Dynamic Malware Analysis

Sample Name: ASKBot.exe
Analysis Type: Behavioral Analysis using Different Tools
Environment: Isolated Virtual Machine (Windows 10, No Internet Access)

PRELIMINARY SETUP :

The analysis was performed in a fully isolated virtual environment to ensure system integrity and avoid unintentional malware propagation. The Windows 10 VM was configured with HOA. Shared folder access, drag-and-drop, and clipboard sharing were disabled. System protections like Windows Defender were turned off to ensure malware behavior could be observed unimpeded. A snapshot of the VM was taken

Table of Contents

1. Introduction

→ Environment Setup

→ Briefly introduces ASKBot.exe

→ objectives of the analysis, and tools used.

2. Task 3.1 – Run and Observe

→Documents dynamic analysis using Procmon,Wireshark etc.

→ monitor real-time behavior of the malware.

3. Task 3.2 – Debugging with x64dbg

→Involves stepping through the binary in x64dbg to uncover internal logic.

→ API calls, and hidden behavior.

1. Introduction

This report presents a detailed dynamic analysis of a suspicious binary file named **ASKBot.exe**, which is believed to demonstrate malicious behavior. The primary goal of this analysis is to investigate how the malware interacts with the operating system, including changes to the file system, registry

The tools employed include **Procmon** for monitoring file and registry event, **Wireshark** for inspecting network traffic, and **x64dbg** for low-level debugging.

1.1 Purpose

The purpose of this report is to conduct a **practical, tool-based investigation** of the ASKBot malware. It aims to **identify any malicious behavior** through observable file system, registry, and network activity. The report will also determine whether the malware uses **evasion techniques**, attempts persistence, or initiates any C2 communications.

2. Task 3.1 – Run and Observe

In this section, the sample will be analyzed using various dynamic analysis tools such as **Procmon**, **Regshot**, and others.

2.1 Regshot Analysis

Regshot is a lightweight tool that lets you take a snapshot of the **Windows registry and file system**,
Regshot taken before running exe file

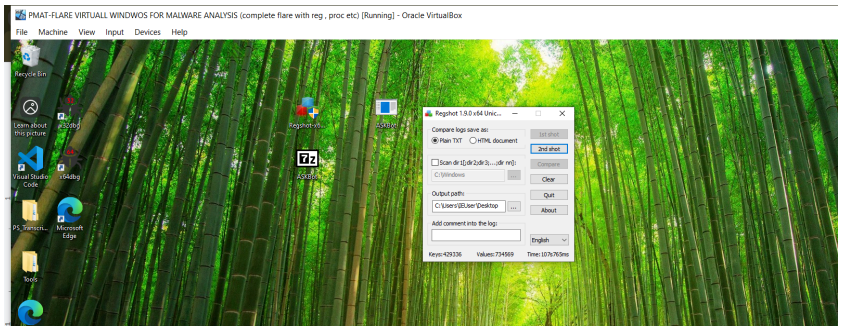

Figure 1

2nd regshot taken after running

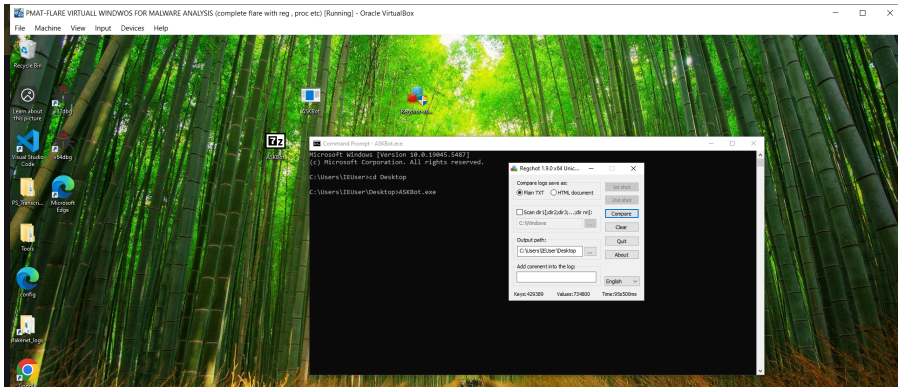

Figure 2

Compare:

