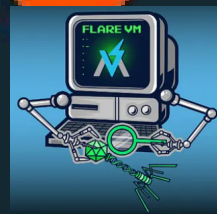


# *MALWARE ANALYSIS LIVE DETONATION*

*PROJECT*

Alexander Chait



# **Malware Analysis Lab Project: Dynamic & Static Analysis**

## **Project Goals**

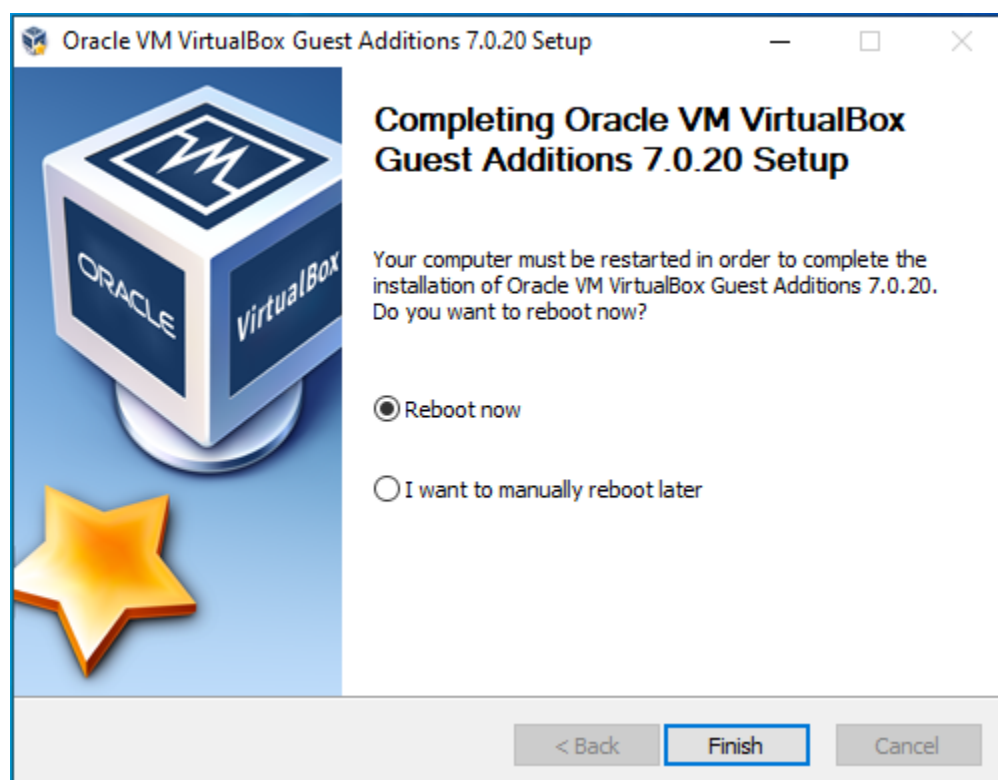
- **Dynamic Analysis:** Execute malware in a controlled environment to observe its behavior in real-time, including network activity, file system changes, and process creation.
- **Static Analysis:** Examine malware without execution to analyze its code, structure, and potential threats using disassembly, decompilation, and signature-based techniques.

# Installation & Configurations

I'll begin with installing the Virtual Machine



And add Guest Additions for QoL



Next, I'll configure the VM by following these steps:

► **Disable Windows Update**

(Go to Services.exe -> Windows Update -> Click Stop -> Startup type is 'Disabled'-> Apply then OK)

► **Disable Windows Defender**

(Go to windows Security-> Manage Settings → Realtime protection off -> Cloud delivered off -> Automatic sampling off -> Tamper protection off)

Then Do (window+R-> gpedit.msc. -> Administrator templates -> Windows Components -> Microsoft defender anti-virus -> Real time protection -> Enable Turn off Realtime protection → Enable Turn off Microsoft windows defender anti virus)  
REBOOT

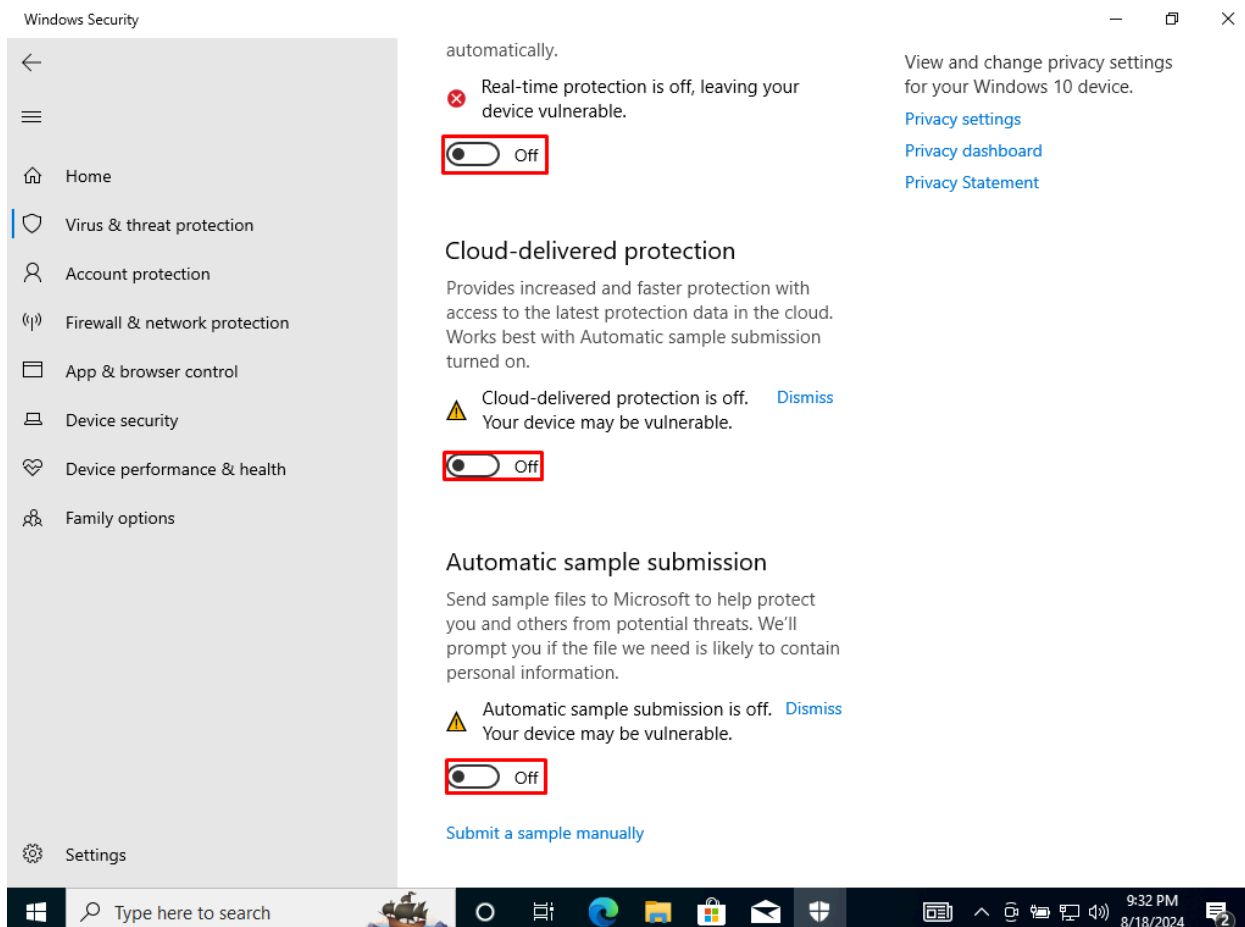
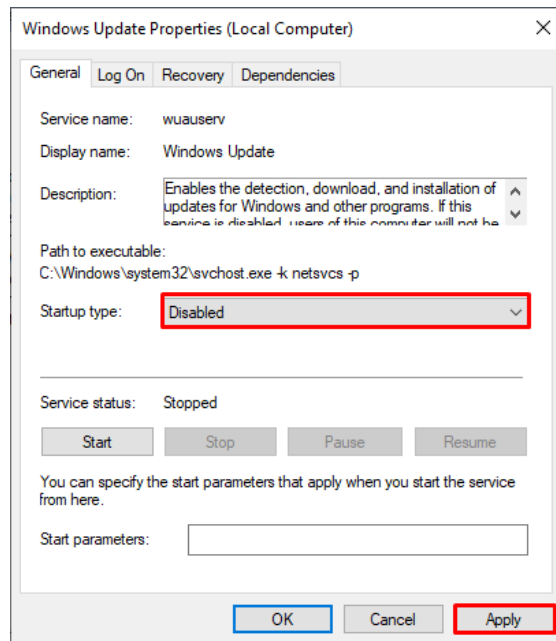
► **Disable Hide Extensions**

Open file explorer >> View Options >> Change options->view-> uncheck hide extensions for known filetypes>> Also check the second circle

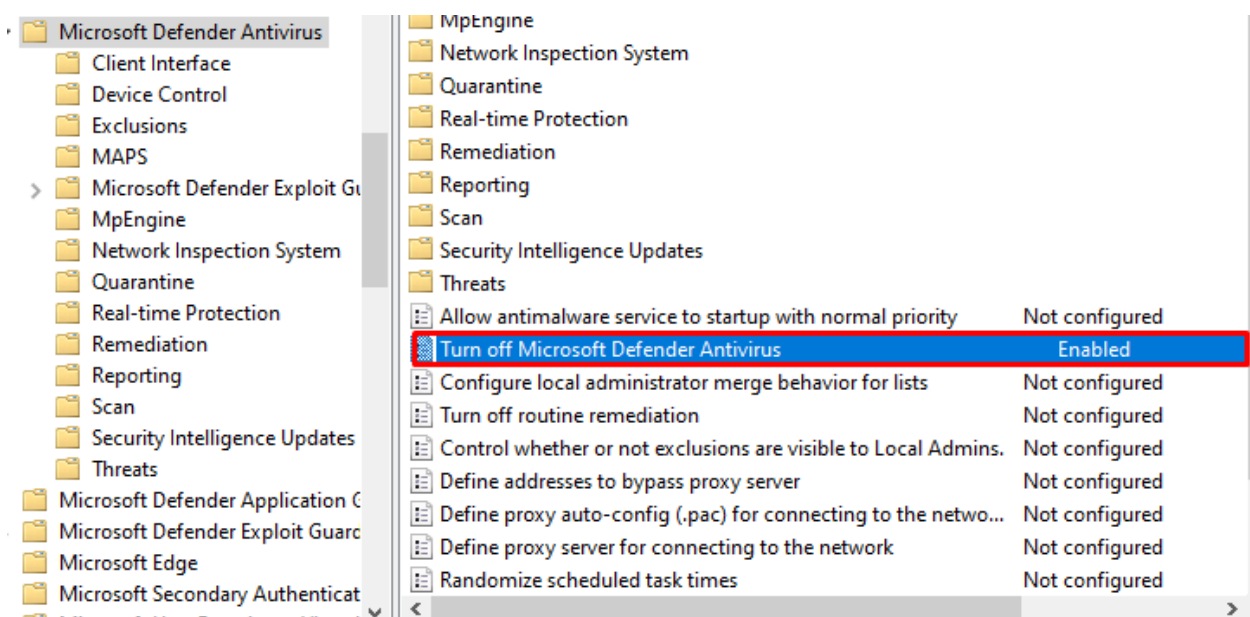
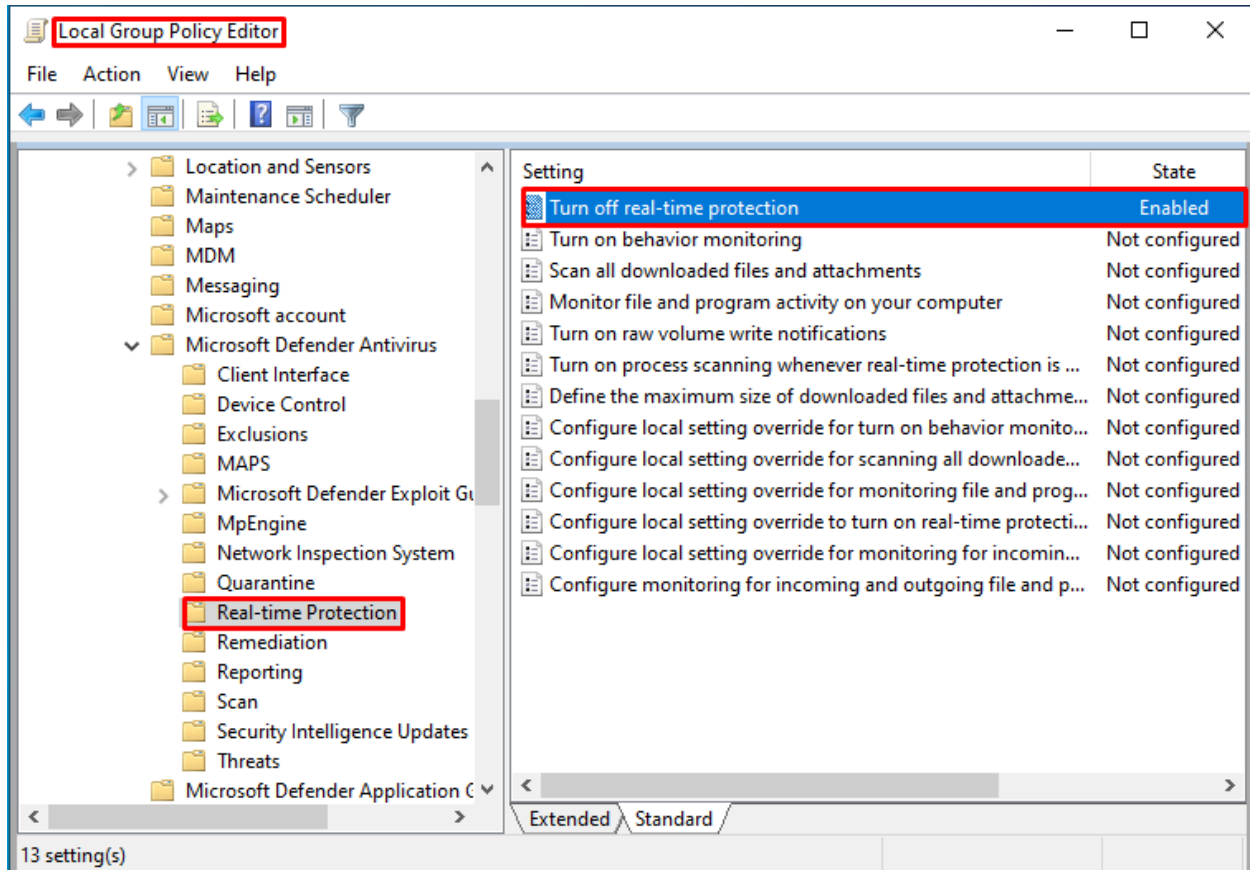
► **Show Hidden Files and Folders**

