EXTRACT MALWARE STRINGS FROM A MEMORY DUMP AND CREATE YARA RULES

Jordan Patterson

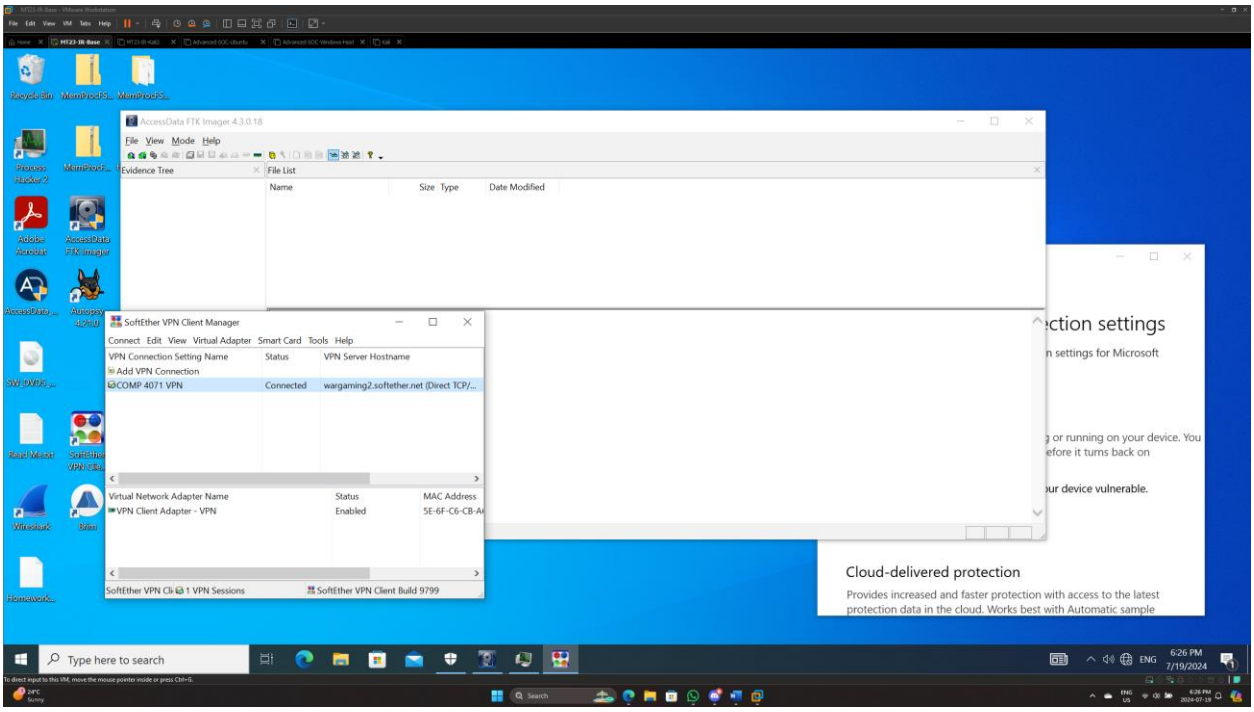
2024

Goals of this Lab:

- Generate a memdump file after successfully executing all ransomware in the Oct 2019 folder on your vm
- do not revert the vm, just pause it
- Copy memdump to Kali VM
- Run volatility
- Dump malicious processes (screenshot for each)
- String analysis to create yara rule for each process
- Put the rule in thor lite custom signature folder
- download the scanner for thor lite to windows, unpause the vm and run the thor lite scanner for custom signature
- collect report
- The word report for this activity must contain:
    - What evidence you started with
    - What analysis was done
    - What were the findings
    - Did your yara rules work? If no, why?
    - Final analysis in 2000 words. This must include a flow of thoughts, have an opening, a body, screenshots with explanations and conclusion.

- ## Submission items:
- Word report
- Thorlite Output HTML report

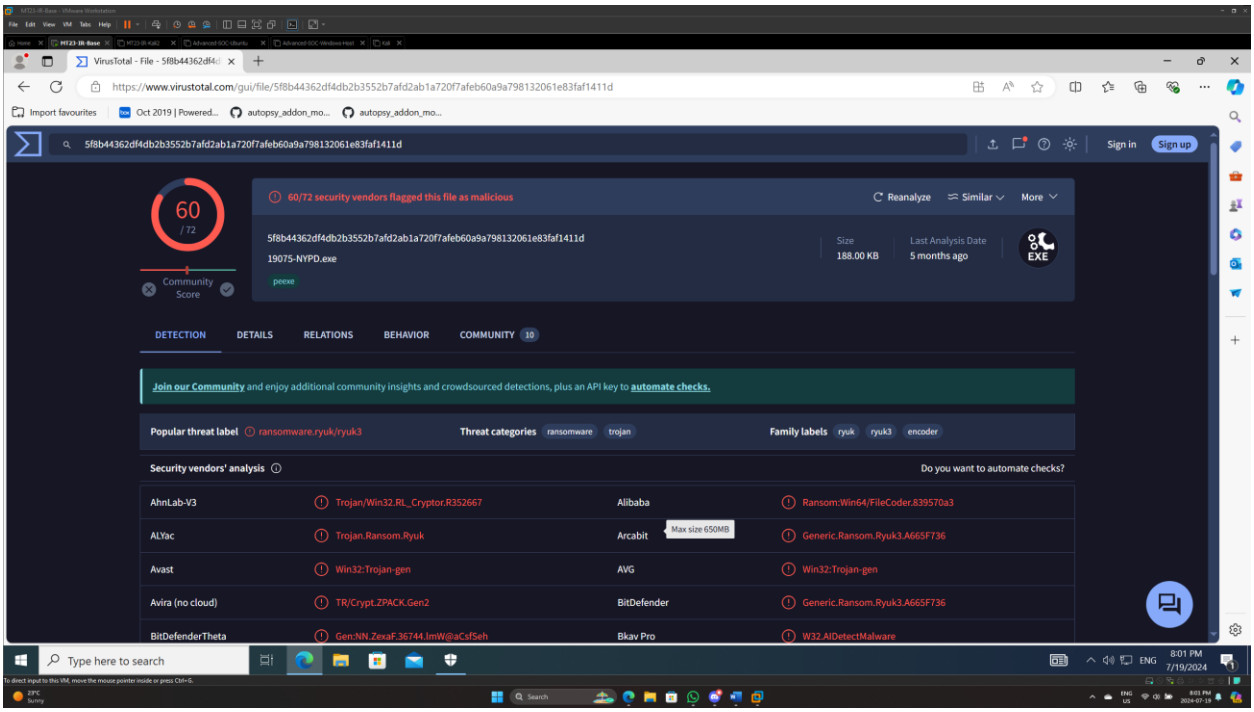
I began this lab with my Windows VM. I took a snapshot before beginning. I downloaded the required malware, connected to the VPN, ensured that real-time protection was off, and ensured that FTK Imager was open and ready to capture the memory of the system after the ransomware had been run.



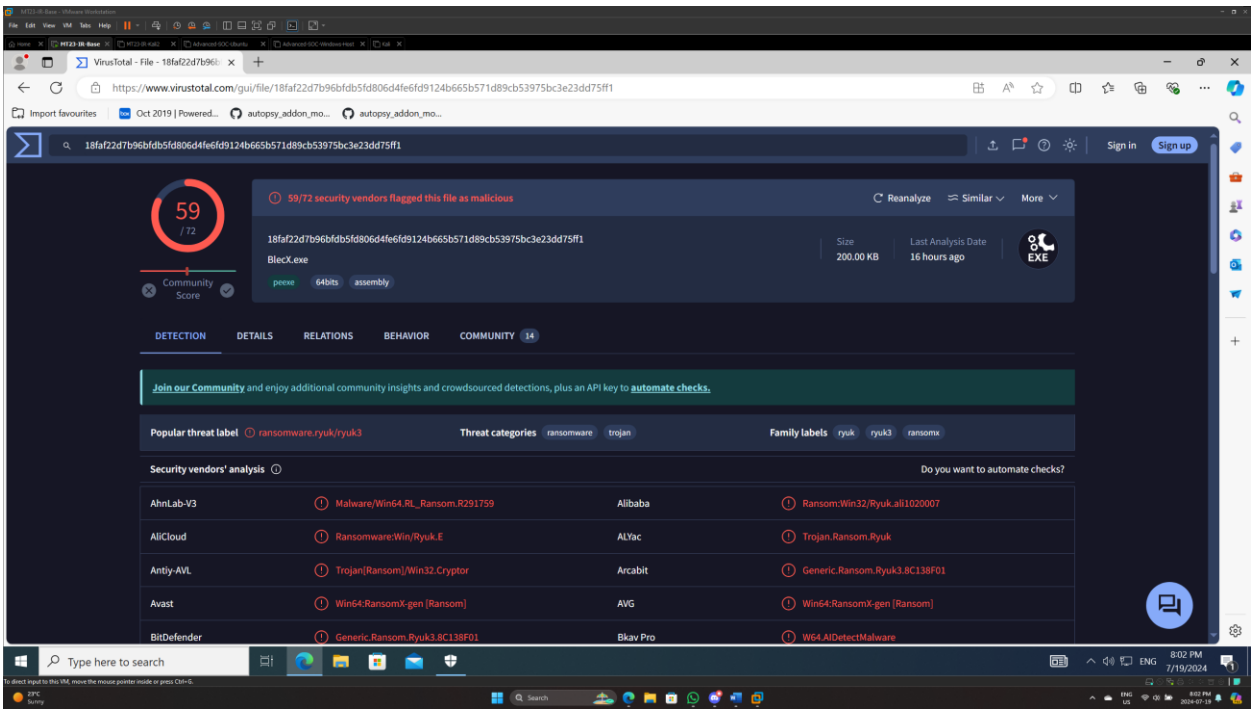
I then gathered all of the malware in the Oct 2019 folder and compared them with VirusTotal to determine which of them were ransomware. I identified 4 of the executables as ransomware.