Key Changes:So total 351 changes happened

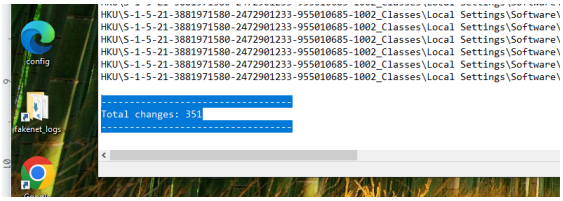

Figure 3

→**Keys deleted:** 2
→**Keys added:** 55
→**Values deleted:** 2

Meaningfull Findings

1:

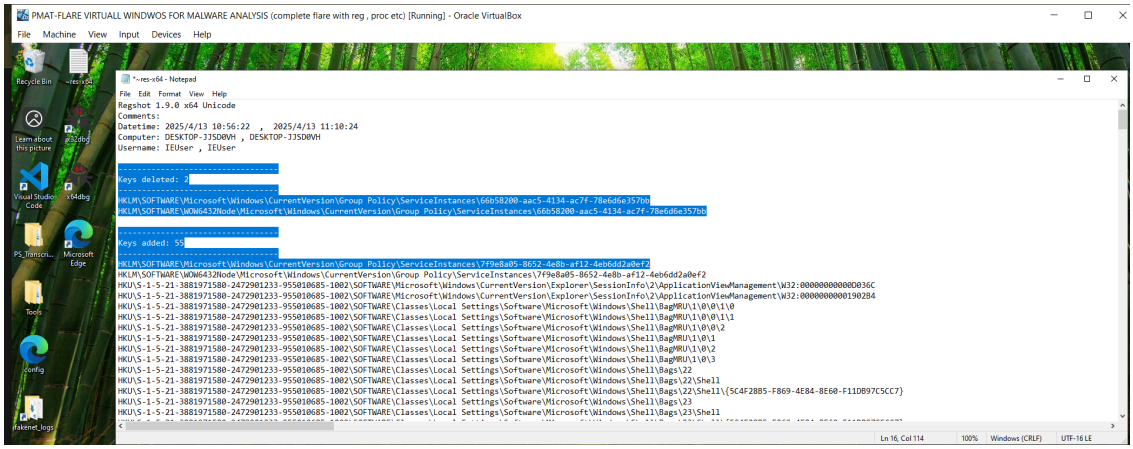

Figure 4

The malware created multiple entries under **Group Policy ServiceInstances in above (Figure 4), it suggesting it may be setting up a form of system-level persistence**.

2:

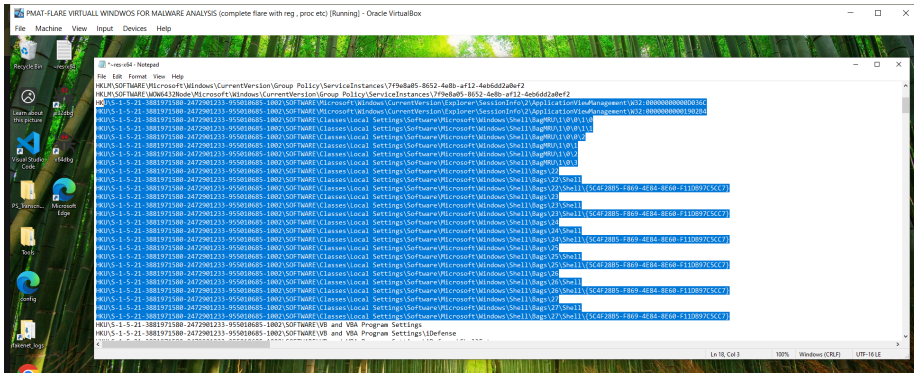

Figure 5

New entries under **`Shell\BagMRU` and `Shell\Bags`** indicate folder view changes, possibly used to track **user activity or disguise malicious file placement**. Such behavior is often linked to stealth tactics in file-based malware

3:

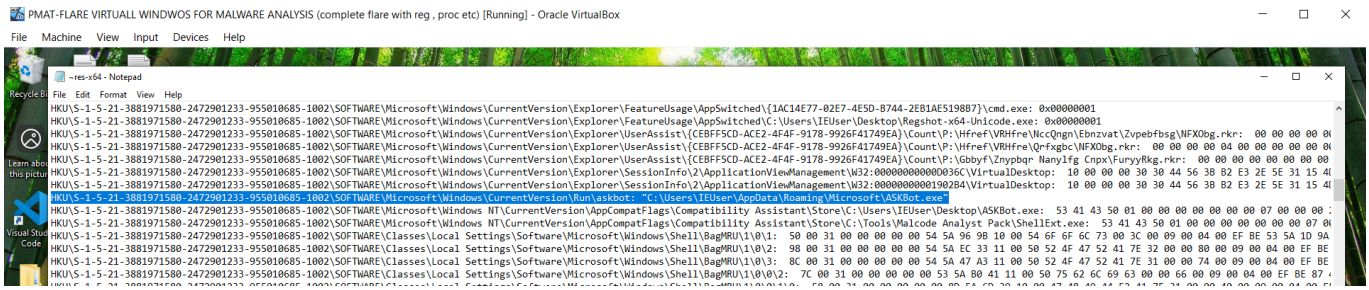

Figure 6

One of the most critical entries:(above Figure 6)

This confirms persistence — the malware is configured to execute automatically at startup for the current user by using the **Run key**.

4:

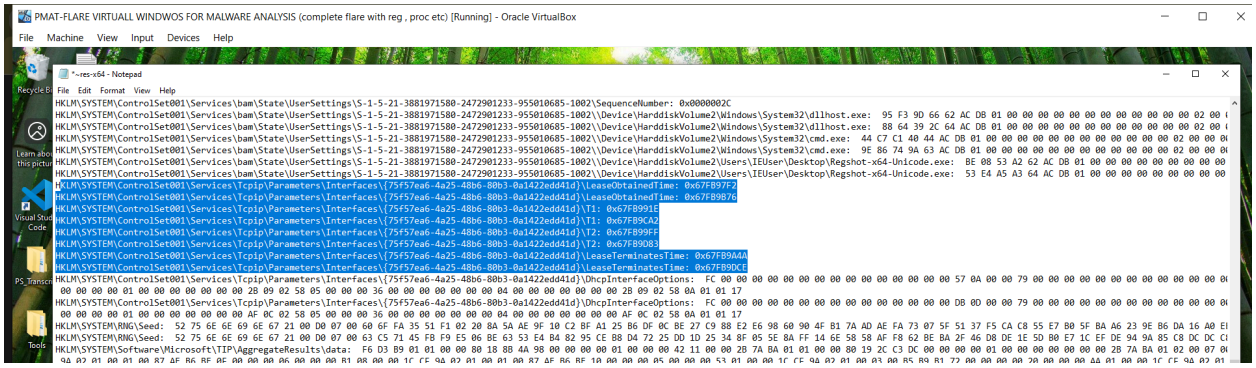

Figure 7

DHCP and lease-related values changed:

- **LeaseObtainedTime, T1, T2, LeaseTerminatesTime**

Implies potential manipulation or monitoring of **network configurations** or **DHCP behavior**,Changes to network-related registry keys (e.g., DHCP settings) might point to attempts to reroute or intercept network traffic.