► **Create a Snapshot**

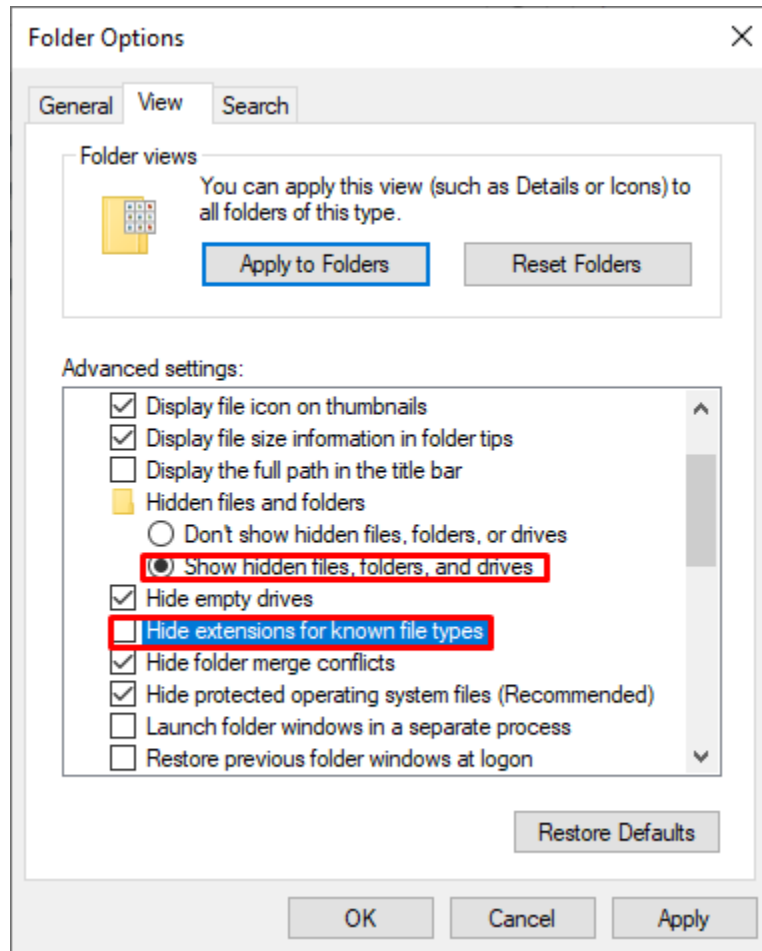
# I'll start with disabling the Windows update & Windows Defender



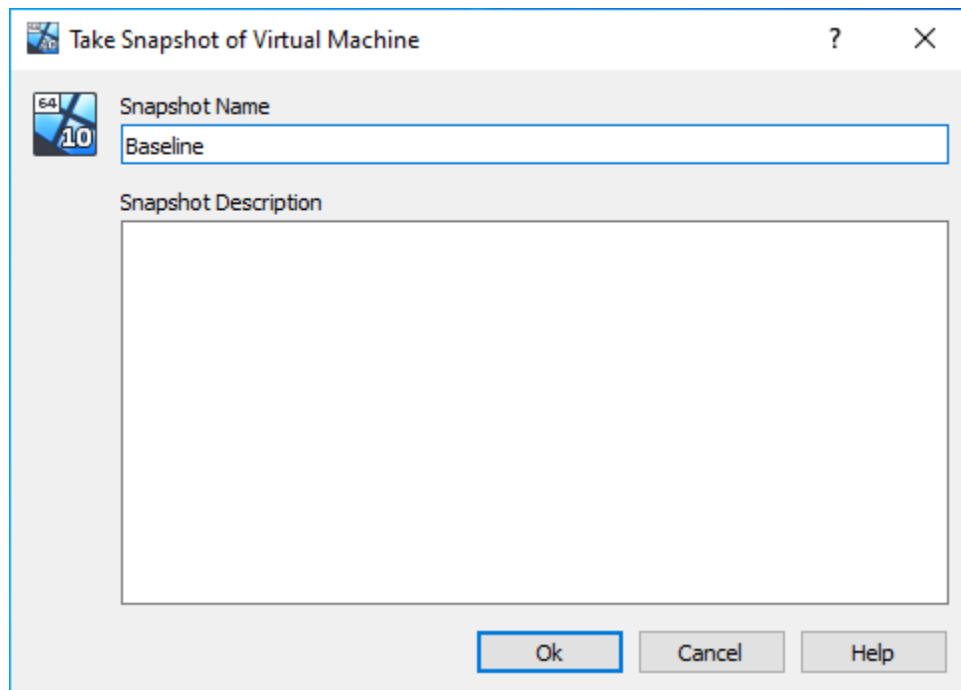
I'll also disable the AntiVirus+real-time protection using the LGP Editor



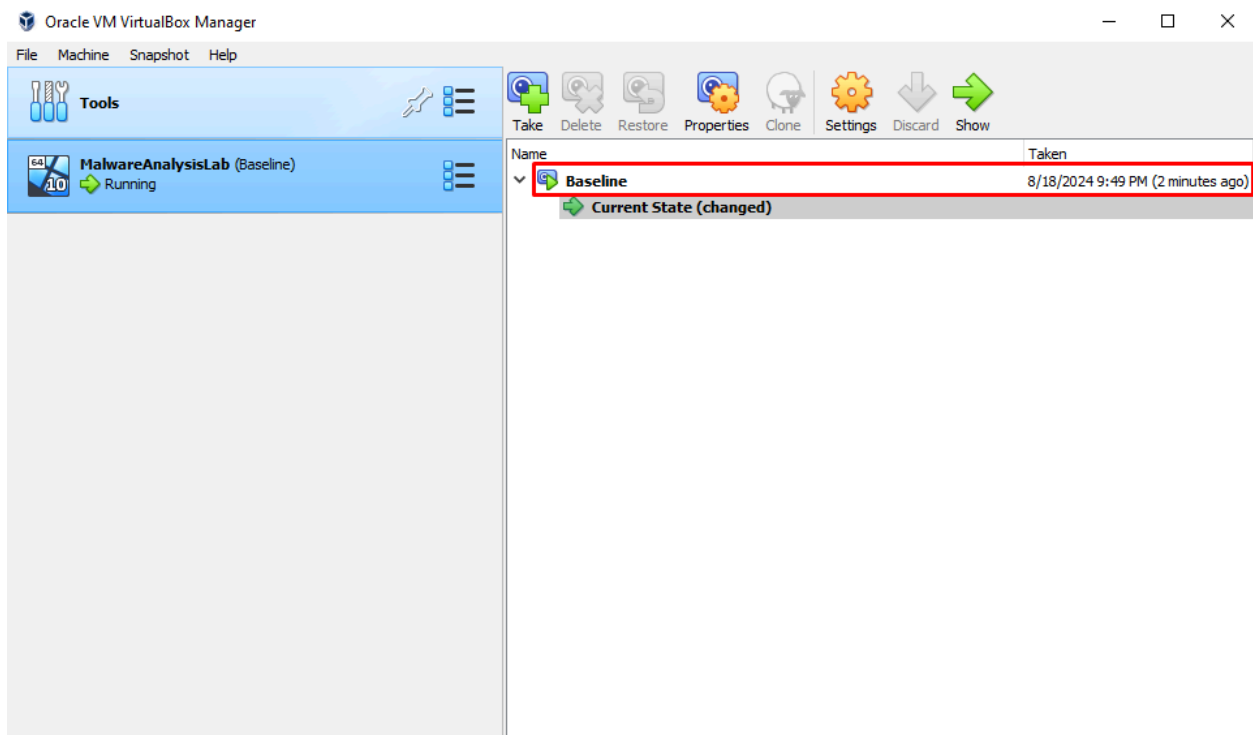
And I'll reboot after the changes, next I will disable the hide extensions & show hidden files



After configuring the VM, I'll take a baseline snapshot



The Snapshot could be useful in case I make a mistake, I could revert to the Baseline snapshot.





For now, I'll shutdown the machine, and do some needed configurations on VirtualBox, after that I'll start installing FlareVM

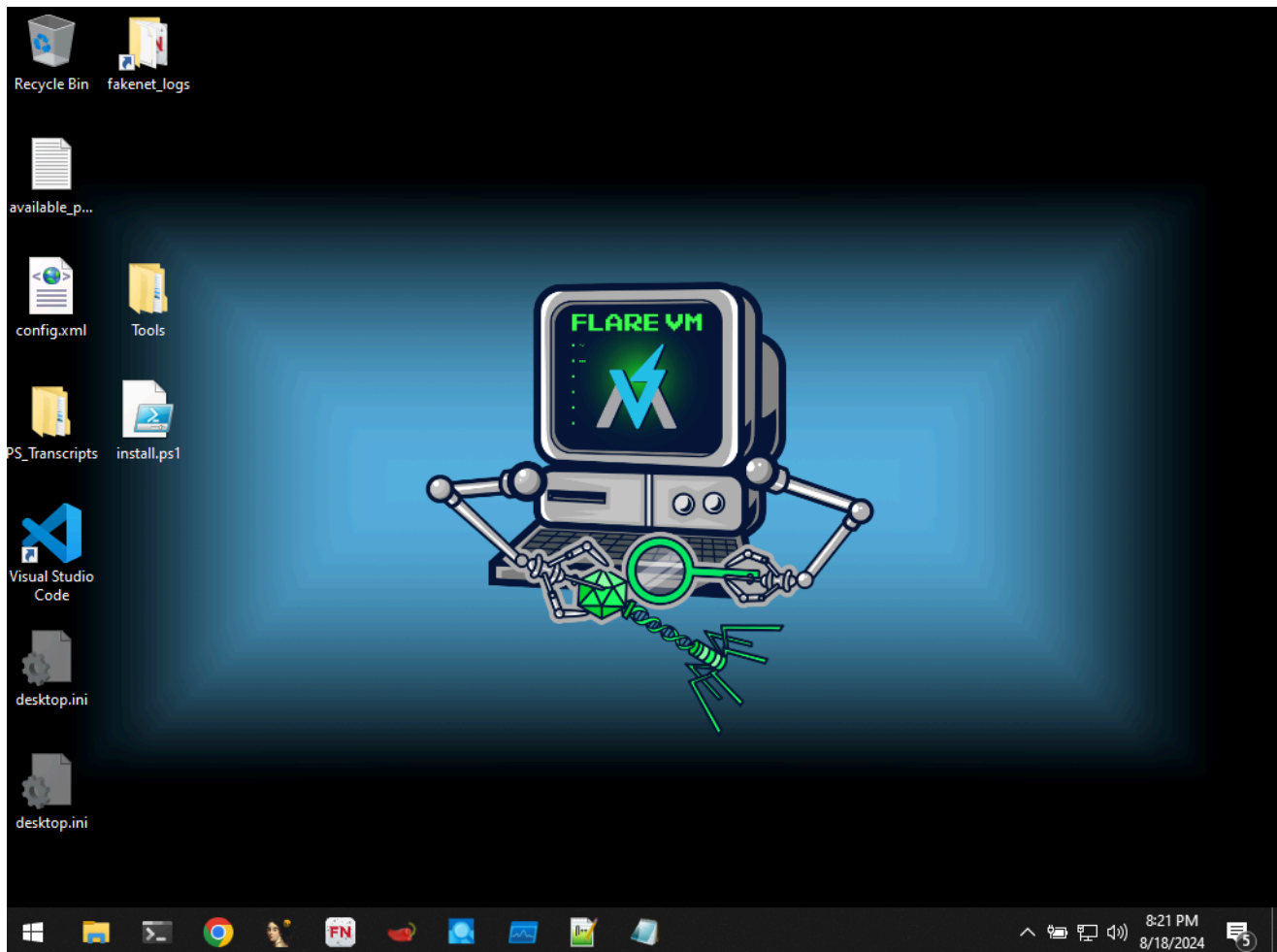
(FlareVM is an open-source Windows-based virtual machine (VM) designed for malware analysis, reverse engineering, and incident response.)