The identified Ransomware were:

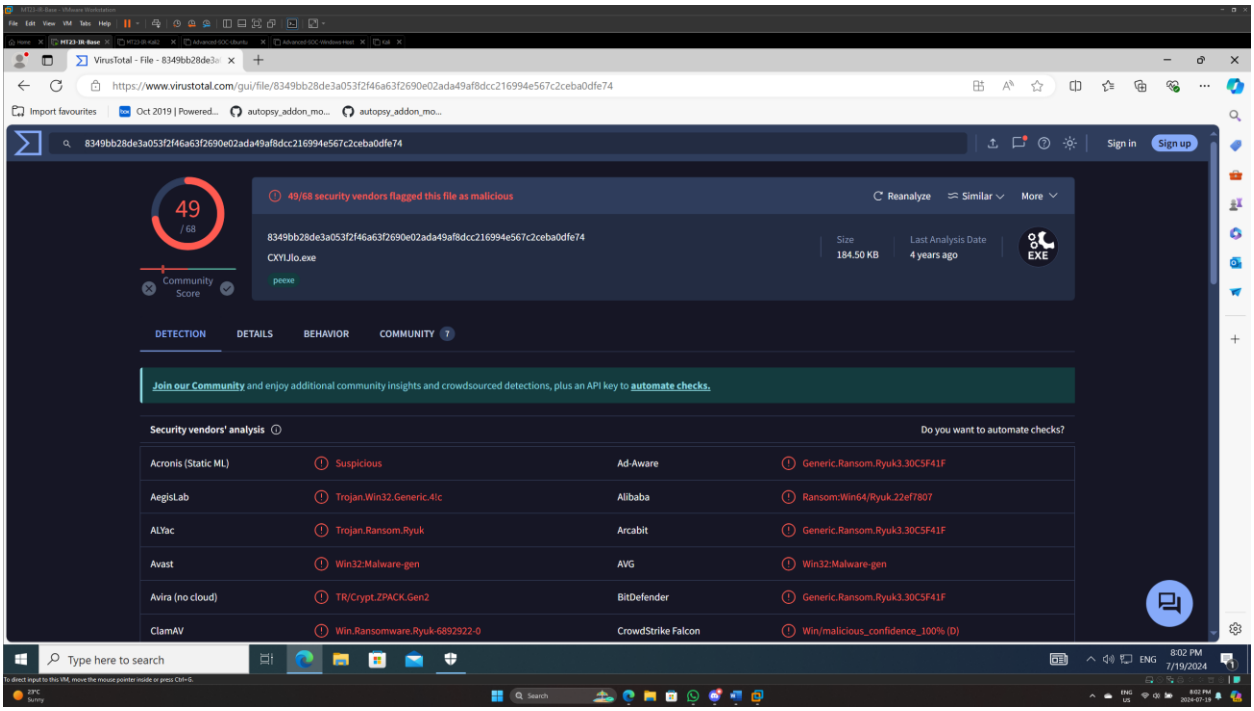
19075-NYPD.exe



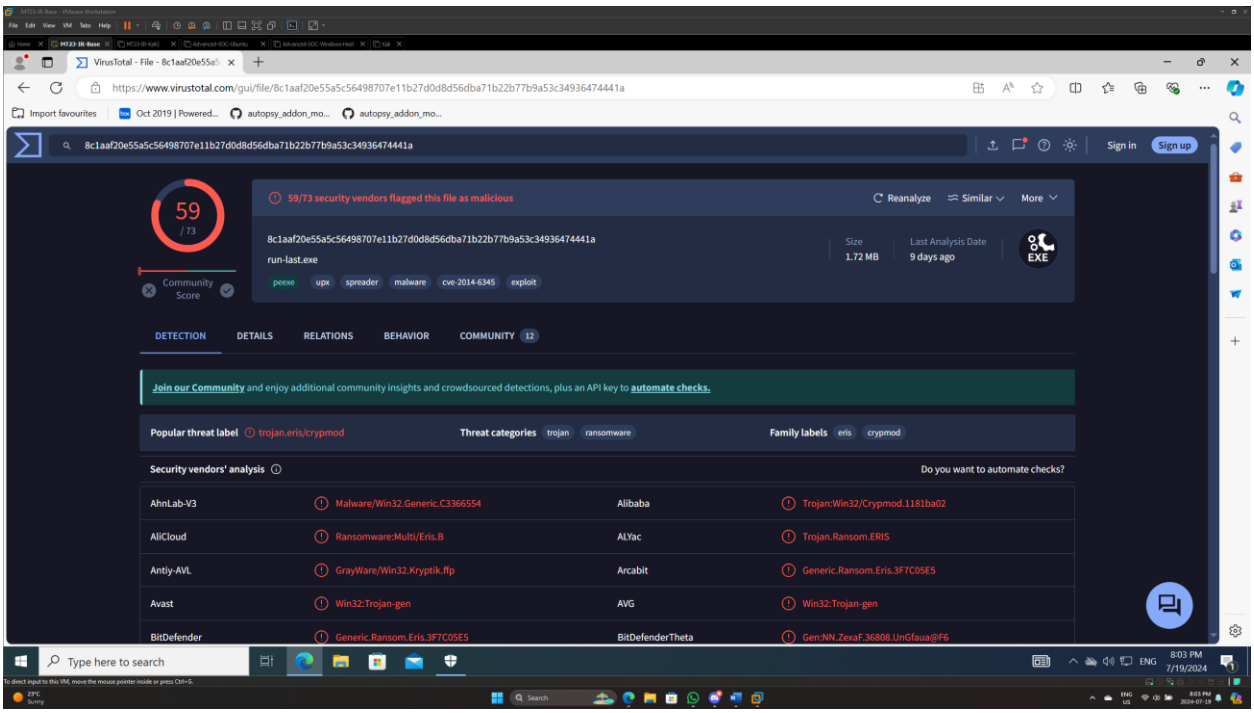
BlecX.exe



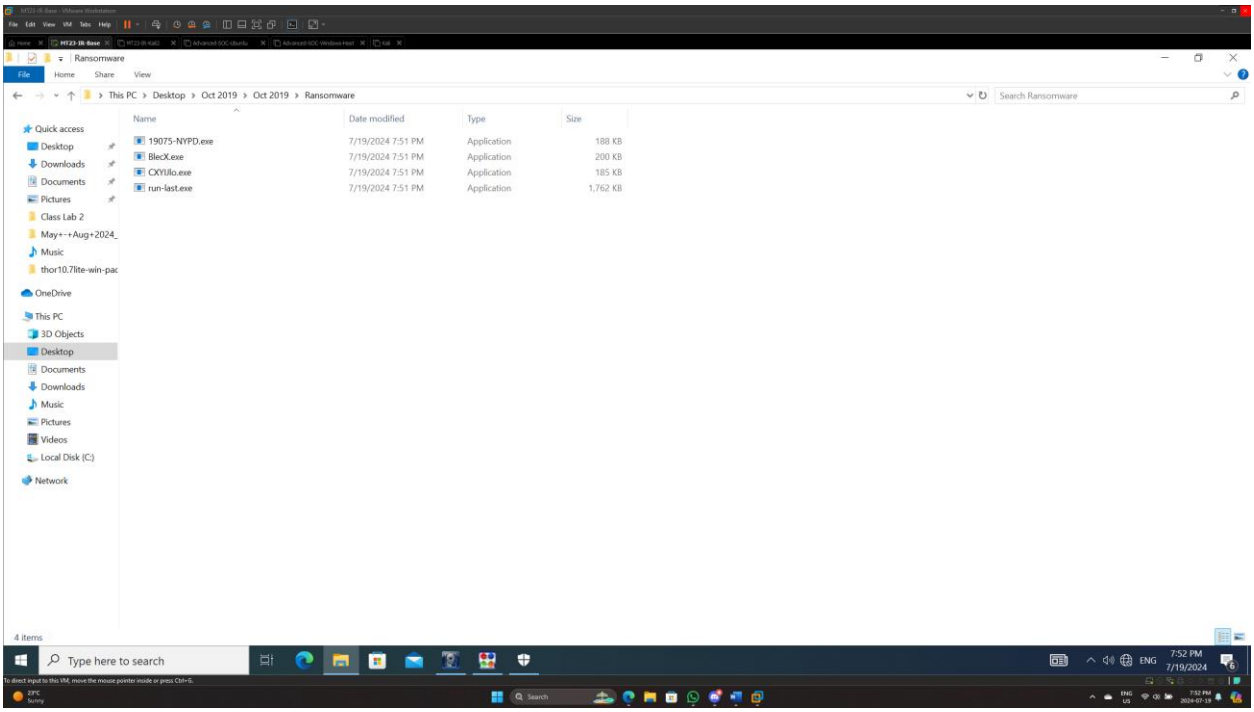
CXYIJlo.exe



Run-last.exe

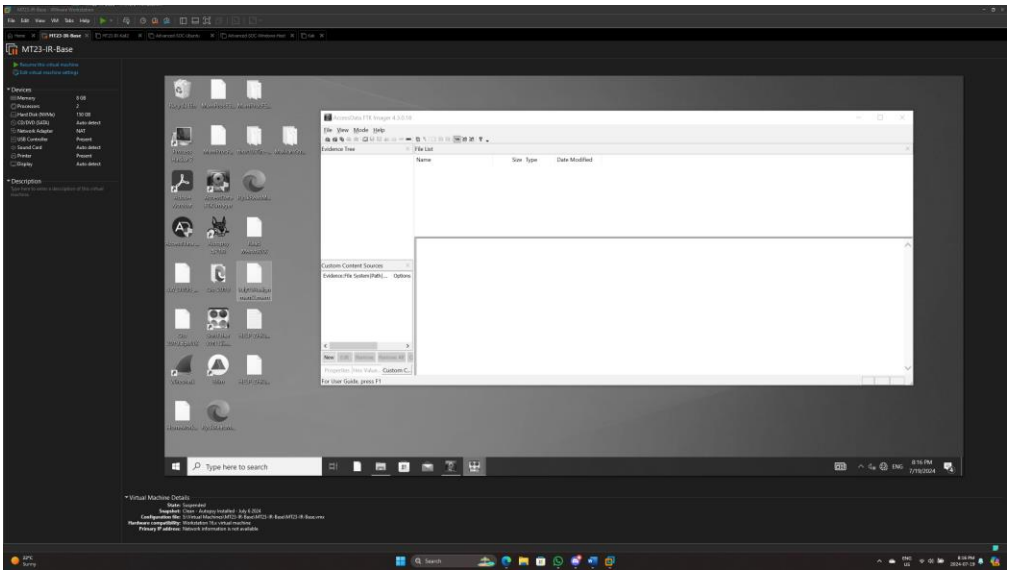


I then ran all identified ransomware from the Oct 2019 folder as administrator and allowed them time to run properly.

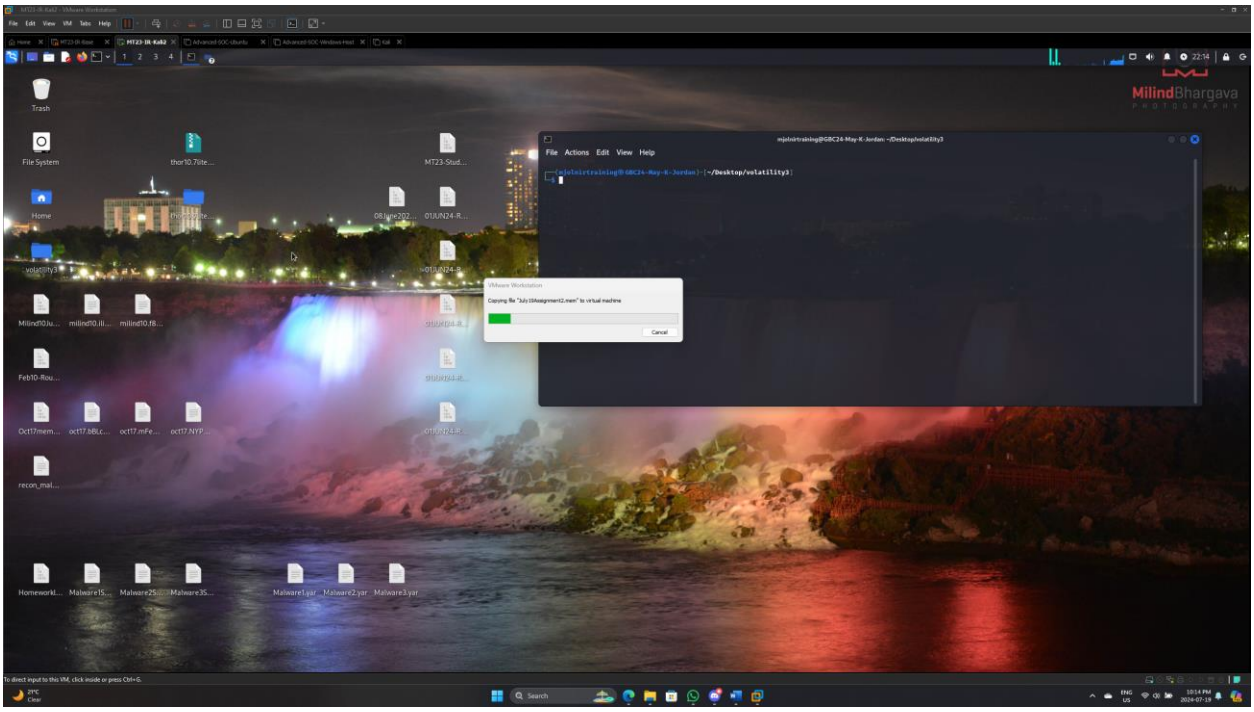


After all Ransomware had been run, I captured the systems memory using FTK Imager and successfully copied it out of the VM before it became encrypted.