2.2 wirshark Analysis

Network communication indications through strings

Figure 8

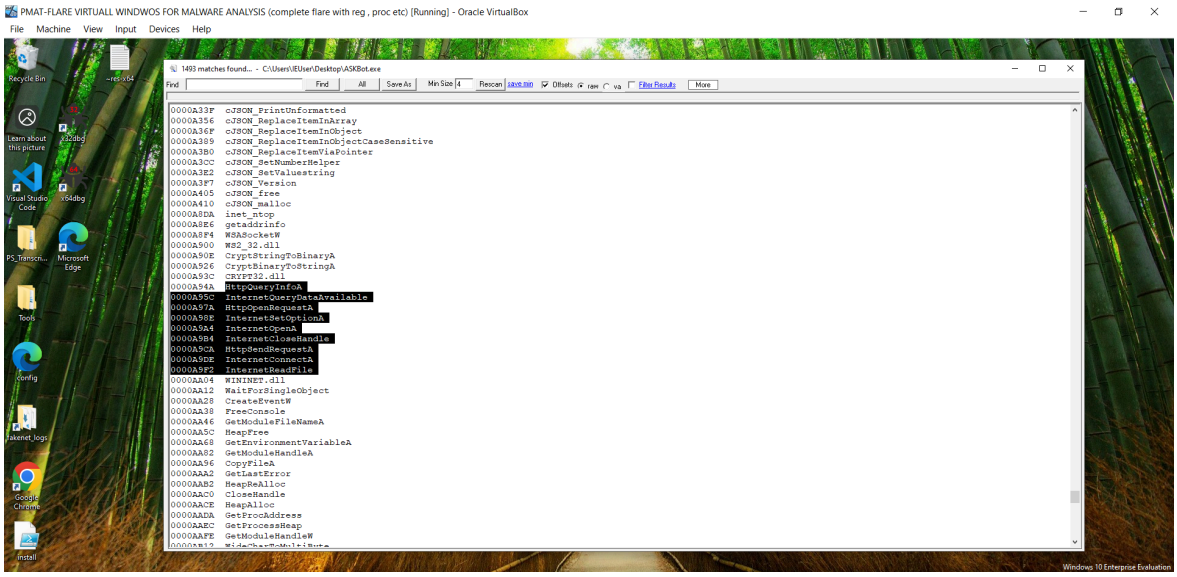

Figure 9

These API calls are used by malware to connect to the internet, send requests, and read responses as you can seen in above **Figure 9**,And Also These show it's using raw HTTP communication etc…

Setup the FakeNet (INETSIM) ON Remnux

Firstly i make the separate adapter for separate connection from the real internet

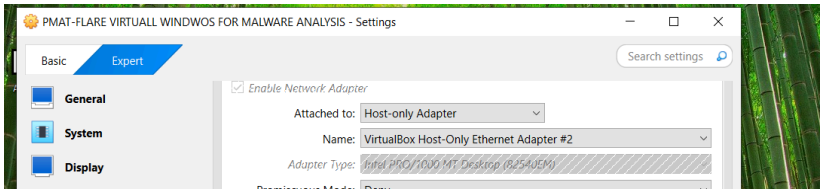

Figure 10

Then paste the address of the remnux in the dns of ipv4 for the fake dns service

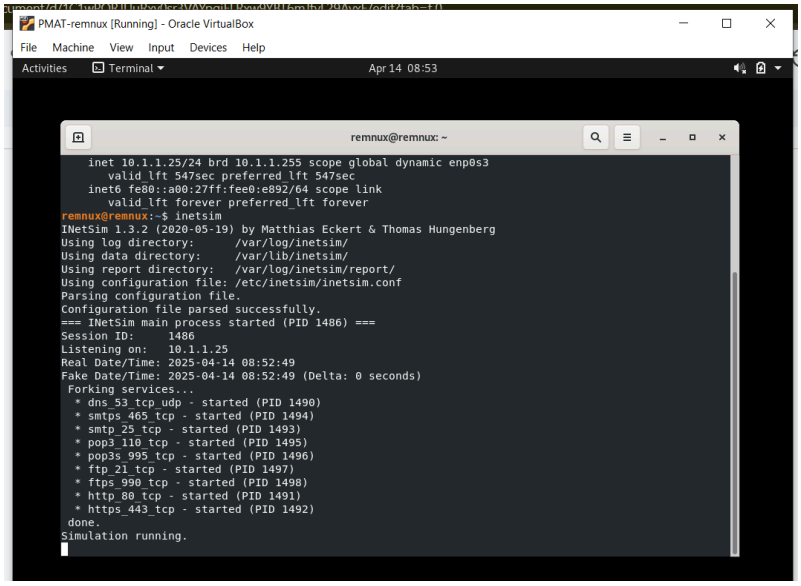

Figure 11

Run the INETSIM

Now everything is done so now open the wireshark then start the **ASKBot.exe** then observe the traffic in the wireshark

capture:

So i see the lot of traffic in wireshark after running the exe file

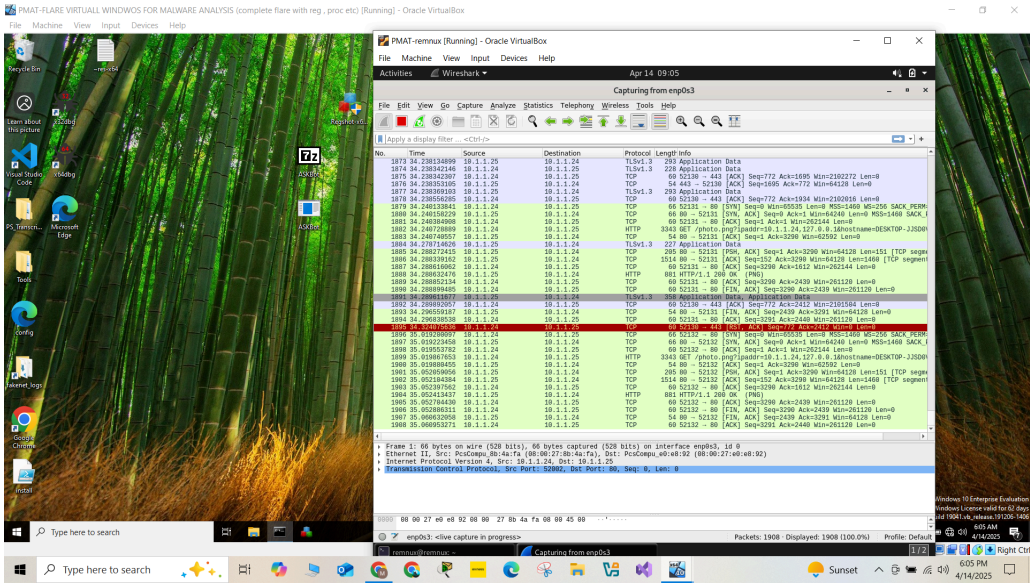

Figure 12

Let analyze the meaningfull traffic in it

First of all i use filter **(http)** because we now that we have the indicator of http in string

```
00009220  %s?ipaddr=%s&hostname=%s&procs=%s
00009248  photo.png
```

It indicate that it can downloads the photo from the internet using http

```
0000B2D8  Connection: close
0000B2F0  HTTP/1.1
0000B300  Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36
0000B378  ntdll.dll
0000B388  NtQuerySystemInformation
```