I'll start PowerShell with Administration privileges & follow the installation steps below

#### FLARE-VM installation

- Open a PowerShell prompt as administrator
- Download the installation script `installer.ps1` to your Desktop:
  - `(New-Object net.webclient).DownloadFile('https://raw.githubusercontent.com/mandiant/flare-vm/main/install.ps1','$([Environment]::GetFolderPath("Desktop"))\install.ps1')`
- Unblock the installation script:
  - `Unblock-File .\install.ps1`
- Enable script execution:
  - `Set-ExecutionPolicy Unrestricted -Force`
    - If you receive an error saying the execution policy is overridden by a policy defined at a more specific scope, you may need to pass a scope in via `Set-ExecutionPolicy Unrestricted -Scope CurrentUser -Force`. To view execution policies for all scopes, execute `Get-ExecutionPolicy -List`
- Finally, execute the installer script as follow:
  - `.\install.ps1`
    - To pass your password as an argument: `.\install.ps1 -password <password>`
    - To use the CLI-only mode with minimal user interaction: `.\install.ps1 -password <password> -noWait -noGui`
    - To use the CLI-only mode with minimal user interaction and a custom config file: `.\install.ps1 -customConfig <config.xml> -password <password> -noWait -noGui`
- After installation it is recommended to switch to `host-only` networking mode and take a VM snapshot

Flare VM is now installed with the correct configurations



Next, I'll make sure these following tools are installed:

**FakeNet:** Simulates network traffic for malware analysis.

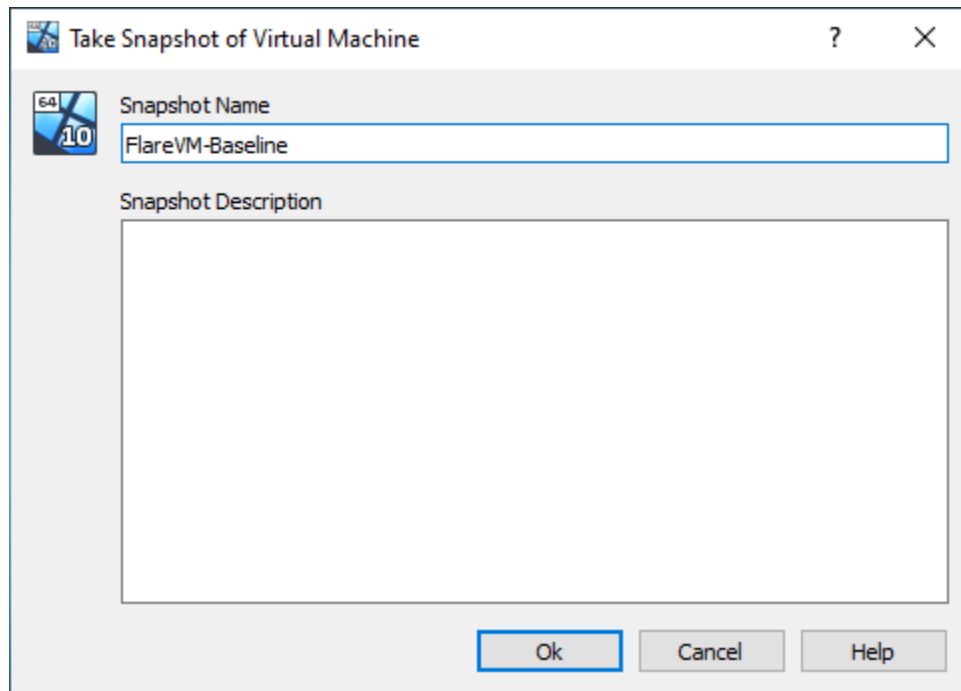
**HashMyFiles:** Calculates file hash values.

**Regshot:** Compares Windows registry snapshots.

**Ghidra:** Disassembles and analyzes binary files.

Great, they're all already installed

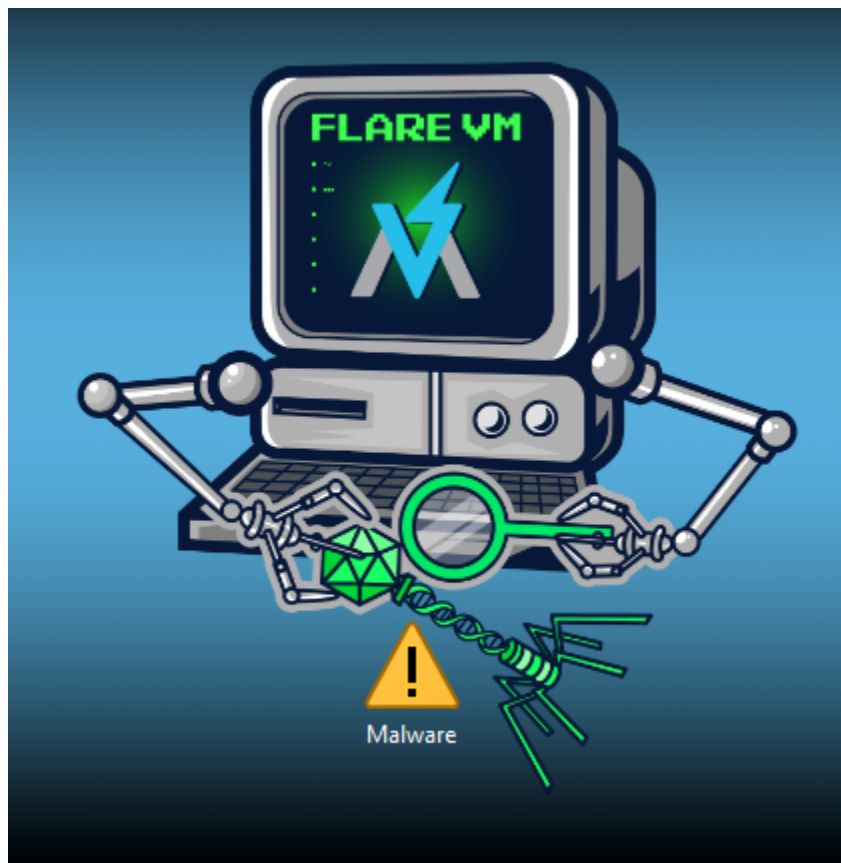
Now I'll create another snapshot of the new FlareVM-Baseline image, so I can revert if needed



Now I can begin with the static malware analysis

# Static Malware Analysis

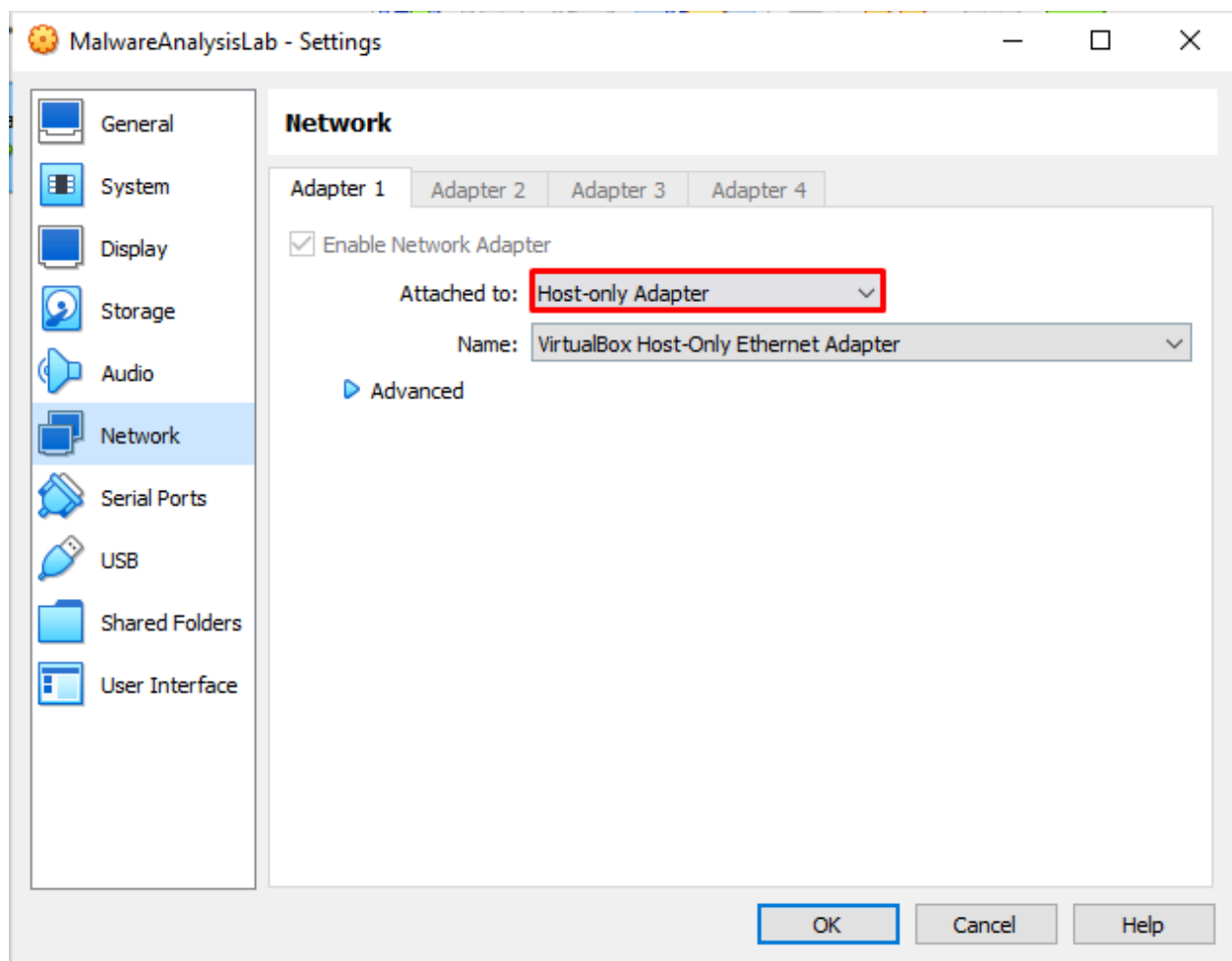
It's time to analyze some Static Malware



Before I'll start with the investigations and analysis

I will make sure that my VM is isolated from any outside connections.

Safety is key in these test since you don't want any malware transferring over to your host machine



# 1. Malware Handling and Safety

Covers basic malware handling and safety, including defanging malware and safe practices for transfer and storage.

# 2. Static Analysis

I'll covers initial triage, static analysis, initial detonation, and methodology of static analysis.

