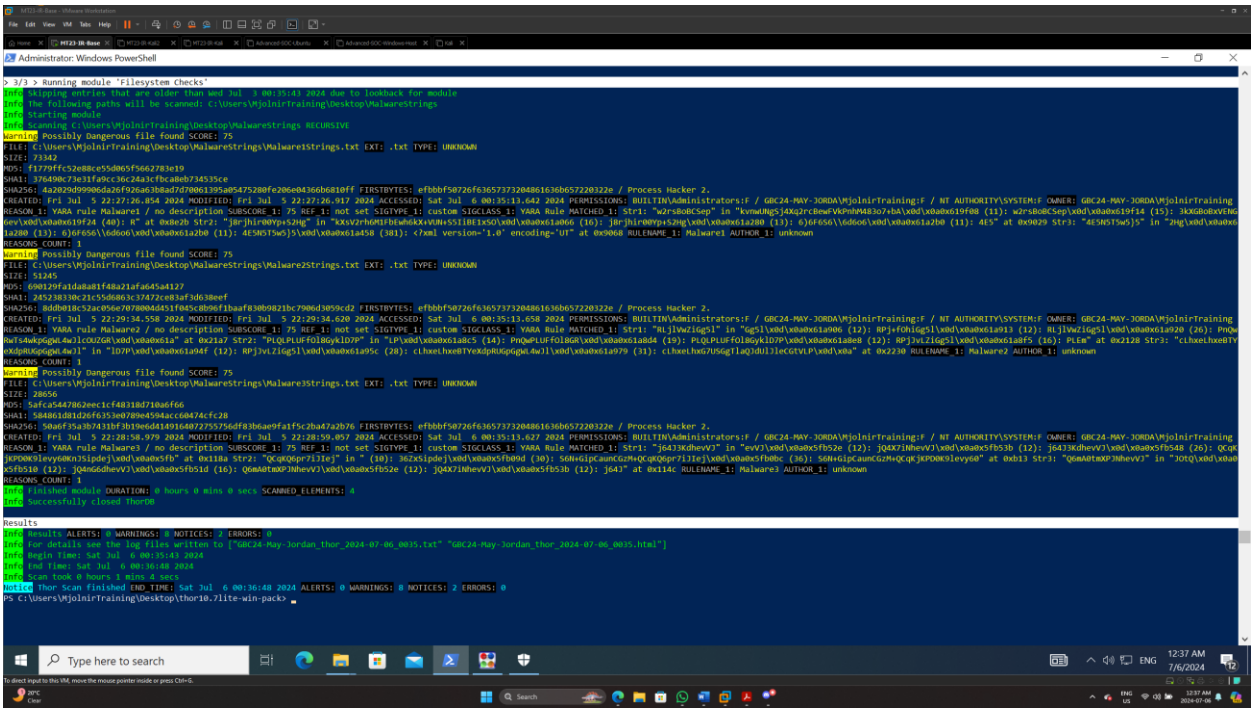


Figure 18 - Yara rules successfully identify the malware strings