I then paused the Windows VM in its current state.

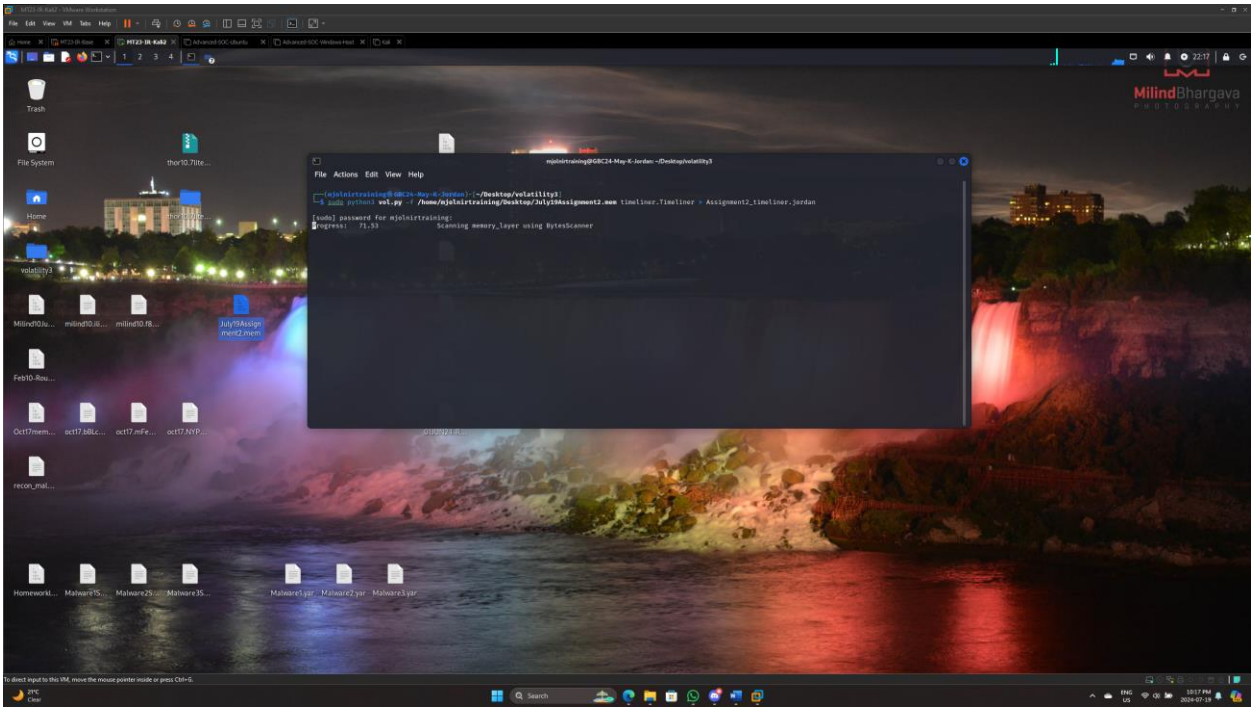


I turned on the Kali VM, transferred the memory dump, and opened Volatility.



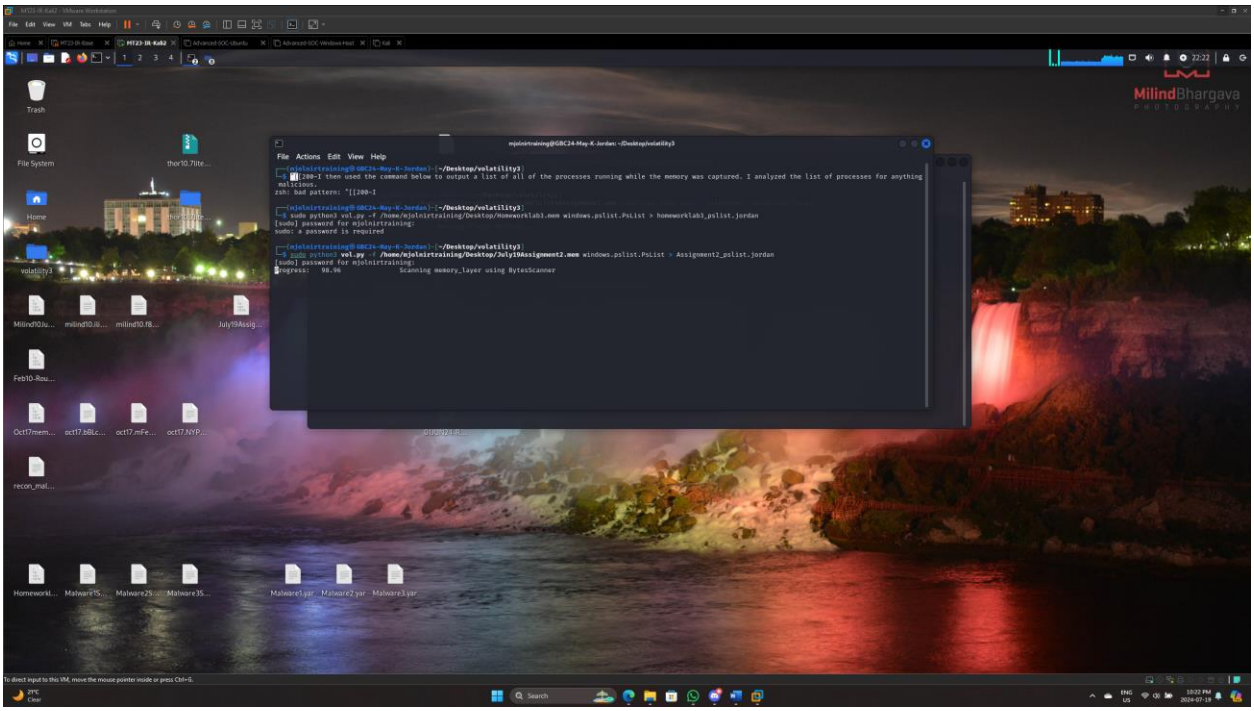
I used the following command to run Timeliner and output the results to a separate file.

sudo python3 vol.py -f /home/mjolnirtraining/Desktop/July19Assignment2.mem timeliner.Timeliner > Assignment2_timeliner.jordan

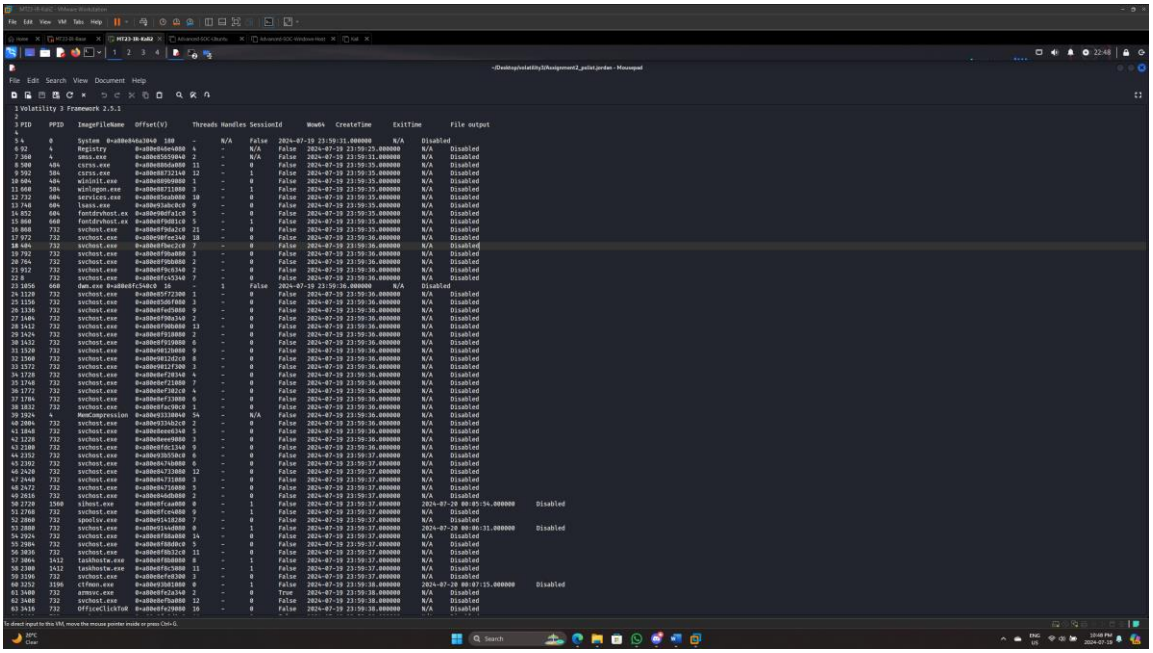


I then used the command below to output a list of all the processes running while the memory was captured. I analyzed the list of processes for anything malicious.

sudo python3 vol.py -f /home/mjolnirtraining/Desktop/July19assignment2.mem windows.pslist.PsList > Assignment2_pslist.jordan



Once my pslist file was generated, I opened it and my timeliner file and began determining which processes are malicious.