# 3. Dynamic Analysis

I'll covers initial triage, dynamic analysis, detonation, and methodology of dynamic analysis.

## - Malware Handling and Safety

**For best practice doing this Lab, I'll check the malware samples hashes, I'll make sure all the malware files downloaded are protected by a password & zipped, I'll check again to make sure there's no outside connection after changing the Network adapter to Host-Only**

**This site can't be reached**

**google.com's** server IP address could not be found.

Try:

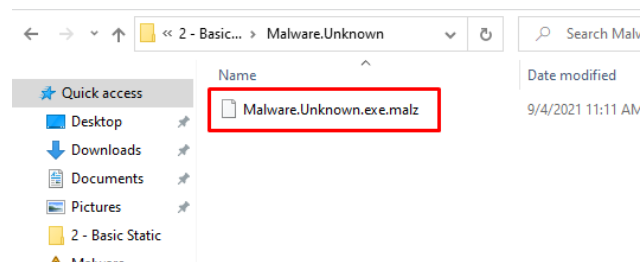
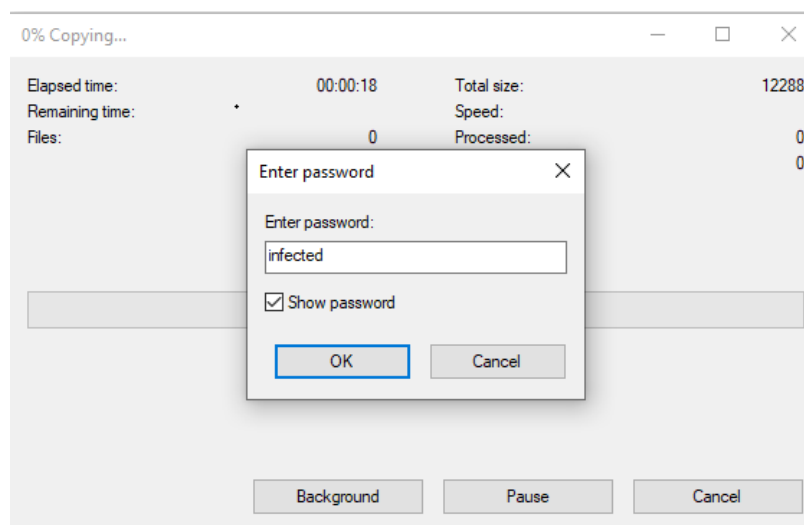
- Checking the connection
- [Checking the proxy, firewall, and DNS configuration](#)
- [Running Windows Network Diagnostics](#)

ERR\_NAME\_NOT\_RESOLVED

Name	Date modified	Type	Size
Malware.Calc.exe (1).7z	8/18/2024 10:30 PM	7Z File	41 KB
md5sum.txt	8/18/2024 10:30 PM	Text Document	1 KB
password.txt	8/18/2024 10:30 PM	Text Document	1 KB
sha256sum.txt	8/18/2024 10:30 PM	Text Document	1 KB

## - Static Analysis

Let's begin with extracting the first Malware



filetype:

```

C:\Users\Admin
λ file C:\Users\Admin\Desktop\Malware.Unknown.exe.malz
C:\Users\Admin\Desktop\Malware.Unknown.exe.malz: PE32 executable (console) Intel 80386, for MS Windows

C:\Users\Admin
λ

```

I'll begin with using the Cmder CLI to get the SHA256 & MD5 from the file

```
C:\Users\Admin\Desktop
λ sha256sum.exe Malware.unknown.exe.malz
92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a *Malware.unknown.exe.malz

C:\Users\Admin\Desktop
λ
C:\Users\Admin\Desktop
λ md5sum.exe Malware.unknown.exe.malz
1d8562c0adcaee734d63f7baaca02f7c *Malware.unknown.exe.malz

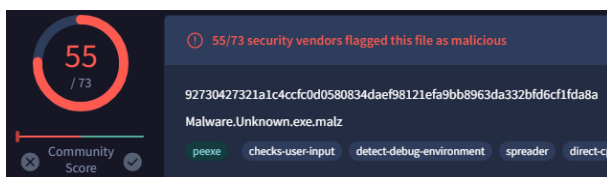
C:\Users\Admin\Desktop
λ |
```

```
NotesMalware.txt - Notepad
File Edit Format View Help
SHA256

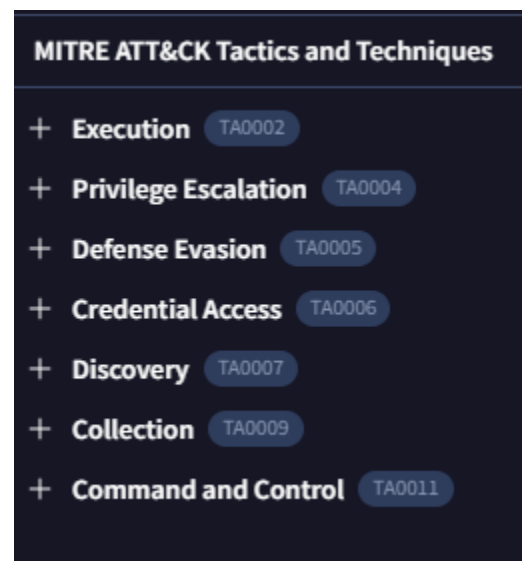
92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a *Malware.unknown.exe.malz

MD5
|
1d8562c0adcaee734d63f7baaca02f7c *Malware.unknown.exe.malz
```

Now I will go to the Host Machine in order to scan the file hashes



We can see it's an C&C Trojan  
Privilege Escalation Executable





Next I'll be extracting the strings out of the binary code.

I'll use Floss that will also deobfuscate/decode it, so I can more easily understand the output

And I can already see some troubling Artefacts

```
KERNEL32.dll
ShellExecuteW
SHELL32.dll
_Query_perf_frequency
_Thrd_sleep
_Query_perf_counter
_Xtime_get_ticks
MSVCP140.dll
JURLDownloadToFileW
urlmon.dll
InternetOpenUrlW
InternetOpenW
WININET.dll
__current_exception
__current_exception_context
memset
```

We can even see the origin of the original Malware release, this isn't common most likely happened because this was made for educational purposes

```
C:\Users\Matt\source\repos\HuskyHacks\PMAT-maldev\src\DownloadFromURL\Release\DownloadFromURL.pdb
GCTL
text$mp
```

```
+-----+
| FLOSS STATIC STRINGS: UTF-16LE (8) |
+-----+

jjjj
cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q "%s"
http://ssl-6582datamanager.helpdeskbro.s.local/favicon.ico
C:\Users\Public\Documents\CR433101.dat.exe
Mozilla/5.0
http://huskyhacks.dev
ping 1.1.1.1 -n 1 -w 3000 > Nul & C:\Users\Public\Documents\CR433101.dat.exe
open
```

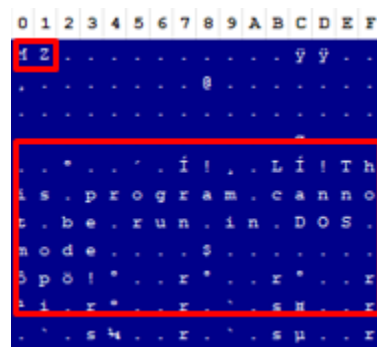
This could be thrown in to trick us or be a part of the actual functionality of the binary code, we don't know yet

**First Command:** Introduces a 3-second delay, attempts to delete a file (possibly incorrectly), and interacts with a suspicious URL.

**Second Command:** Introduces a 3-second delay, then executes a potentially malicious file from the public directory.

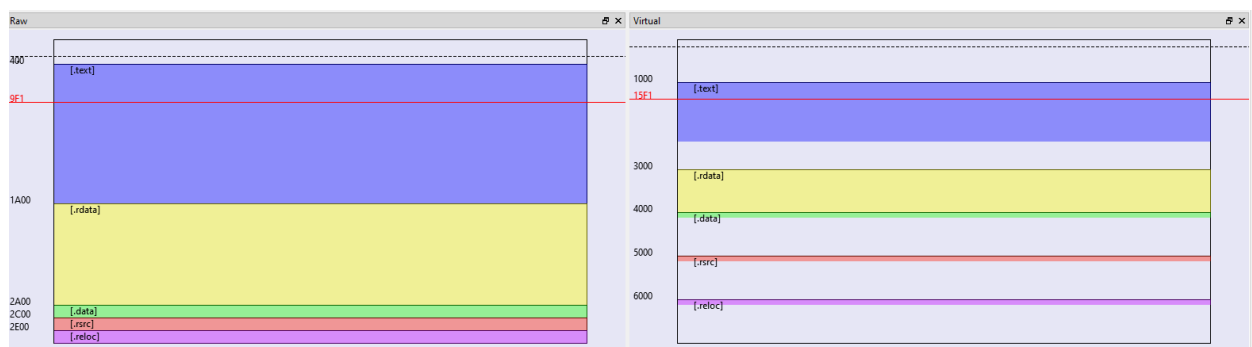
I need to go deeper in order to understand what these strings are doing exactly, I'll be using PE-BEAR (PE-bear is a lightweight tool for inspecting PE files, offering a detailed view of PE headers, sections, imports, exports, and more.)

Here I can see that its an MZ file which means its a windows portable executable file, and we can see a little message "This program cannot be run in DOS mode



You can compare virtual data with raw data value too, if the virtual size is much higher than the raw data size, it means the actual bite sizes are much lower than what it means when it runs.

Which means there's more to this binary than what is initially available to us aka a packed binary



Next I'll take a look at the URLDownloadToFileW

	Offset	Type	Length	
23	1f1c	A	97	C:\Users\Matt\source\repos'
64	24e0	A	18	URLDownloadToFileW
65	24f4	A	10	urlmon.dll
66	2502	A	16	InternetOpenUrlW

## URLDownloadToFile function

Article • 07/13/2016

### In this article

- [Syntax](#)
- [Parameters](#)
- [Return value](#)
- [Remarks](#)
- [Requirements](#)

Downloads bits from the Internet and saves them to a file.

We can see it downloads bits from the internet and saves them to a file & the ShellExecuteW performs an operation on a specific file

	Offset	Type	Length	
57	2466	A	13	ShellExecuteW
58	2474	A	11	SHELL32.dll

## ShellExecuteW function (shellapi.h)

Article • 02/09/2023

### In this article

[Syntax](#)

[Parameters](#)

[Return value](#)

[Remarks](#)

[Show 2 more](#)

Performs an operation on a specified file.

ShellExecuteW is an windows API function

1b40	W	59	cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q "%s"
1bb8	W	57	http://ssl-6582datamanager.helpdeskbro.local/favicon.ico
1c30	W	42	C:\Users\Public\Documents\CR433101.dat.exe
1c88	W	11	Mozilla/5.0
1ca0	W	21	http://huskyhacks.dev
1cd0	W	76	ping 1.1.1.1 -n 1 -w 3000 > Nul & C:\Users\Public\Documents\CR433101.dat.exe
1f04	A	5	RSDSL
1f1c	A	97	C:\Users\Matt\source\repos\HuskyHacks\PMAT-maldev\src\DownloadFromURL\Release\DownloadFromURL.pdb

I'll check the following command lines using chatGPT and try to get a better picture

Command: `cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f/q "%s"`

- `cmd.exe /C`: This command runs a command prompt and then closes it after executing the command that follows.
- `ping 1.1.1.1 -n 1 -w 3000`: This command sends a single ping (`-n 1`) to the IP address `1.1.1.1` and waits for 3000 milliseconds (3 seconds) for a response (`-w 3000`). The IP `1.1.1.1` is commonly used as a placeholder for testing network connectivity.
- `> Nul`: This redirects the output of the ping command to `Nul`, effectively discarding any output from the command.
- `& Del /f/q "%s"`: After the ping command completes, this part attempts to delete a file specified by the placeholder `"%s"`. The `/f` flag forces the deletion, and `/q` makes it "quiet," meaning no confirmation prompts.

**Suspicious Behavior:** This sequence can be used as part of a script to delay execution by 3 seconds (ping as a delay mechanism) and then delete a specified file. The command might be part of malware to clean up or hide traces after execution.

Command: `ping 1.1.1.1 -n 1 -w 3000 > Nul & C:\Users\Public\Documents\CR433101.dat.exe`

- Same ping command as above: It creates a 3-second delay.
- `C:\Users\Public\Documents\CR433101.dat.exe`: This part attempts to execute the file located at `C:\Users\Public\Documents\CR433101.dat.exe` after the delay.

**Suspicious Behavior:** Using a ping command to delay execution and then running an executable file is often seen in malicious scripts. It may indicate an attempt to run malware after a short delay.

## Other Elements

### 3. `http://ssl-6582datamanager.helpdeskbro.s.local/favicon.ico`:

- **Favicon.ico**: Typically, this is a small icon associated with a website. However, it could be part of a URL that delivers a payload or initiates some other form of unwanted behavior, especially if the site is malicious.

**Suspicious Behavior**: If this URL is linked to a command, it could be trying to download or connect to a malicious server. The `.local` domain suggests it's a local network address, which might indicate an attempt to avoid detection by external security tools.

### 4. `C:\Users\Public\Documents\CR433101.dat.exe`:

- **Executable File**: The `.exe` file in a public directory is likely designed to be accessible by any user on the system. This file might be part of a larger malicious operation.

**Suspicious Behavior**: The presence of an executable with a seemingly random name in a public directory is highly suspicious and could indicate malware.

## Summary

- **Commands**: The commands involve creating a **delay**, **deleting files**, and executing potentially malicious software. The use of **ping for delays and Del for file deletion is a common technique in malicious scripts**.
- **URLs and Files**: The URLs and executable files mentioned could be related to malicious activities, such as downloading additional payloads or executing harmful programs.
- **Overall Suspicious Behavior**: The sequence of commands and elements **suggests an attempt to execute and then delete traces of malware, likely with the intention to remain undetected**.

Next I'll also use Xorsearch in order to find specific encoded strings, and were able to find the following

```
C:\Users\Admin
λ Xorsearch C:\Users\Admin\Desktop\Malware.Unknown.exe.malz HTTP

C:\Users\Admin
λ Xorsearch C:\Users\Admin\Desktop\Malware.Unknown.exe.malz http

C:\Users\Admin
λ Xorsearch C:\Users\Admin\Desktop\Malware.Unknown.exe.malz This
Found XOR 00 position 004E: This program cannot be run in DOS mode....$

C:\Users\Admin
λ Xorsearch C:\Users\Admin\Desktop\Malware.Unknown.exe.malz Create
Found XOR 00 position 2446: CreateProcessW

C:\Users\Admin
λ Xorsearch C:\Users\Admin\Desktop\Malware.Unknown.exe.malz URL
Found XOR 00 position 1F5E: URL\Release\DownloadFromURL.pdb
Found XOR 00 position 1F76: URL.pdb
Found XOR 00 position 24E0: URLDownloadToFileW
Found XOR 20 position 24F4: URLMON.DLL . INTERNETOPENURLw . INTERNETOPENw wi
Found ADD E0 position 24F4: URLMON.DLL...)INTERNET/PEN5RL7...)INTERNET/PEN7.7)
Found ADD E6 position 2764: URLVEY..P.ZKXSOTGZK.GV0.SY.]0T.IXZ.YZJOU.R.....JR
```

Next I'll check if the malware is packed using PESTUDIO, packing is a technique where malware engineers could compress or encrypt the data

I'm going to look for other interesting information as well

PESTUDIO is a great tool that also provides threat lvl indicators and has MITRE ATT&CK integrated

indicator (30)	detail	level
groups > API	dynamic-library   execution   reconnaissance   file   synchronization   exception   network   memory	+++++
libraries > flag	OLE32 Extensions for Win32 (urlmon.dll)	+++++
libraries > flag	Internet Extensions for Win32 Library (WININET.dll)	+++++
mitre > technique	T1106   T1057   T1124   T1082	+++++
string > URL	http://ssl-6582datamanager.helpdeskbro.local/favicon.ico	++
string > URL	http://huskyhacks.dev	++
imports > flag	9	++

Here we can see 9 flagged strings, we can see execution, shellexe, network internet OpenURL ect

flag (9)	label (67)	group (8)	technique (4)	value
x	import	reconnaissance	T1057   Process Discovery	GetCurrentProcessId
x	import	network	-	URLDownloadToFile
x	import	network	-	InternetOpenUrl
x	import	network	-	InternetOpen
x	import	execution	T1106   Execution through API	CreateProcess
x	import	execution	T1106   Execution through API	ShellExecute
x	import	execution	T1057   Process Discovery	GetCurrentProcess
x	import	execution	-	TerminateProcess
x	import	execution	T1057   Process Discovery	GetCurrentThreadId

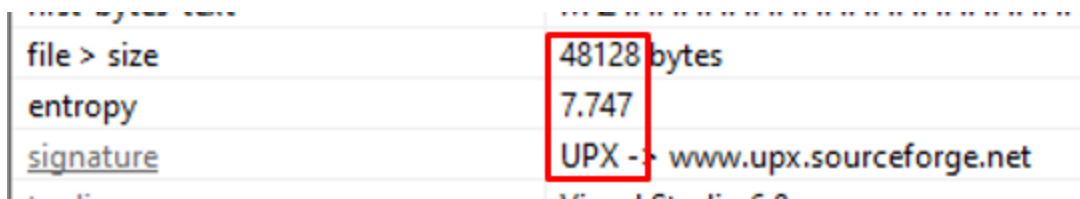
With an entropy of 5.719 the file doesn't appear to heavily compressed

file > size	12288 bytes
entropy	5.719



Now I'll be comparing a packed vs unpacked Malware to learn better how to detect it

Testing packed malware in PEStudio we can see that the entropy score 7.747 large byte size and UPX signature have all the indicators of a packed file,



A screenshot of the PEStudio application's 'File Properties' dialog box, specifically the 'Summary' tab. The table lists three indicators: 'file > size' with a value of '48128 bytes', 'entropy' with a value of '7.747', and 'signature' with a value of 'UPX -> www.upx.sourceforge.net'. The values '48128 bytes' and '7.747' are highlighted with a red rectangular box, indicating they are key indicators of a packed file.

file > size	48128 bytes
entropy	7.747
signature	UPX -> www.upx.sourceforge.net

also, there are many human-readable and the format is very easy to understand in the unpacked version compared to a packed version of the same malware.

Another difference is a small import table compared to a large import table, up next dynamic analysis!

# Dynamic Malware Analysis

Things I'll be looking for in a dynamic analysis

- Process activity
- Network activity
- Registry activity (persistence)
- File activities (persistence)

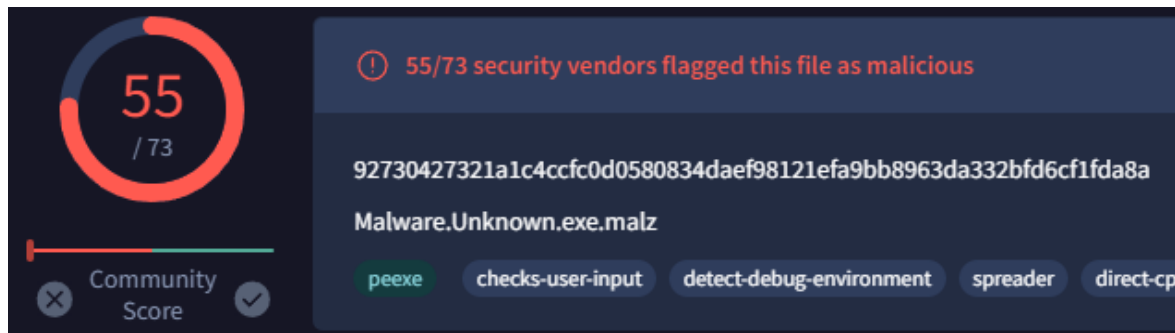
Tools I'll use:

- Process Hacker: Monitors and manages processes and services.
- Procmon: Tracks real-time file, registry, and process activity.
- Wireshark: Analyzes network traffic.
- Regshot: Compares registry snapshots to detect changes.

But first I'll begin with a simple VirusTotal static analyst to get an idea of what I'm dealing with here before I detonate it

I'll get the SHA256 & check VirusTotal

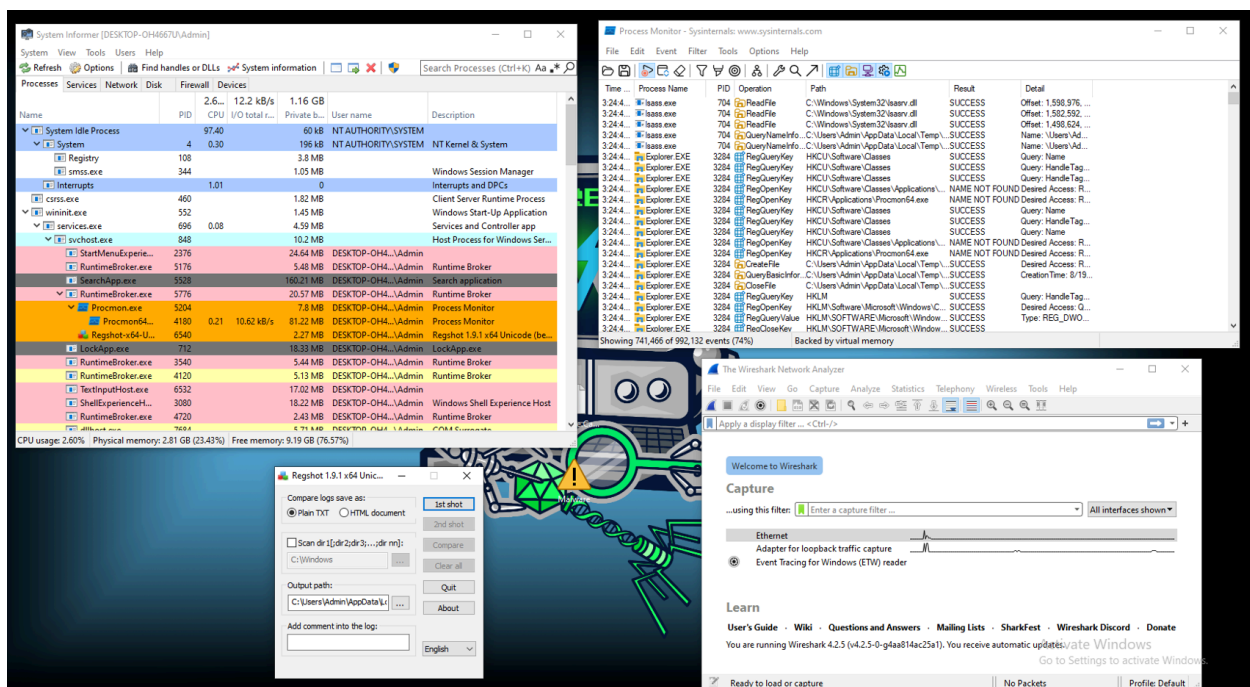
```
C:\Users\Admin\Desktop  
λ sha256sum.exe Malware.unknown.exe.malz  
92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a
```



MITRE ATT&CK Tactics and Techniques	
+ Execution	TA0002
+ Privilege Escalation	TA0004
+ Defense Evasion	TA0005
+ Credential Access	TA0006
+ Discovery	TA0007
+ Collection	TA0009
+ Command and Control	TA0011

Before I detonate, I'll make sure I'll have these 4 tools active

- Process monitor
- Wireshark
- Regshot
- System Informer



Even though the malware hid itself from the live services almost instantly, using the Procmon process tree allowed me to capture his actions regardless of his disguise attempts.

Here we can see the malware running malicious commands, trying to get outside access aswell.