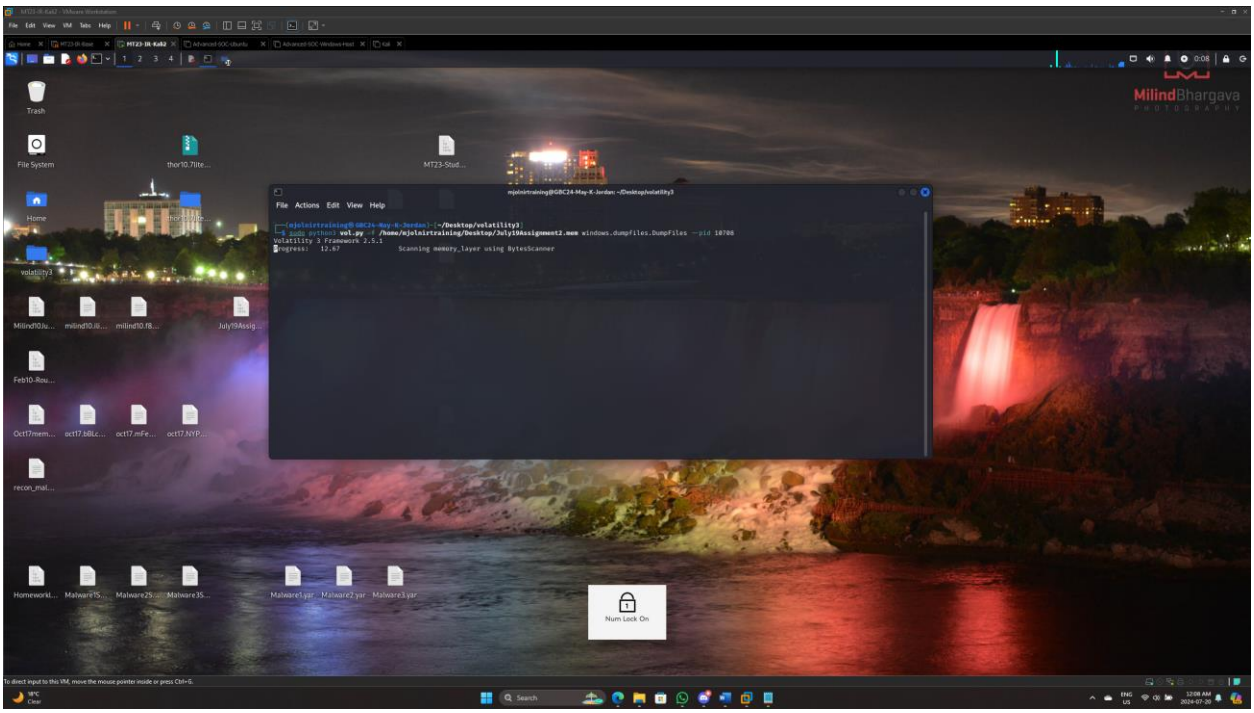


I identified the following process ID's:

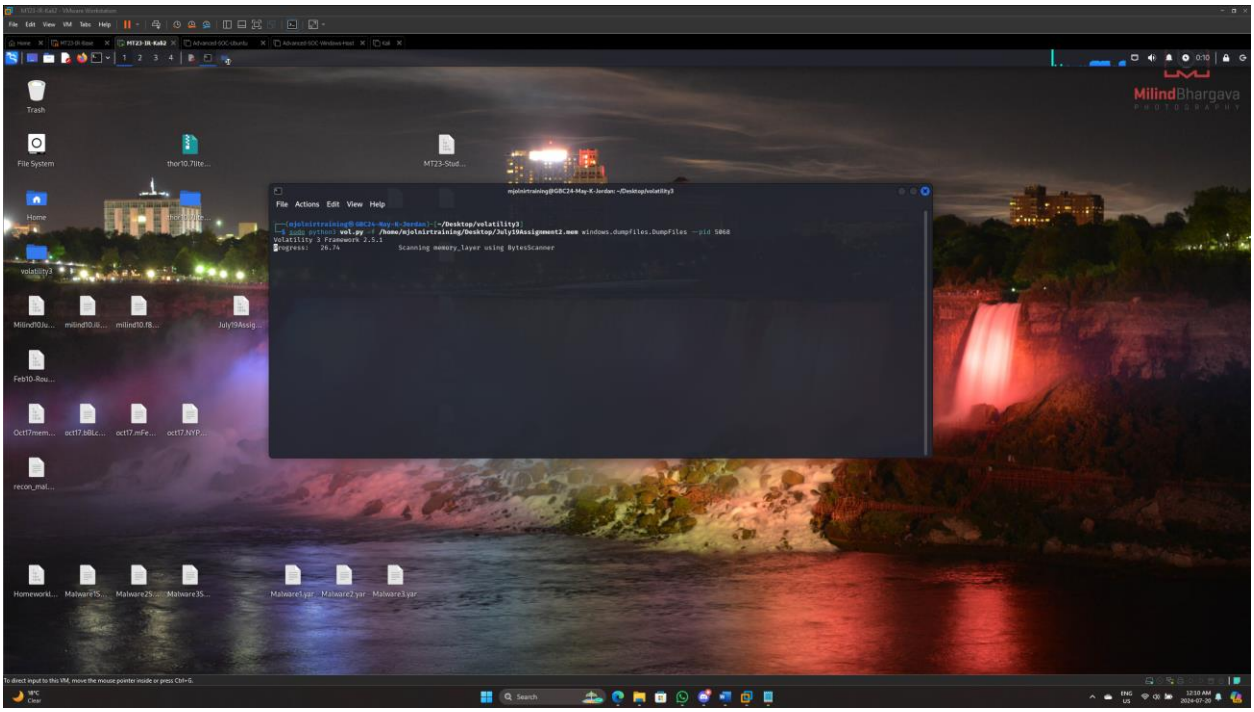
- **PID: 10708**
  - **Process: 19075-NYPD.exe**
- **PID: 5068**
  - **Process: BlecX.exe**
- **PID: 536208**
  - **Process: CXYIJlo.exe**
- **PID: 600084**
  - **Process: run-last.exe**

I used the following command to dump the executable for the 19075-NYPD.exe ransomware. I then repeated these steps, changing the PID, to dump the executables for the other ransomware as well.

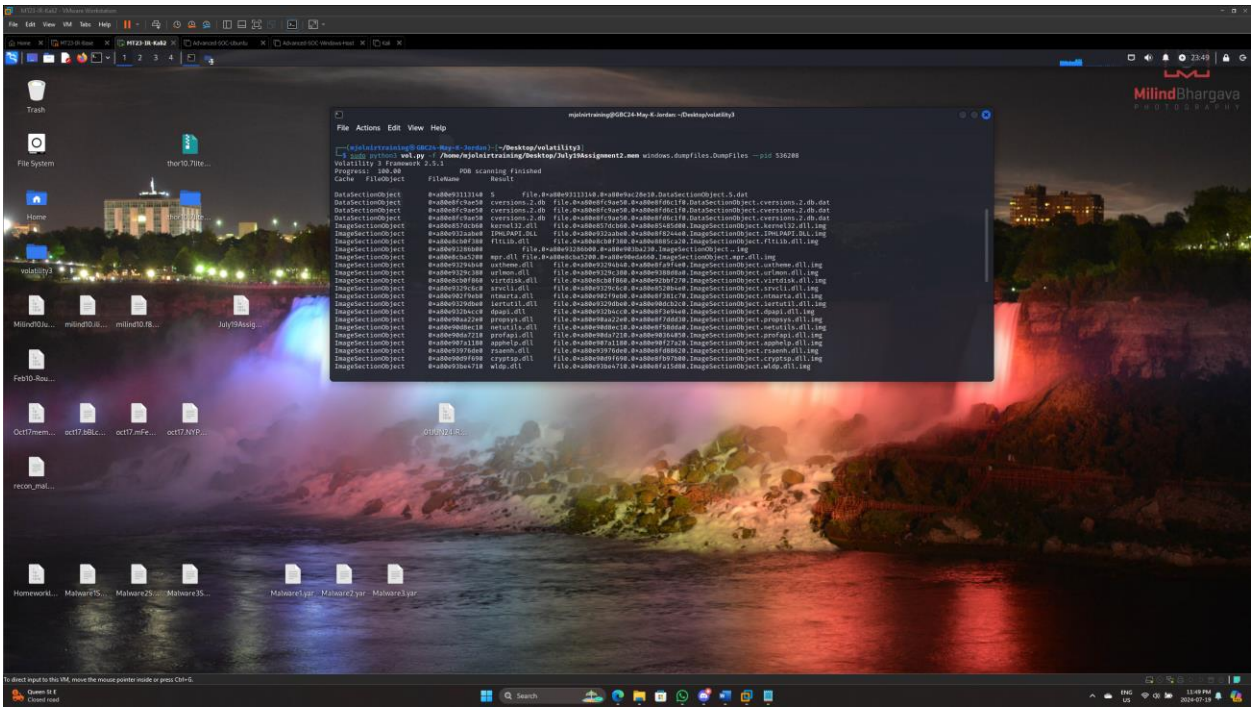
sudo python3 vol.py -f /home/mjolnirtraining/Desktop/July19Assignment2.mem windows.dumpfiles.DumpFiles --pid 10708



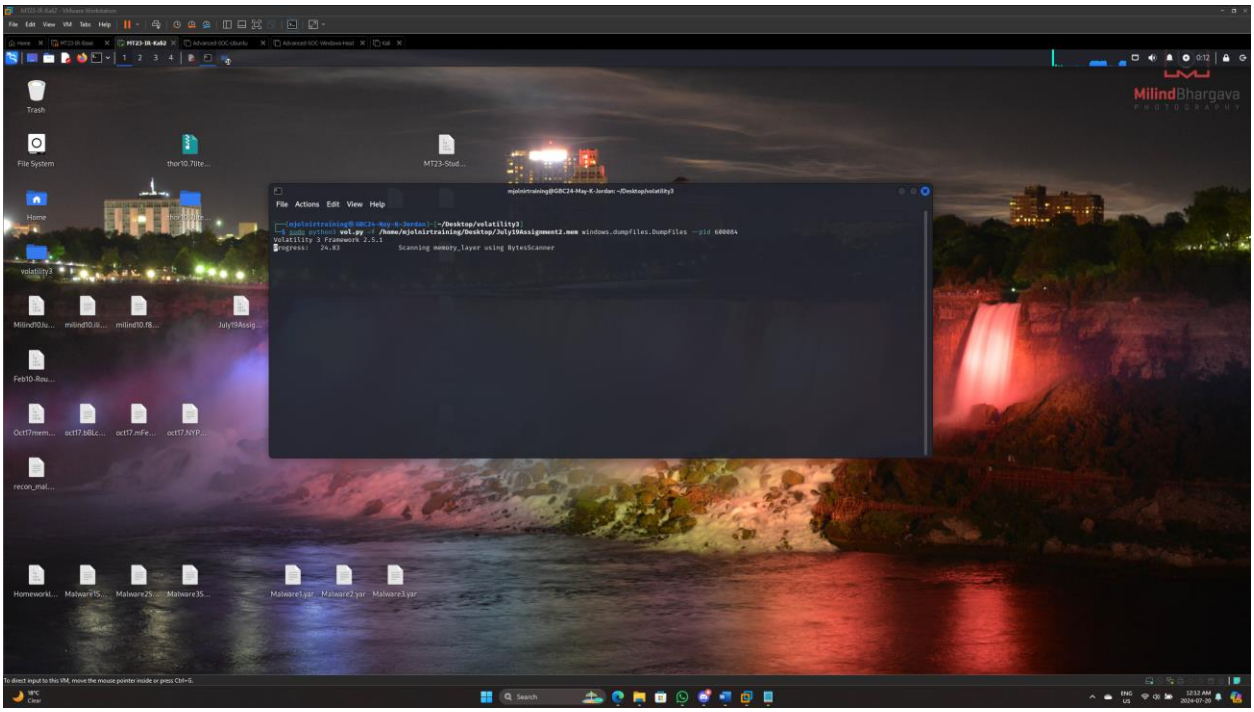
sudo python3 vol.py -f /home/mjolnirtraining/Desktop/July19Assignment2.mem windows.dumpfiles.DumpFiles --pid 5068



sudo python3 vol.py -f /home/mjolnirtraining/Desktop/July19Assignment2.mem windows.dumpfiles.DumpFiles --pid 536208