Process Tree

☐ Only show processes still running at end of current trace  
☒ Timelines cover displayed events only

Process	Company	Owner	Command
conhost.exe (2632)	Microsoft Corporation	DESKTOP-OH46...	\\??C:\Windows\system32\conhost.exe 0x4
Regshot-x64-Unicode.exe (562)	Regshot Team	DESKTOP-OH46...	"C:\Tools\Regshot-x64-Unicode\Regshot-x64-Unicode.exe"
SystemInformer.exe (8036)	System Informer	DESKTOP-OH46...	"C:\Tools\SystemInformer\amd64\SystemInformer.exe"
7zFM.exe (7452)	Pavlov	DESKTOP-OH46...	"C:\Program Files\7-Zip\7zFM.exe" "C:\Users\Admin\Desktop\Malw...
Procmon.exe (1772)	Internals - ww...	DESKTOP-OH46...	"C:\Tools\sysinternals\Procmon.exe"
Procmon64.exe (7384)	Internals - ww...	DESKTOP-OH46...	"C:\Users\Admin\AppData\Local\Temp\Procmon64.exe" /originalp...
Malware.Unknown.exe (2684)		DESKTOP-OH46...	"C:\Users\Admin\Desktop\Malware.Unknown.exe"
Conhost.exe (7340)	Microsoft Corporation	DESKTOP-OH46...	\\??C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
cmd.exe (8056)	Microsoft Corporation	DESKTOP-OH46...	cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q "C:\Users\Ad...
Conhost.exe (2872)	Microsoft Corporation	DESKTOP-OH46...	\\??C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
PING.EXE (6272)	Microsoft Corporation	DESKTOP-OH46...	ping 1.1.1.1 -n 1 -w 3000
SearchIndexer.exe (5548)	Microsoft Corporation	NT AUTHORITY\...	C:\Windows\system32\SearchIndexer.exe /Embedding
SearchProtocolHost.exe (5080)	Microsoft Corporation	NT AUTHORITY\...	"C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltP...
SearchFilterHost.exe (720)	Microsoft Corporation	NT AUTHORITY\...	"C:\Windows\system32\SearchFilterHost.exe" 0 808 812 820 8192 ...
Idle (0)			
System (4)		NT AUTHORITY\...	
Registry (108)		NT AUTHORITY\...	
smss.exe (344)	Microsoft Corporation	NT AUTHORITY\...	\SystemRoot\System32\smss.exe

Description: Console Window Host  
Company: Microsoft Corporation  
Path: C:\Windows\System32\Conhost.exe  
Command: \\??C:\Windows\system32\conhost.exe 0xffffffff -ForceV1  
User: DESKTOP-OH4667U\Admin  
PID: 7340 Started: 8/19/2024 4:19:42 AM  
Exited: 8/19/2024 4:19:44 AM

Go To Event Include Process Include Subtree Close

# Dynamic Malware Analysis With telemetry aka a “fake internet” active

Now I'll do a more interesting investigation, for this I will need to install Remnux and configure INetSim for me to be able to use Wireshark and for the Malware to act realistically

I'll start configuring the Remnux & The Flare VM Network now so I'll be able to safely detonate the malware while analyzing it now

Adapter DHCP Server

☐ Configure Adapter Automatically

☒ Configure Adapter Manually

IPv4 Address: 10.0.0.1

IPv4 Network Mask: 255.255.255.0

IPv6 Address: fe80::2e88:ff7d:bfed:9d88

IPv6 Prefix Length: 64

Apply Reset

Adapter DHCP Server

☒ Enable Server

Server Address: 10.10.10.2

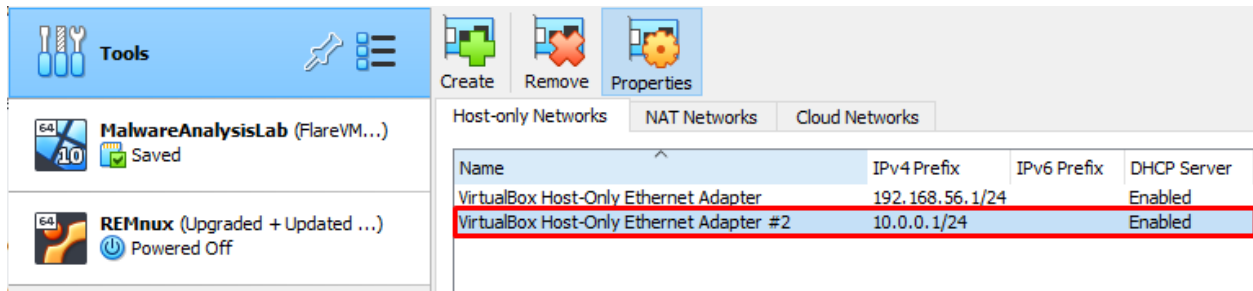
Server Mask: 255.255.255.0

Lower Address Bound: 10.10.10.3

Upper Address Bound: 10.10.10.254

Apply Reset

Using best practice and changing the IP address to something completely different from the rest or usual 192.168 octets



To make sure, I'll actively check both hosts network settings

```
remnux@remnux:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
n 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
efault qlen 1000
    link/ether 08:00:27:ee:12:e5 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.3/24 brd 10.10.10.255 scope global dynamic
        valid_lft 584sec preferred_lft 584sec
    inet6 fe80::a00:27ff:feee:12e5/64 scope link
        valid_lft forever preferred_lft forever
remnux@remnux:~$ ping 10.10.10.4
PING 10.10.10.4 (10.10.10.4) 56(84) bytes of data.
64 bytes from 10.10.10.4: icmp_seq=1 ttl=128 time=0.370 ms
64 bytes from 10.10.10.4: icmp_seq=2 ttl=128 time=0.464 ms
64 bytes from 10.10.10.4: icmp_seq=3 ttl=128 time=0.450 ms
64 bytes from 10.10.10.4: icmp_seq=4 ttl=128 time=0.506 ms
```

```
Cmder
C:\Users\Admin
λ ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . : 
    Link-local IPv6 Address . . . . . : fe80::d522:f485:b498:9709%5
    IPv4 Address. . . . . : 10.10.10.4
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

C:\Users\Admin
λ ping 10.10.10.3

Pinging 10.10.10.3 with 32 bytes of data:
Reply from 10.10.10.3: bytes=32 time<1ms TTL=64
Reply from 10.10.10.3: bytes=32 time<1ms TTL=64
Reply from 10.10.10.3: bytes=32 time<1ms TTL=64
Reply from 10.10.10.3: bytes=32 time<1ms TTL=64

Ping statistics for 10.10.10.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\Admin
λ ping google.com
Ping request could not find host google.com. Please check the name and IP address.

C:\Users\Admin
λ ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
remnux@remnux: ~
remnux@remnux:~$ ping google.com
ping: google.com: Temporary failure in name resolution
remnux@remnux:~$ ping 8.8.8.8
ping: connect: Network is unreachable
remnux@remnux:~$
```

Great now I have two hosts that can talk with each other but are blocked from accessing the internet.

It's ready to go, I'll be safe detonating now.

Now I'll start setting up the INetSim, which will enable me to have a "fake internet" for the malware to go to, which will allow me to study the Host Based indicators and the Network based indicators with tools like Wireshark



Now I'm configuring the inetsim.config file

```
remnux@remnux:~$ sudo nano /etc/inetsim/inetsim.conf
```

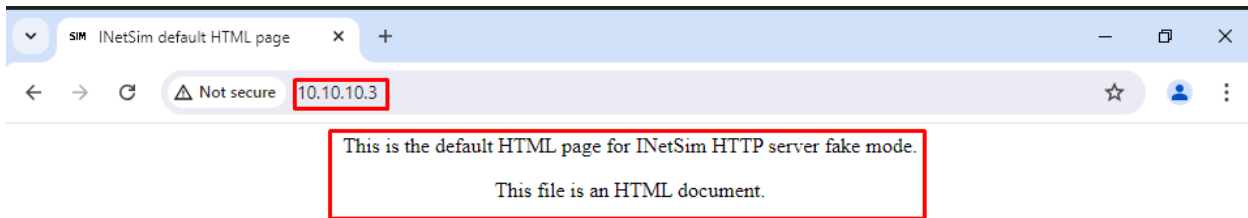
I'm going to:

- Start service DNS
- Change DNS default IP to 10.10.10.3 (Remnux)
- Change server bind address to 0.0.0.0 (will bind to all interfaces on the host)

We can see that now the DNS service is also active now at default 53 port

```
remnux@remnux:~$ inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 1779) ===
Session ID:      1779
Listening on:    10.10.10.3
Real Date/Time:  2024-08-19 12:55:33
Fake Date/Time: 2024-08-19 12:55:33 (Delta: 0 seconds)
Forking services...
* dns_53_tcp_udp - started (PID 1783)
* pop3_110_tcp - started (PID 1788)
* smtp_25_tcp - started (PID 1786)
* smtps_465_tcp - started (PID 1787)
* ftp_21_tcp - started (PID 1790)
* ftps_990_tcp - started (PID 1791)
* pop3s_995_tcp - started (PID 1789)
* http_80_tcp - started (PID 1784)
* https_443_tcp - started (PID 1785)
done.
Simulation running.
```

FlareVM Using 10.10.10.3 which is our Remnux server actually gives us a server fake mode on both HTTP & HTTPS, pretty cool!



Lastly I will change the FlareFM default DNS to 10.10.10.3 which is the Remnux, Now I have a functional “fake” dns server and a “fake internet” to do some dynamic malware analysis

**Now I can safely start with the Dynamic Analysis!**

I'll be on the lookout for both host and network indicators, for example like a host deleting files or installing C&C persistence, or the network calls to a domain or downloads a file.

If I can get a good idea of what's going on both the host and network during the attack or detonation, that would be great information for the later stages of dynamic analysis

Before detonating the Malware, I'll activate Wireshark & start recording all networking traffic.

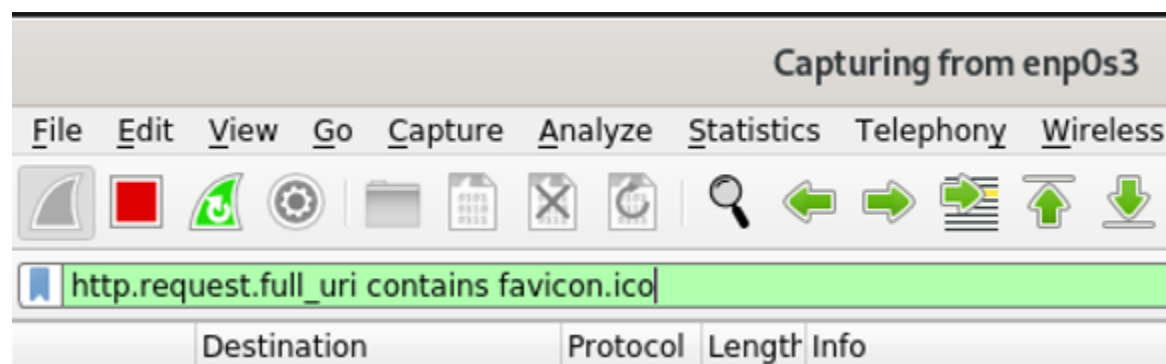
I'll test if it's working by going to google, it

10.10.10.4	10.10.10.3	TLSv1.3	1874 Client Hello (SNI=www.google.com)
10.10.10.3	10.10.10.4	TCP	60 443 → 49990 [ACK] Seq=1 Ack=1821 Win=63488 Len
10.10.10.4	10.10.10.3	TLSv1.3	1849 Client Hello (SNI=update.googleapis.com)
10.10.10.3	10.10.10.4	TCP	60 443 → 49991 [ACK] Seq=1 Ack=1796 Win=63488 Len
10.10.10.3	10.10.10.4	TLSv1.3	1509 Server Hello, Change Cipher Spec, Application
10.10.10.4	10.10.10.3	TLSv1.3	84 Change Cipher Spec, Application Data

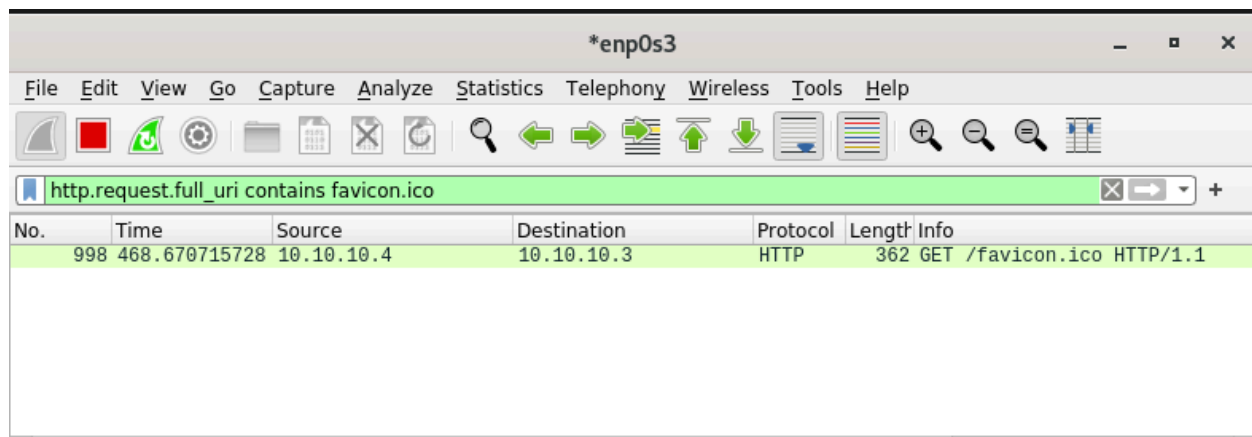


First, I'll do a dynamic analysis of the same malware which we did static (Malware.unknown.exe)