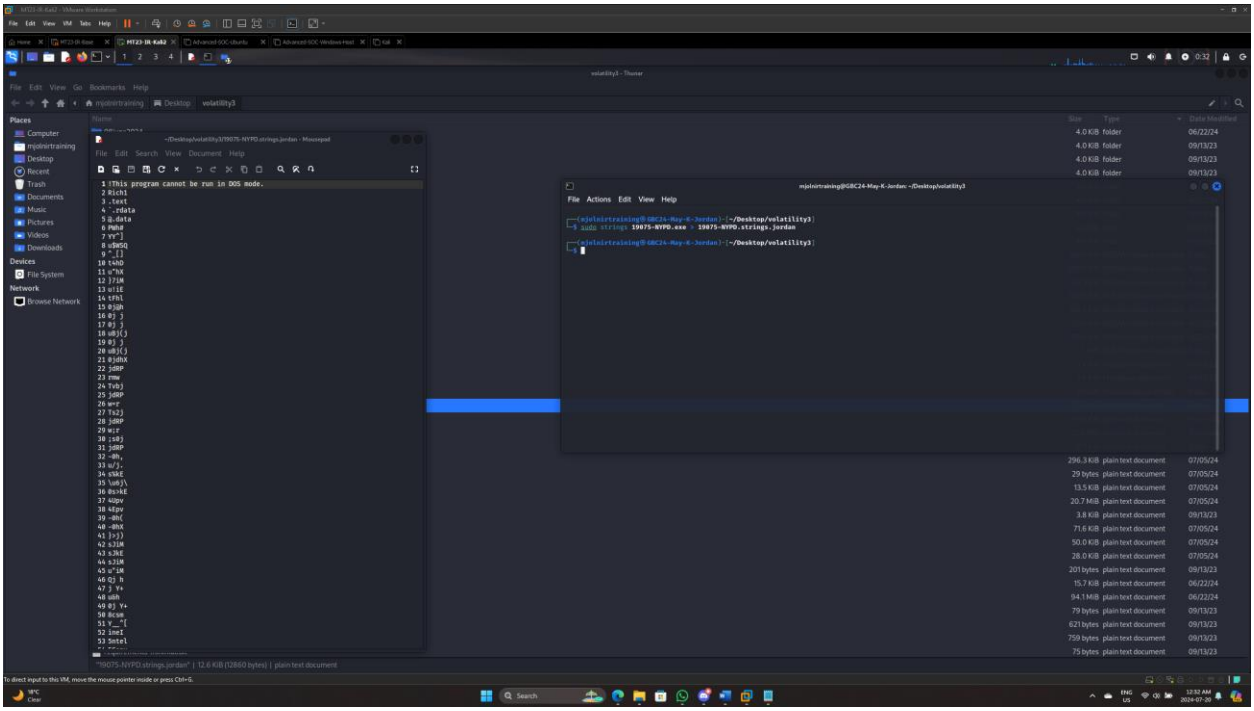
sudo python3 vol.py -f /home/mjolnirtraining/Desktop/July19Assignment2.mem windows.dumpfiles.DumpFiles --pid 600084



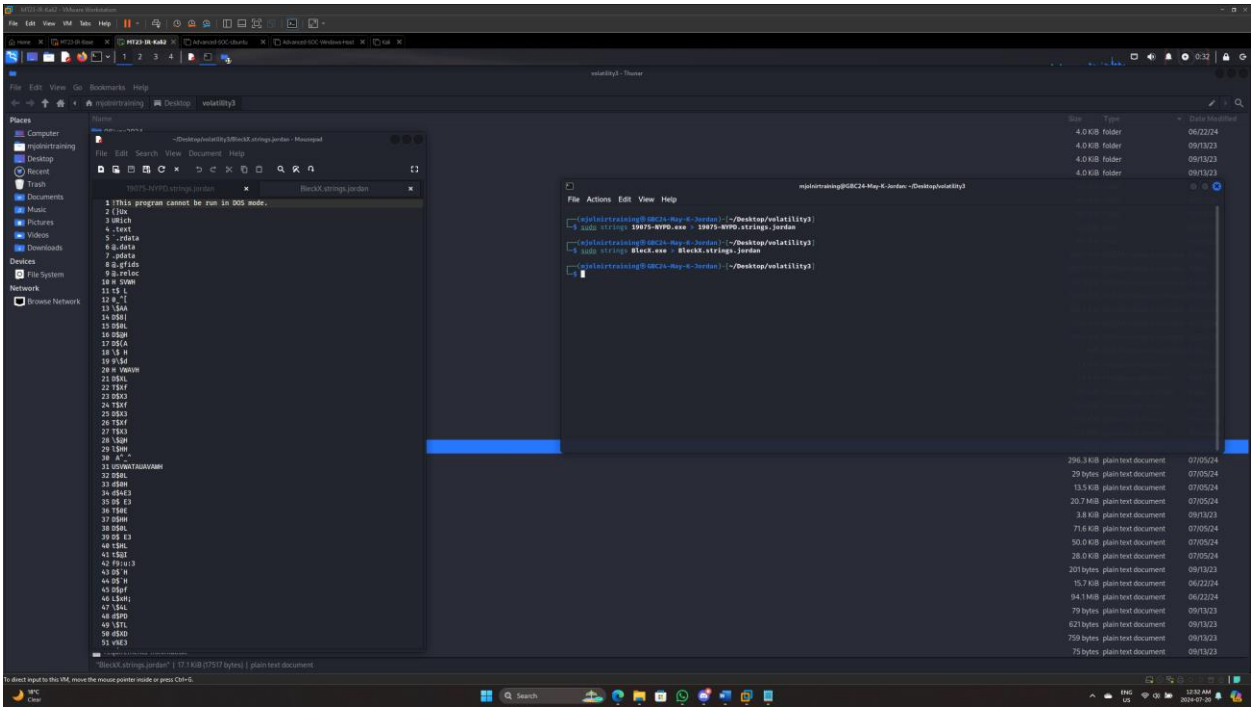
Once this was complete, I duplicated the ransomware files and renamed them so that I had the executables. I then removed .img and .dat files.

I then ran the strings command for each executable in my possession using the following commands:

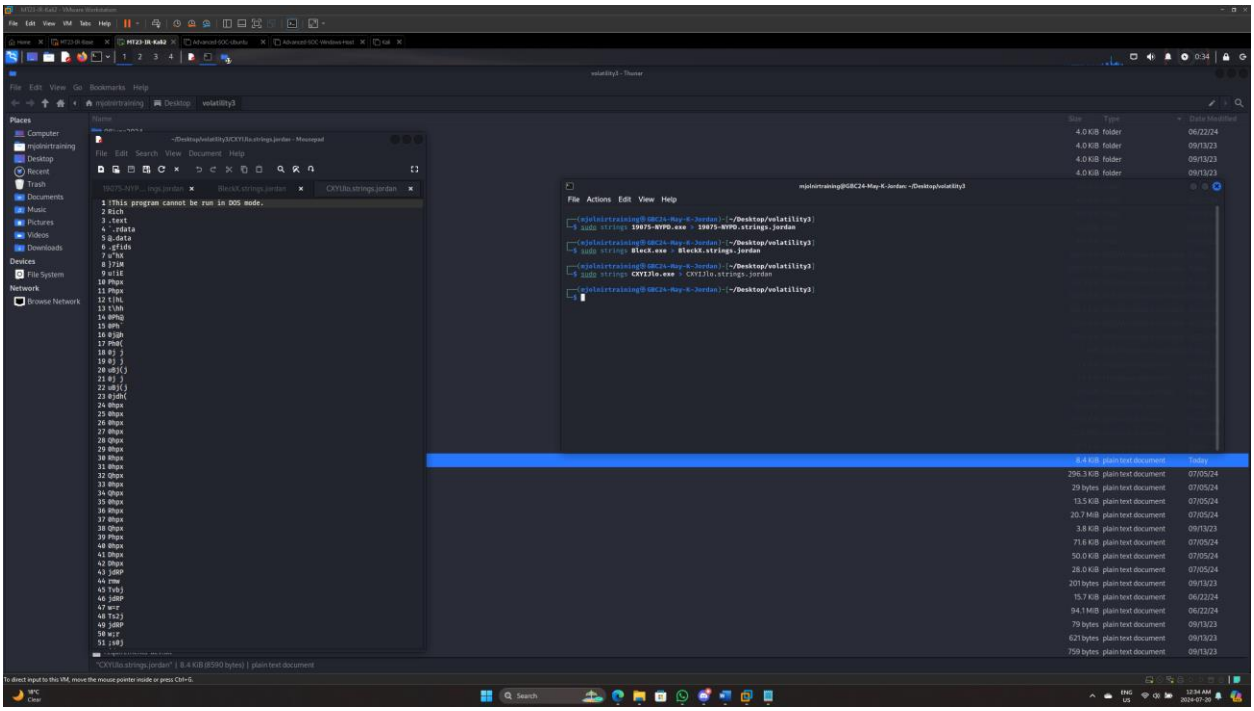
sudo strings 19075-NYPD.exe > 19075-NYPD.strings.jordan



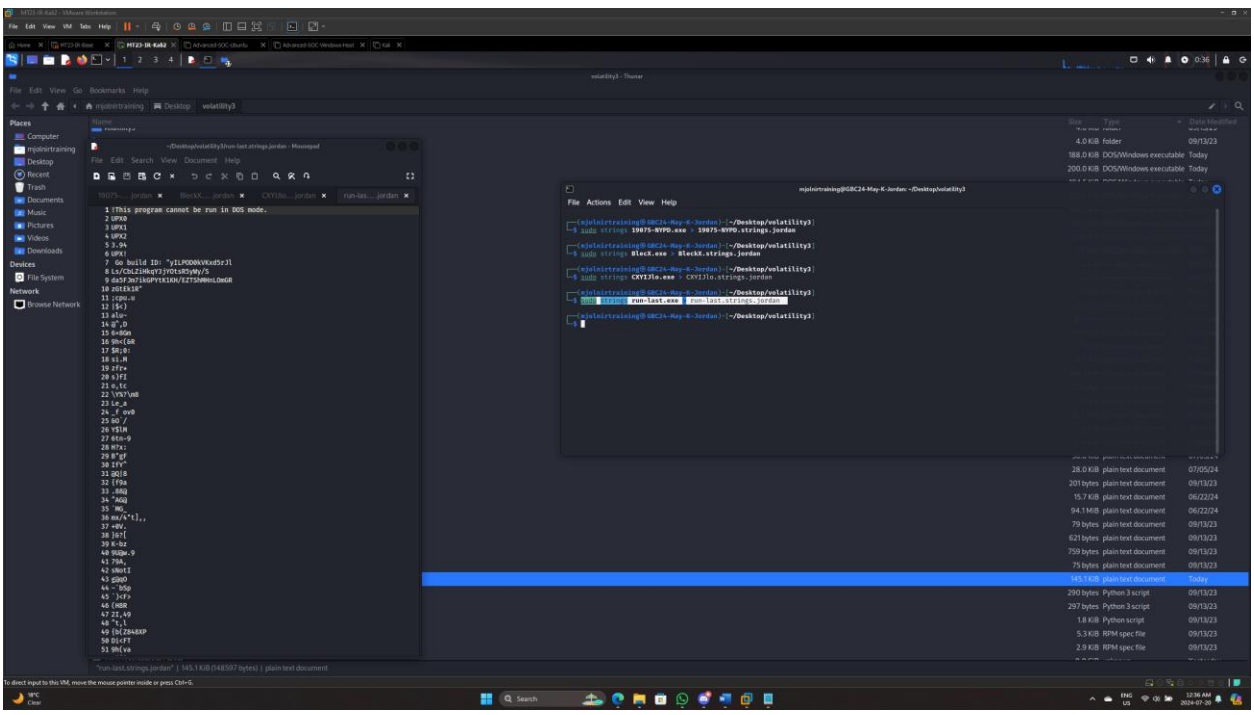
sudo strings BlecX.exe > BleckX.strings.jordan