I could use the URL request found in the static analysis and have the remnux wireshark active to scan for interactions while I run it



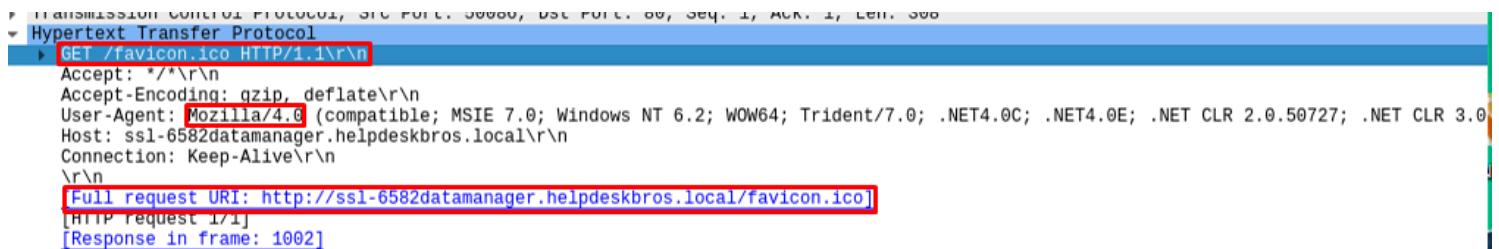
I'll detonate the Malware.Unknown file



The image shows a Wireshark packet capture window titled '\*enp0s3'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. A filter bar at the top shows 'http.request.full\_uri contains favicon.ico'. The packet list table below shows a single entry:

No.	Time	Source	Destination	Protocol	Length	Info
998	468.670715728	10.10.10.4	10.10.10.3	HTTP	362	GET /favicon.ico HTTP/1.1

And we can see the live log getting fed, I'll investigate this log further on Wireshark



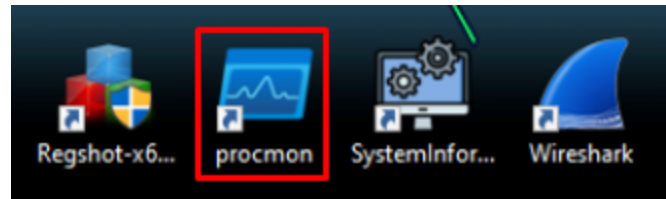
The image shows the packet details view for the selected packet. The 'Hypertext Transfer Protocol' section is expanded, showing the following fields:

- GET /favicon.ico HTTP/1.1
- Accept: \*/\*
- Accept-Encoding: gzip, deflate
- User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0)
- Host: ssl-6582datamanager.helpdeskbro.s.local
- Connection: Keep-Alive
- Full request URI: http://ssl-6582datamanager.helpdeskbro.s.local/favicon.ico
- [HTTP request 1/1]
- [Response in frame: 1002]

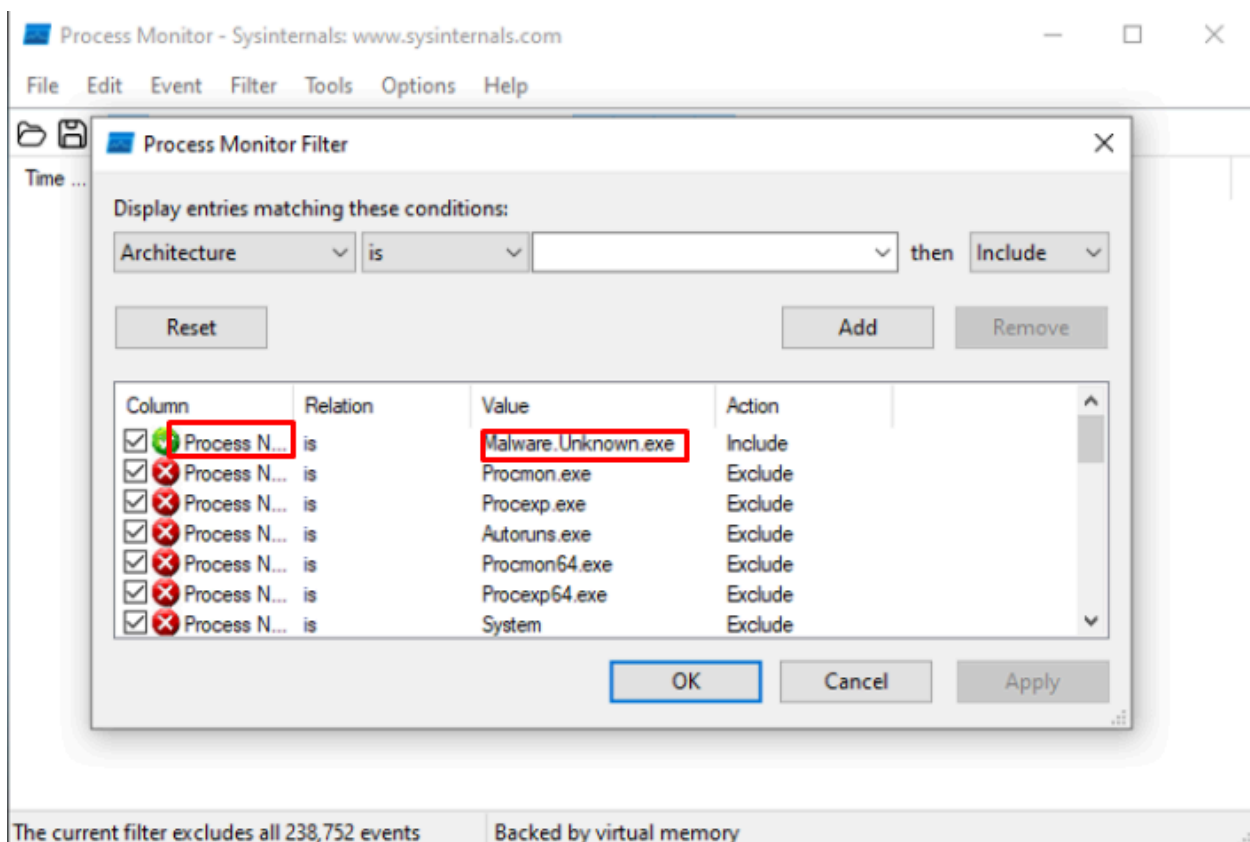
We have a GET request using Mozilla, looking at the full request URI we can see that it correlates with the static analysis and is trying to get favicon.ico

This is a good Network indicator and could be used in the future as an IPBlock list for example

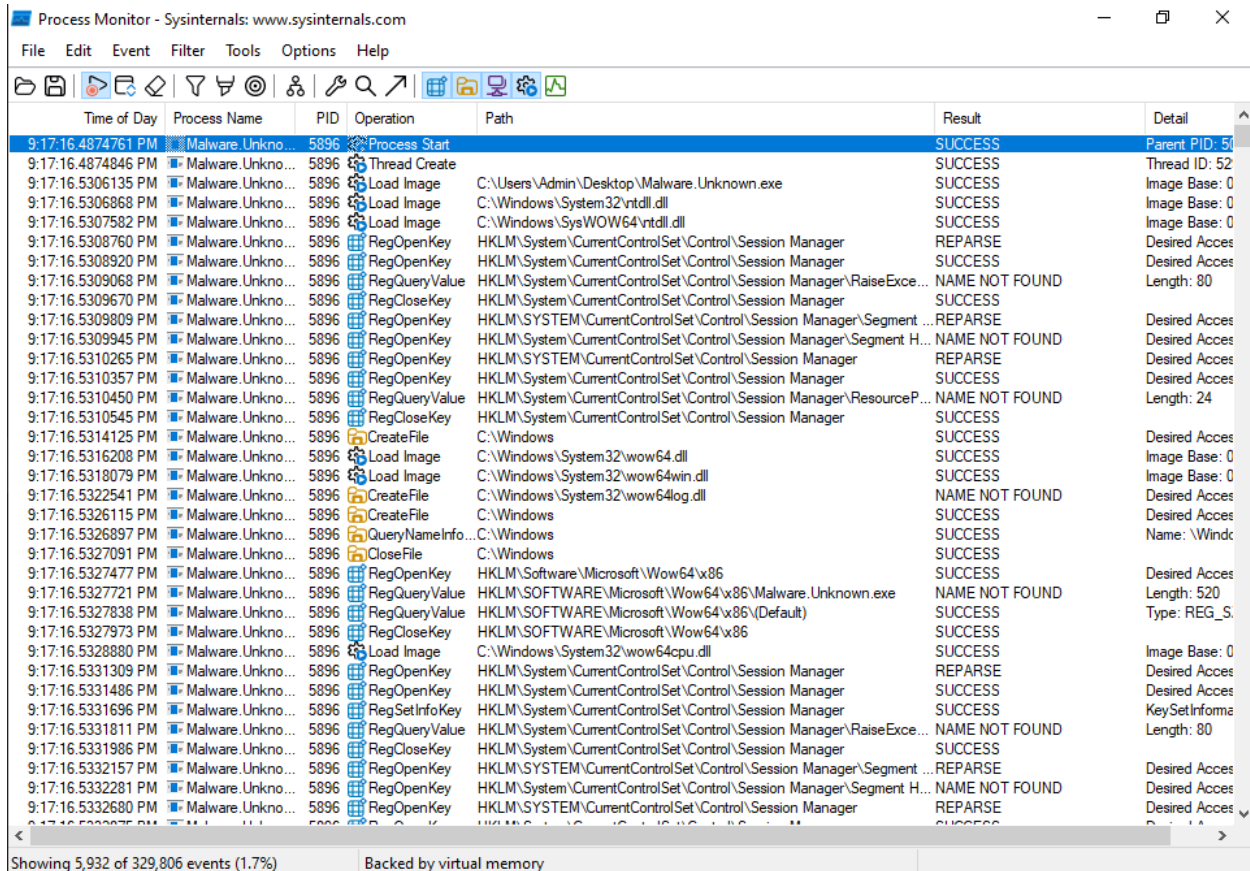
I'll reset to last snapshot, and do more investigations, this time for host based indicators, starting Procmon



I'll filter in the process name for the specific detonation



Alot of results after detonation, I'll dig deeper



Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path	Result	Detail
9:17:16.4874761 PM	Malware.Unkno...	5896	Process Start		SUCCESS	Parent PID: 52
9:17:16.4874846 PM	Malware.Unkno...	5896	Thread Create		SUCCESS	Thread ID: 52
9:17:16.5306135 PM	Malware.Unkno...	5896	Load Image	C:\Users\Admin\Desktop\Malware.Unknown.exe	SUCCESS	Image Base: 0
9:17:16.5306868 PM	Malware.Unkno...	5896	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0
9:17:16.5307582 PM	Malware.Unkno...	5896	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Image Base: 0
9:17:16.5308760 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	REPARSE	Desired Acces
9:17:16.5308920 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Acces
9:17:16.5309068 PM	Malware.Unkno...	5896	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\RaiseExce...	NAME NOT FOUND	Length: 80
9:17:16.5309670 PM	Malware.Unkno...	5896	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
9:17:16.5309809 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Segment ...	REPARSE	Desired Acces
9:17:16.5309945 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager\Segment H...	NAME NOT FOUND	Desired Acces
9:17:16.5310265 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Acces
9:17:16.5310357 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Acces
9:17:16.5310450 PM	Malware.Unkno...	5896	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\ResourceP...	NAME NOT FOUND	Length: 24
9:17:16.5310545 PM	Malware.Unkno...	5896	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
9:17:16.5314125 PM	Malware.Unkno...	5896	CreateFile	C:\Windows	SUCCESS	Desired Acces
9:17:16.5316208 PM	Malware.Unkno...	5896	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0
9:17:16.5318079 PM	Malware.Unkno...	5896	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS	Image Base: 0
9:17:16.5322541 PM	Malware.Unkno...	5896	CreateFile	C:\Windows\System32\wow64log.dll	NAME NOT FOUND	Desired Acces
9:17:16.5326115 PM	Malware.Unkno...	5896	CreateFile	C:\Windows	SUCCESS	Desired Acces
9:17:16.5326897 PM	Malware.Unkno...	5896	QueryNameInfo	C:\Windows	SUCCESS	Name: \Windc
9:17:16.5327091 PM	Malware.Unkno...	5896	CloseFile	C:\Windows	SUCCESS	
9:17:16.5327477 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\Software\Microsoft\Wow64\86	SUCCESS	Desired Acces
9:17:16.5327721 PM	Malware.Unkno...	5896	RegQueryValue	HKLM\SOFTWARE\Microsoft\Wow64\86\Malware.Unknown.exe	NAME NOT FOUND	Length: 520
9:17:16.5327838 PM	Malware.Unkno...	5896	RegQueryValue	HKLM\SOFTWARE\Microsoft\Wow64\86\Default	SUCCESS	Type: REG_S
9:17:16.5327973 PM	Malware.Unkno...	5896	RegCloseKey	HKLM\SOFTWARE\Microsoft\Wow64\86	SUCCESS	
9:17:16.5328880 PM	Malware.Unkno...	5896	Load Image	C:\Windows\System32\wow64cpu.dll	SUCCESS	Image Base: 0
9:17:16.5331309 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	REPARSE	Desired Acces
9:17:16.5331486 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Acces
9:17:16.5331696 PM	Malware.Unkno...	5896	RegSetInfoKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	KeySetInforma
9:17:16.5331811 PM	Malware.Unkno...	5896	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\RaiseExce...	NAME NOT FOUND	Length: 80
9:17:16.5331986 PM	Malware.Unkno...	5896	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
9:17:16.5332157 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Segment ...	REPARSE	Desired Acces
9:17:16.5332281 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager\Segment H...	NAME NOT FOUND	Desired Acces
9:17:16.5332680 PM	Malware.Unkno...	5896	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Acces

Showing 5,932 of 329,806 events (1.7%) Backed by virtual memory

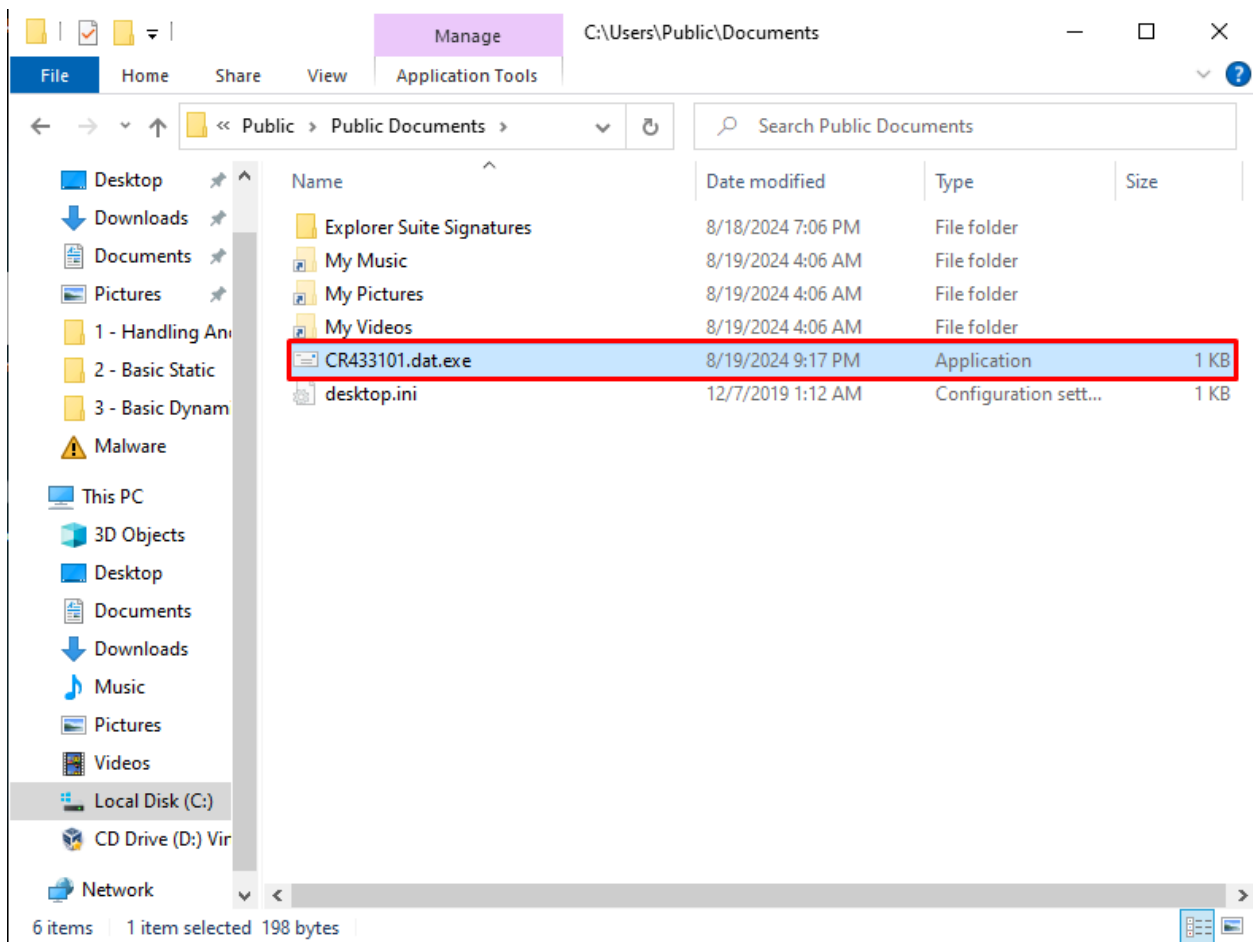
I'll filter for files, we can see that the Malware is creating changing and loading a lot of the DLL files when it runs

We can see the malware creating the correlated file from the static investigation

9:17:18.9260260 PM	M..	5896	CreateFile	C:\Users\Public\Documents\CR433101.dat.exe
9:17:18.9262201 PM	M..	5896	ReadFile	C:\Users\Admin\AppData\Local\Microsoft\Window
9:17:18.9262397 PM	M..	5896	ReadFile	C:\Users\Admin\AppData\Local\Microsoft\Window
9:17:18.9262497 PM	M..	5896	WriteFile	C:\Users\Public\Documents\CR433101.dat.exe
9:17:18.9263198 PM	M..	5896	CloseFile	C:\Users\Public\Documents\CR433101.dat.exe

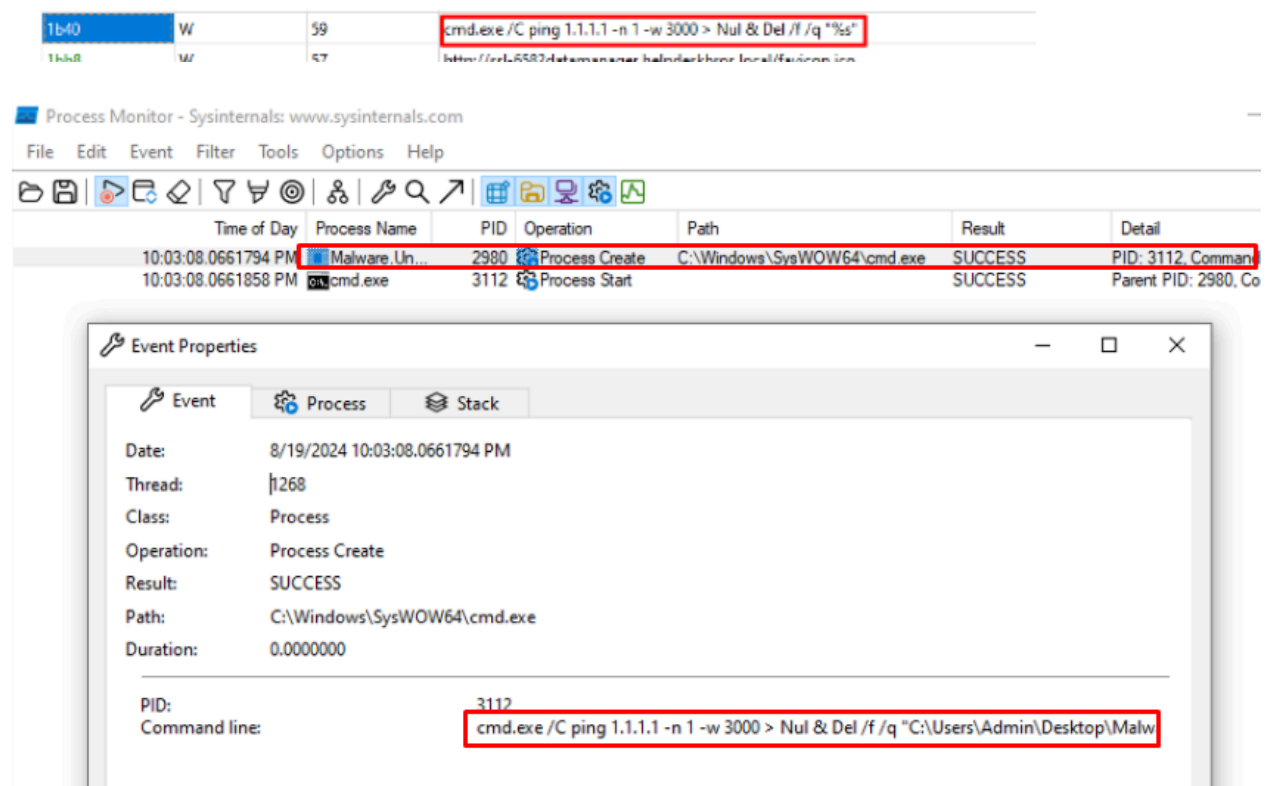
```
C:\Windows\SysWOW64\clbcatq.dll
C:\Users\Admin\Desktop\ping 1.1.1.1 -n 1 -w 3000 > nul & C:\Users\Public\Documents\CR433101.dat.exe
C:\Windows
C:\Users\Admin\Desktop
C:\Windows\System32\en-US\mshwsock.dll.mui
C:\Users\Admin\AppData\Local\Microsoft\Windows\NetCache\IE\0A9G05V\1\EXOF4F.htm
```

We can see the file the malware created as well, this is a great host indicator



We can hypothesize it downloads a file from a remote address, looking to infect a system from a second stage payload from favicon.ico

I'll look for more host based indicators by filtering for this on procmon



Great we got the log, this is the mechanism of the self deletion of this binary, another good host indicator

UnknownMalware.exe =

Now we know: Dropper.DownloadFromURL.exe

URL Exists > Download favicon.ico > Write to Disk (CR433101.exe) > Run favicon.ico (CR433101.exe)

URL Doesn't Exist > Delete from disk > do not Run



# Summary of Live Telemetry Lab for Malware Analysis

I have successfully developed a live telemetry lab tailored for both static and dynamic malware analysis, leveraging the powerful combination of **Remnux** and **FlareVM**. This environment allows for an in-depth understanding of malware behavior within a secure and controlled setting.

## Tools and Techniques Employed:

- **Wireshark**: Used for detailed network traffic analysis, enabling precise filtering and querying to isolate specific logs related to malware activities.
- **INetSim**: Deployed to create a simulated internet environment, this tool emulates network services, allowing you to observe how malware attempts to interact with the internet. This setup captures real-time telemetry while preventing any external exposure.
- **Static Analysis**: with **PEBear**, **PEStudio**, and **FLOSS**:
  - **PEBear** and **PEStudio**: Essential for dissecting executable files, these tools provide insights into the malware's structure and potential threats.
  - **FLOSS**: Utilized to extract hidden or obfuscated strings, revealing critical data such as command-and-control (C2) URLs and file paths embedded within the malware.
- **VirusTotal**: Integrated to scan malware samples against a vast database of known threats, aiding in the identification and categorization of malware.
- **Regshot**: Used to capture and compare system registry snapshots before and after malware execution, providing a clear view of any changes made by the malware.
- **ProcMon (Process Monitor)**: Employed for monitoring real-time file system, registry, and process/thread activity. This tool is vital in tracking the actions of malware at a granular level.
- **System Informer**: Utilized for comprehensive system monitoring, offering detailed insights into the processes, threads, and system resources affected by the malware.

### **Safety and Best Practices:**

I ensured the lab's isolation from the external internet, adhering to best practices for malware detonation and minimizing the risk of unintentional spread or communication with real-world threat actors. This approach guarantees a secure analysis environment.

### **Outcomes:**

The lab's configuration enabled the capture of live telemetry data and provided a thorough understanding of the malware's behavior. By combining static and dynamic analysis tools, I extracted critical information that enhances threat detection and response strategies in real-world scenarios.

This setup not only boosts my malware analysis capabilities but also equips me with the knowledge needed to tackle complex threats effectively.

(GPT is pretty good at making summaries, isn't he 😂)

This project was a fun one, time to run some RATs and see how they work in real-time.