sudo strings CXYIJlo.exe > CXYIJlo.strings.jordan

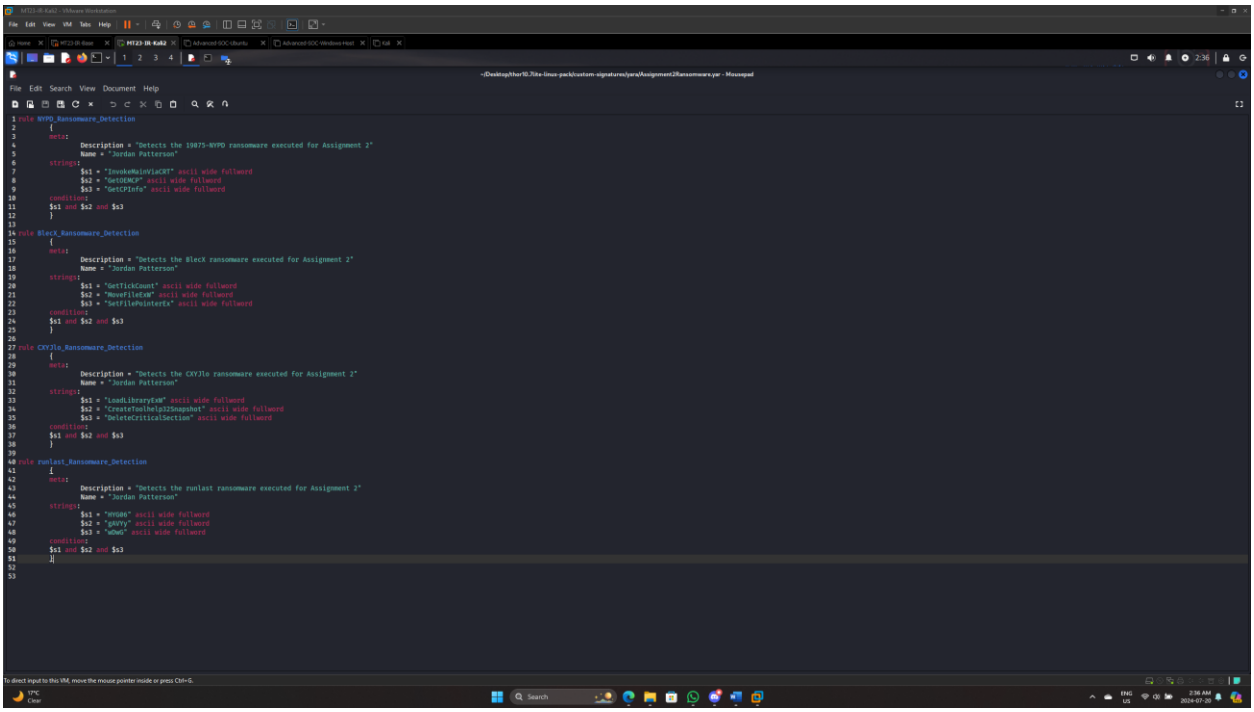


sudo strings run-last.exe > run-last.strings.jordan



Now that we have the strings for all the ransomware, we can begin creating YARA rules.

I created the following YARA rules for each ransomware:



```
rule NYPD_Ransomware_Detection
        {
        meta:
                Description = "Detects the 19075-NYPD ransomware executed for Assignment 2"
                Name = "Jordan Patterson"
        strings:
                $s1 = "InvokeMainViaCRT" ascii wide fullword
                $s2 = "GetOEMCP" ascii wide fullword
                $s3 = "GetCPInfo" ascii wide fullword
        condition:
        $s1 and $s2 and $s3
        }

rule BlecX_Ransomware_Detection
        {
        meta:
                Description = "Detects the BlecX ransomware executed for Assignment 2"
                Name = "Jordan Patterson"
        strings:
                $s1 = "GetTickCount" ascii wide fullword
                $s2 = "MoveFileExW" ascii wide fullword
                $s3 = "SetFilePointerEx" ascii wide fullword
        condition:
        $s1 and $s2 and $s3
        }

rule CXYJlo_Ransomware_Detection
        {
        meta:
                Description = "Detects the CXYJlo ransomware executed for Assignment 2"
                Name = "Jordan Patterson"
        strings:
                $s1 = "LoadLibraryExW" ascii wide fullword
                $s2 = "CreateToolhelp32Snapshot" ascii wide fullword
                $s3 = "DeleteCriticalSection" ascii wide fullword
        condition:
        $s1 and $s2 and $s3
        }

rule runlast_Ransomware_Detection
        {
        meta:
                Description = "Detects the runlast ransomware executed for Assignment 2"
                Name = "Jordan Patterson"
        strings:
                $s1 = "MYG00" ascii wide fullword
                $s2 = "gAnYy" ascii wide fullword
                $s3 = "mDwG" ascii wide fullword
        condition:
        $s1 and $s2 and $s3
        }
```

rule NYPD_Ransomware_Detection

{

meta:

Description = "Detects the 19075-NYPD ransomware executed for Assignment 2"

Name = "Jordan Patterson"

strings:

$s1 = "InvokeMainViaCRT" ascii wide fullword

$s2 = "GetOEMCP" ascii wide fullword

$s3 = "GetCPInfo" ascii wide fullword

condition:

$s1 and $s2 and $s3

}

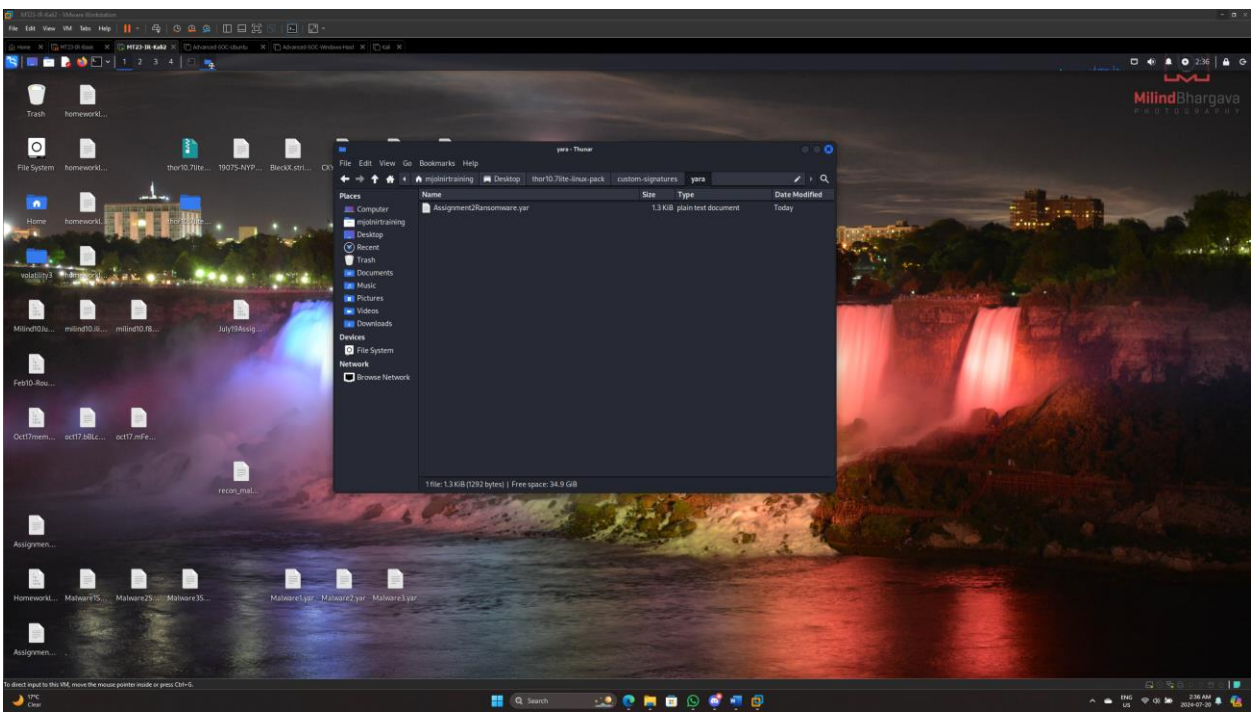
```
rule BlecX_Ransomware_Detection
	{
	meta:
		Description = "Detects the BlecX ransomware executed for Assignment 2"
		Name = "Jordan Patterson"
	strings:
		$s1 = "GetTickCount" ascii wide fullword
		$s2 = "MoveFileExW" ascii wide fullword
		$s3 = "SetFilePointerEx" ascii wide fullword
	condition:
	$s1 and $s2 and $s3
	}

rule CXYJlo_Ransomware_Detection
	{
	meta:
		Description = "Detects the CXYJlo ransomware executed for Assignment 2"
		Name = "Jordan Patterson"
	strings:
		$s1 = "LoadLibraryExW" ascii wide fullword
		$s2 = "CreateToolhelp32Snapshot" ascii wide fullword
		$s3 = "DeleteCriticalSection" ascii wide fullword
	condition:
	$s1 and $s2 and $s3
	}
```

rule runlast_Ransomware_Detection

```
{
meta:
        Description = "Detects the runlast ransomware executed for Assignment 2"
        Name = "Jordan Patterson"
strings:
        $s1 = "HYG06" ascii wide fullword
        $s2 = "gAVYy" ascii wide fullword
        $s3 = "wDwG" ascii wide fullword
condition:
$s1 and $s2 and $s3
}
```

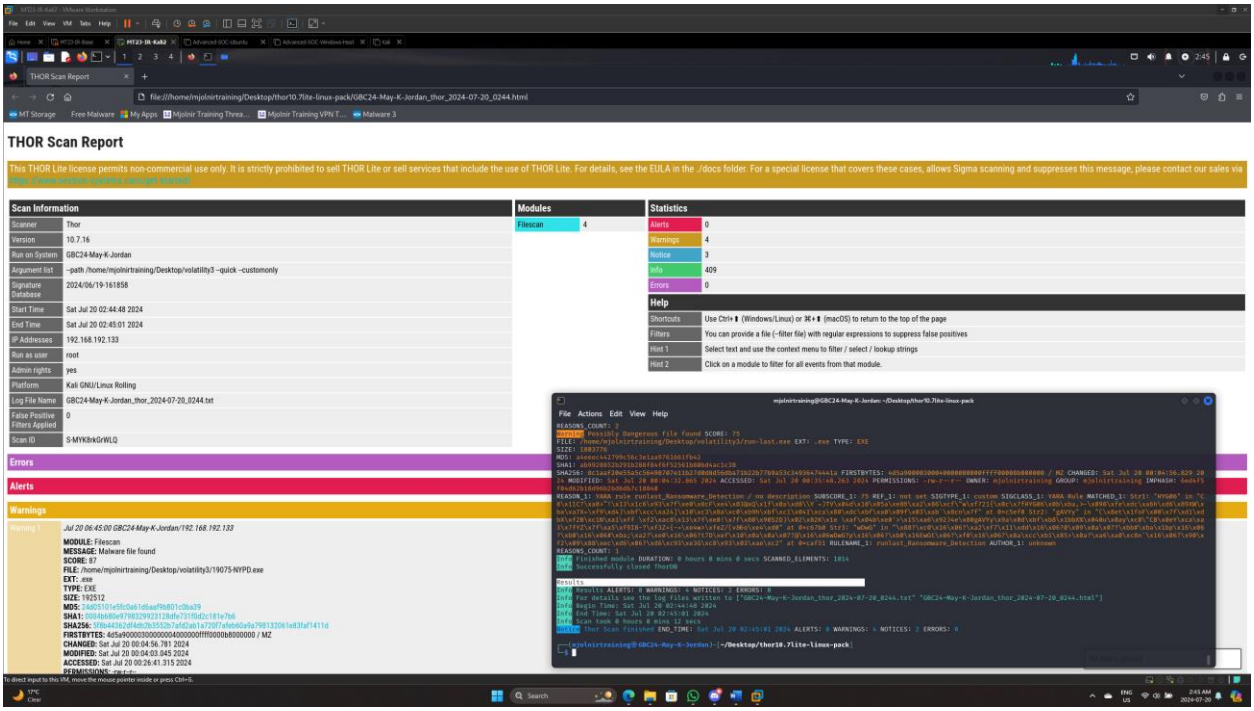
I then placed my custom rules file in Thor Lite's custom signatures folder



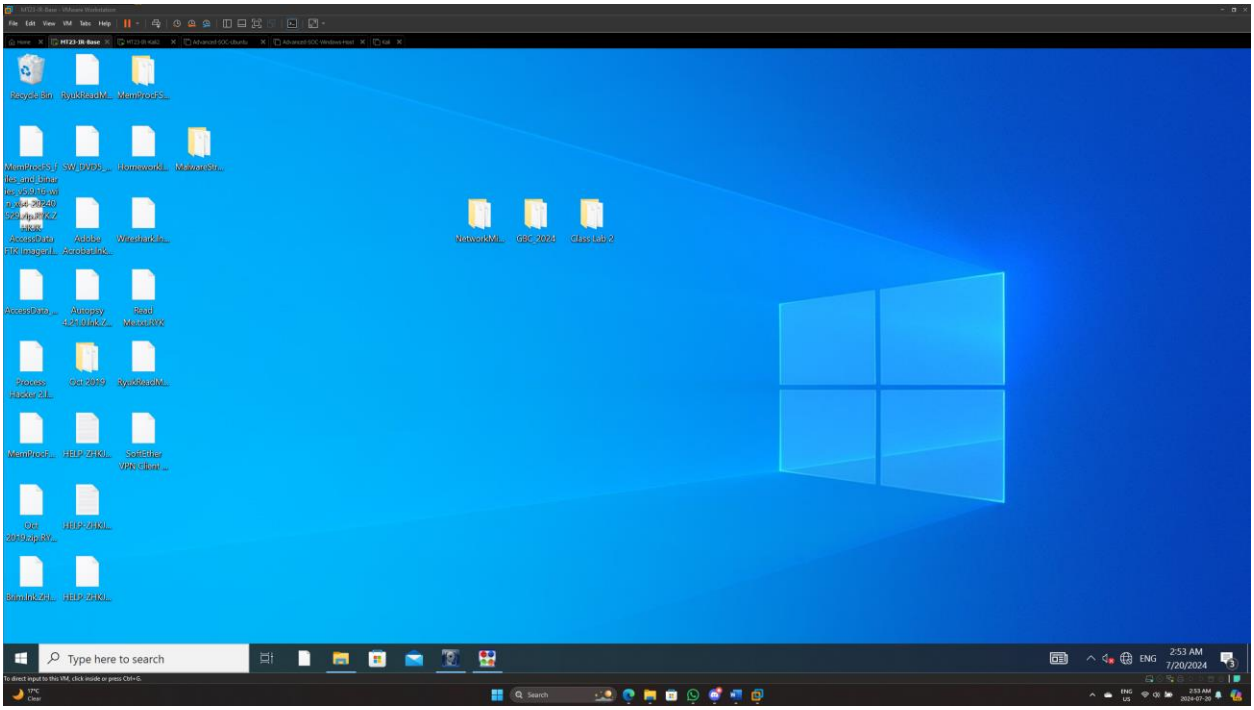
I ran Thor Lite using the command below. I did this to test the validity of my rules before taking the time to transfer them over to the Windows VM.

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

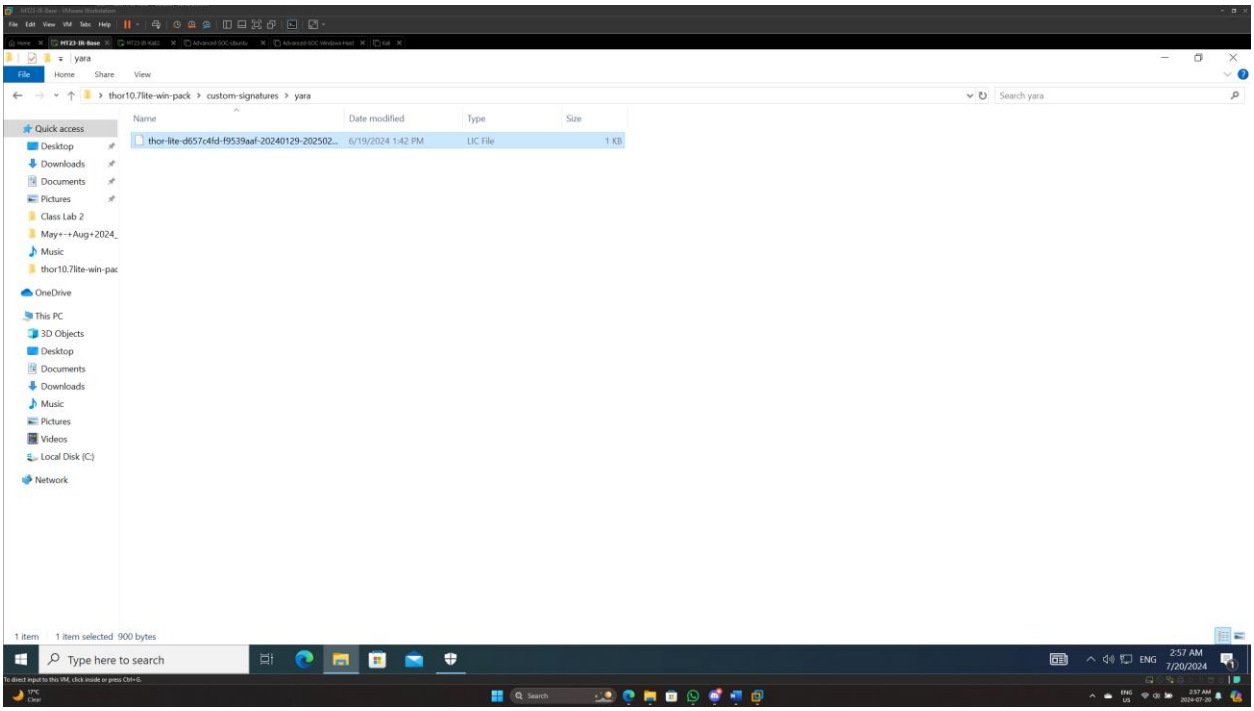
The YARA rules successfully detected the 4 ransomware files in Kali without detecting anything else.



I returned to my Windows VM and un-paused it but within moments the remainder of my files were encrypted including VMware tools, preventing me from transferring over my custom Yara Signatures.

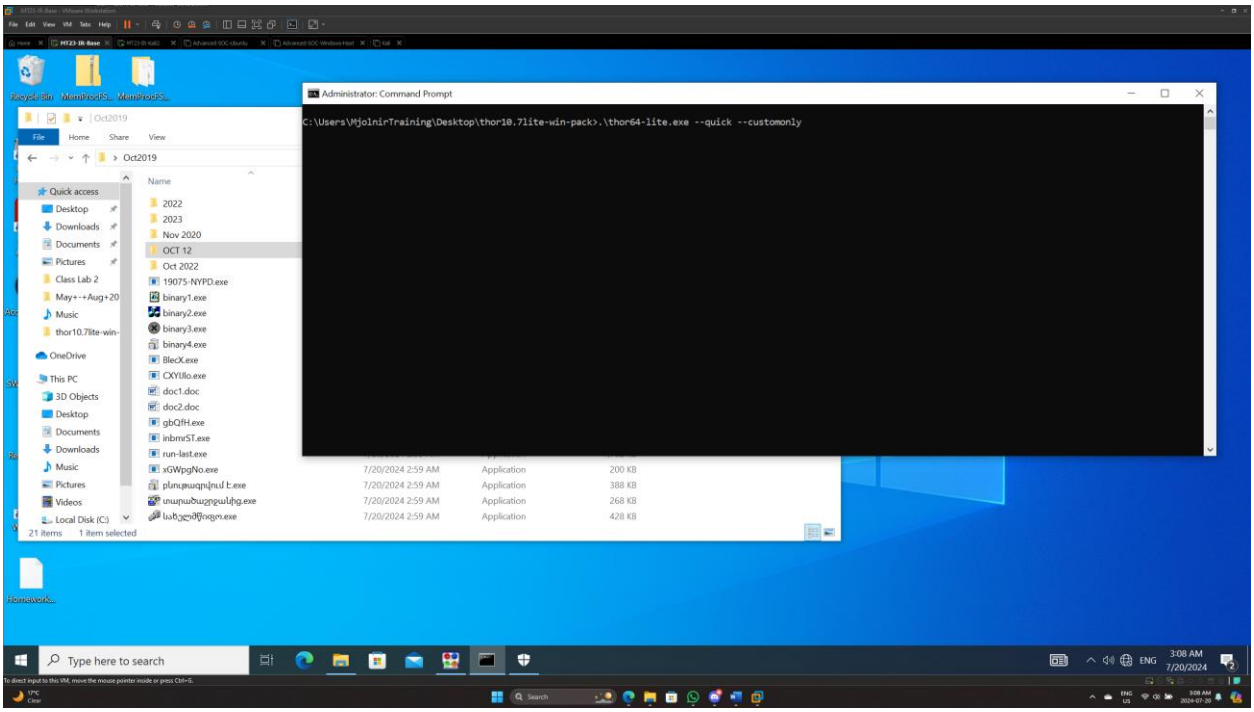


I reloaded the VM and transferred over my custom Yara Signatures.

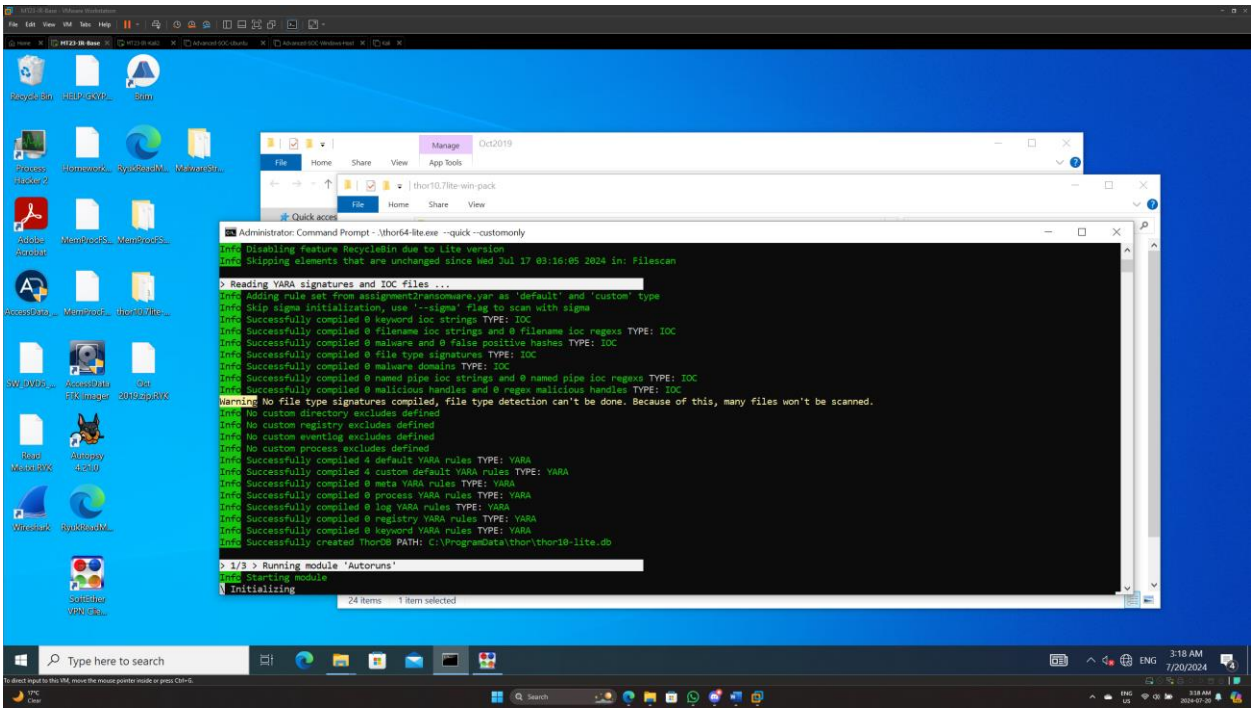


I prepared the following command so that I could run it as soon as the ransomware was ran.

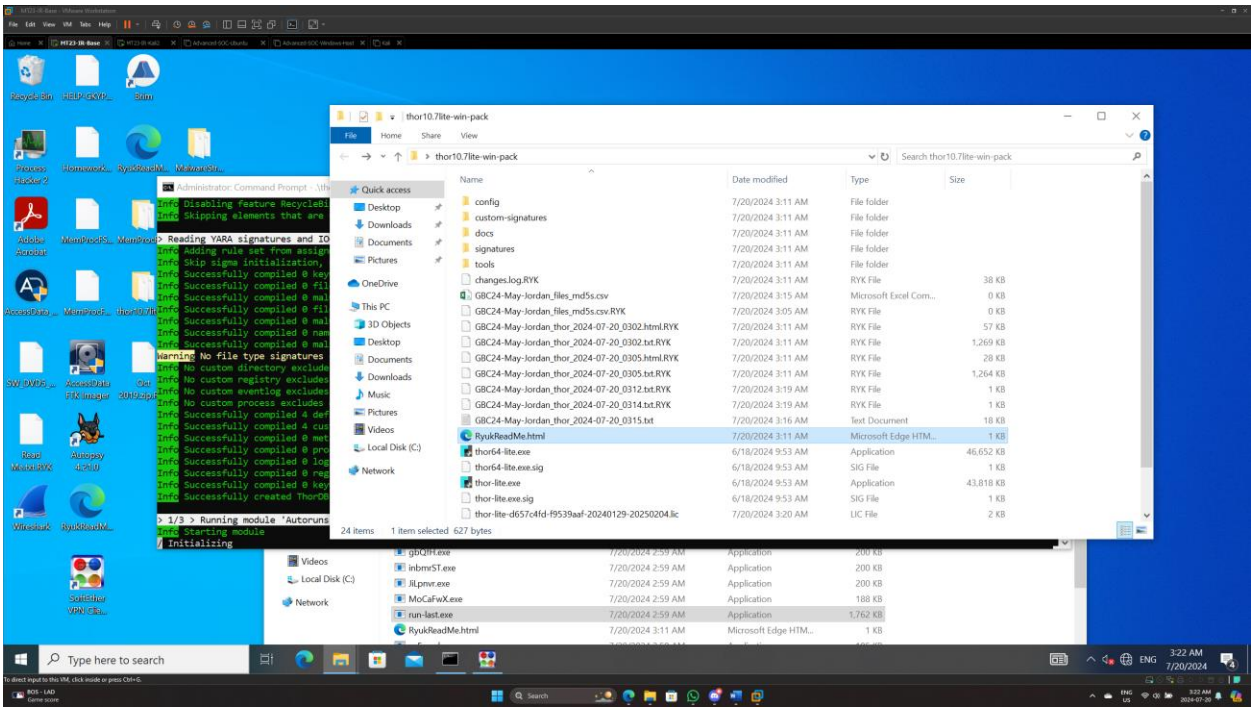
.\thor64-lite.exe --quick --customonly



I turned on my VPN, turned off smartscreen, and re-ran all of the ransomware. I quickly ran a Thor Lite scan before the tool could become encrypted.

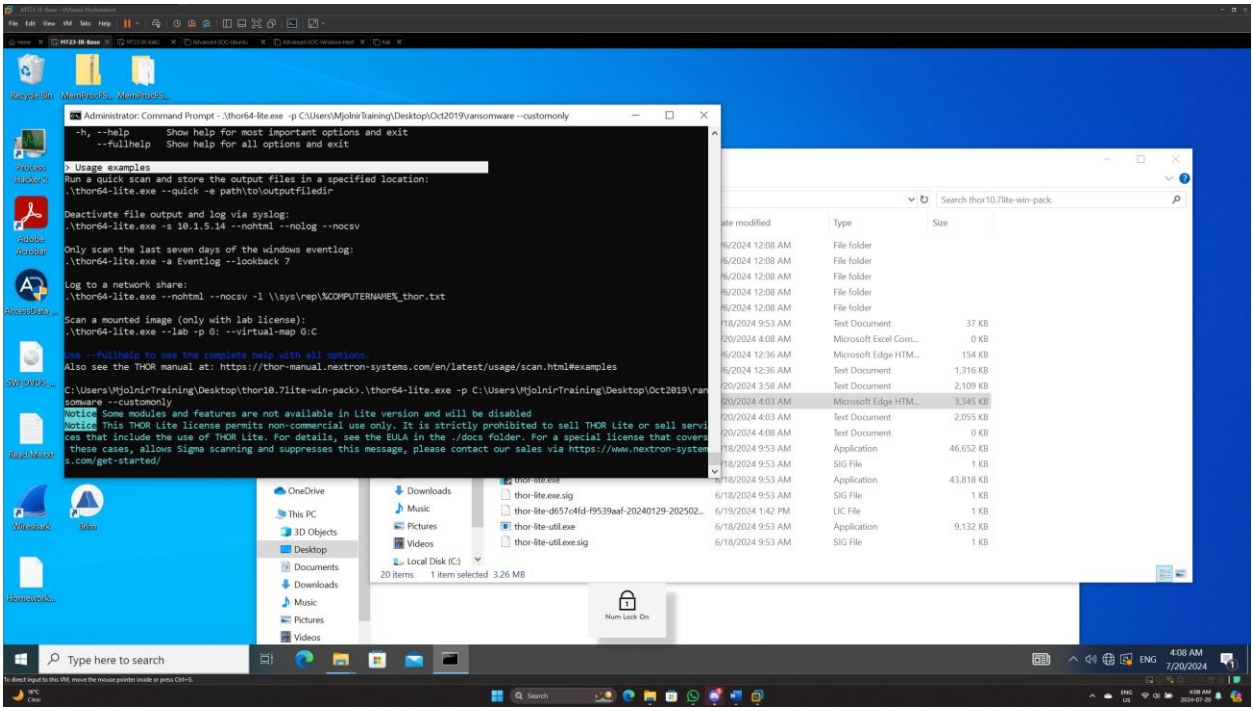


Unfortunately, the tool and my custom signatures became encrypted during the scan, causing the scan to hang.



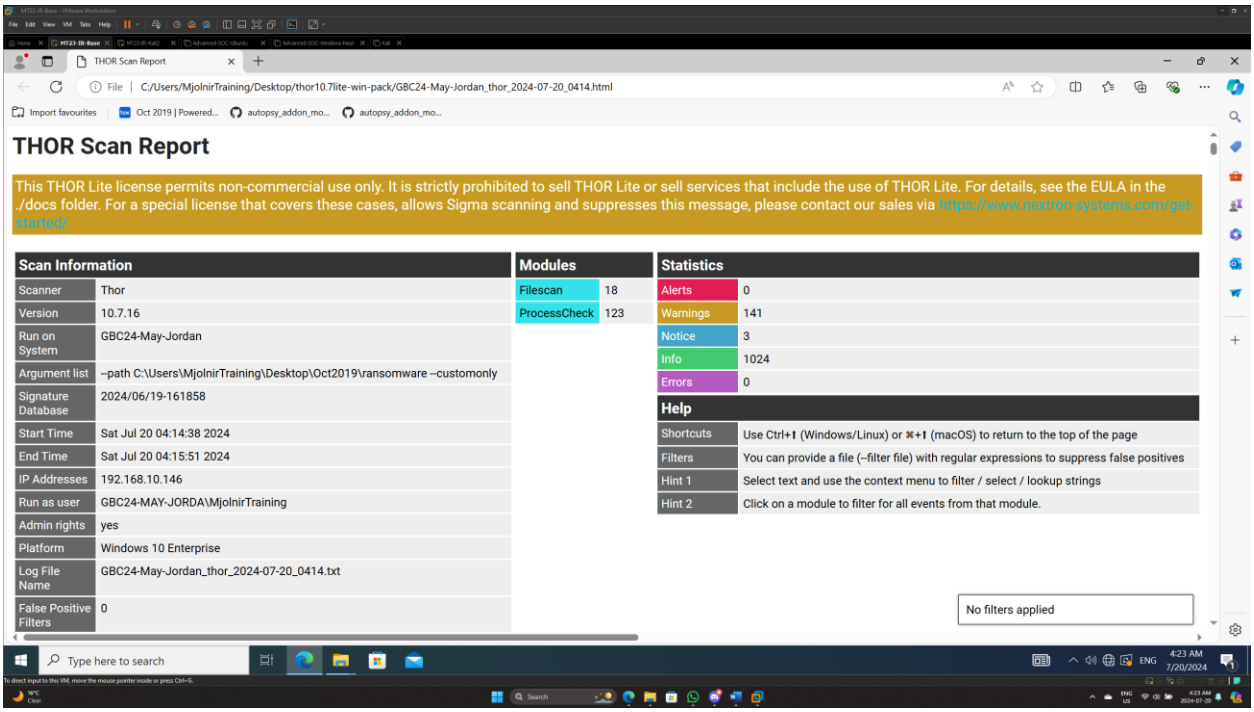
I restarted the VM and reverted to a clean state. I re-ran the scan pointing Thor Lite directly to a directory containing the Ransomware files using the following command.

.\thor64-lite.exe -p C:\Users\MjolnirTraining\Desktop\Oct2019\ransomware --customonly



Despite my specifying the path the scan kept going out of scope and detecting false positives. I tried multiple iterations of the command and tried tweaking my YARA rules but with no change in the result of the scan.

At this stage in the lab I am unsure what's gone wrong with my YARA rules and Thor Lite scan on Windows as they seemed to work perfectly in KALI.



Conclusion:

In conclusion, while I was able to successfully complete the necessary steps to get to the point of running my YARA rules on Windows , including getting my YARA rules working in Kali, I was unable to get my Yara rules working properly in Windows and I was unable to run the scan live while the malware was running.