In wireshark

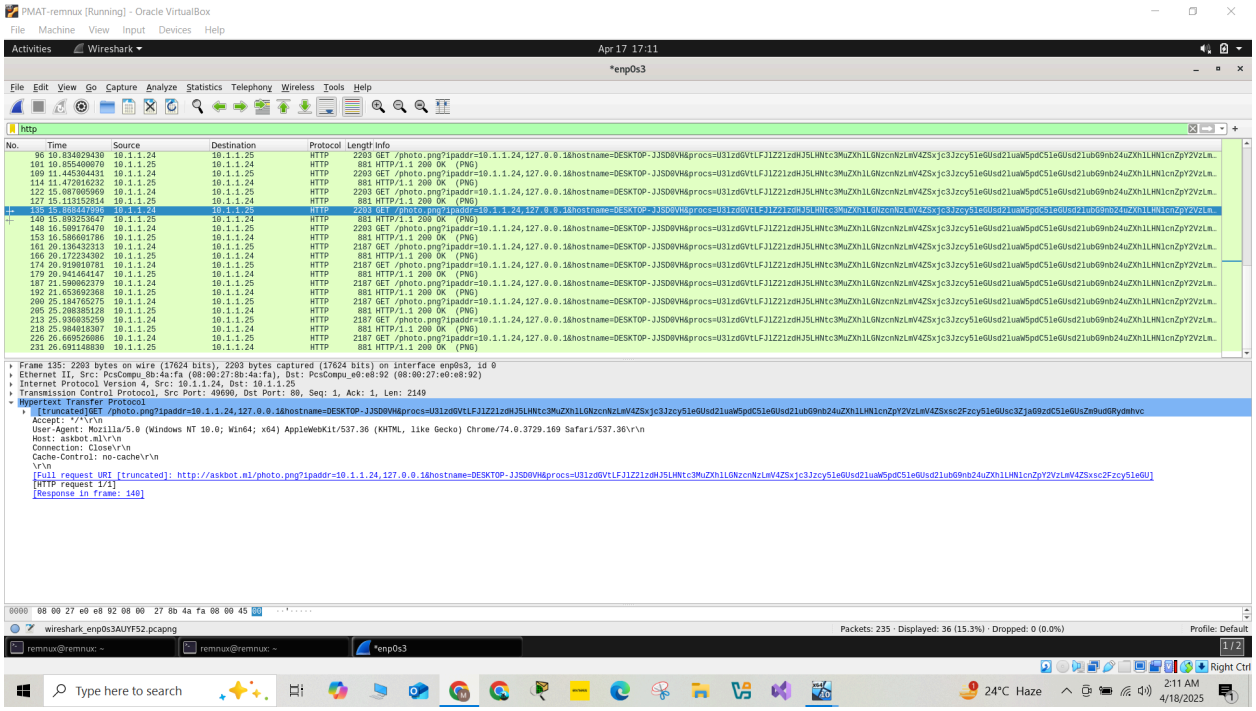

Figure 13

Look **Figure 13** that http request for downloading from malware but my Inetsim give him fake response

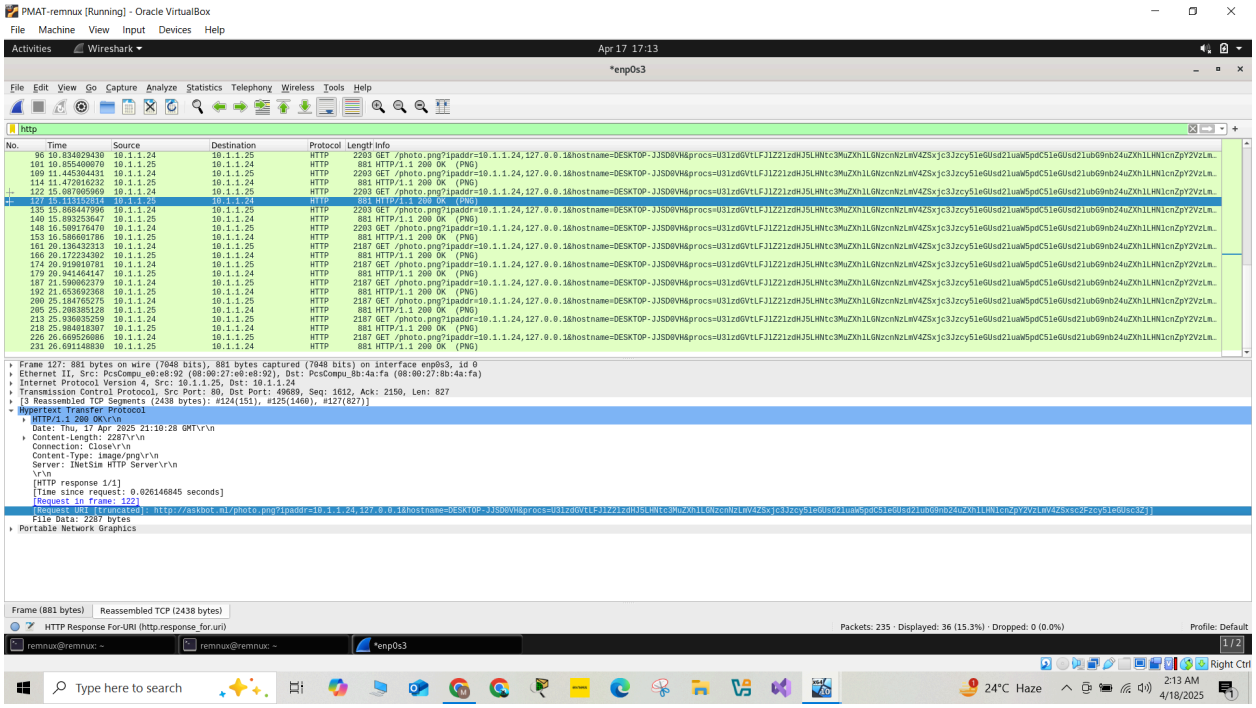

Figure 14

Next filter i use is **“http.user_agent contains "Mozilla/5.0"” as you can see Figure 20** Because this indicator was given in the strings , its means that i also get request from the mozilla browser for png photo

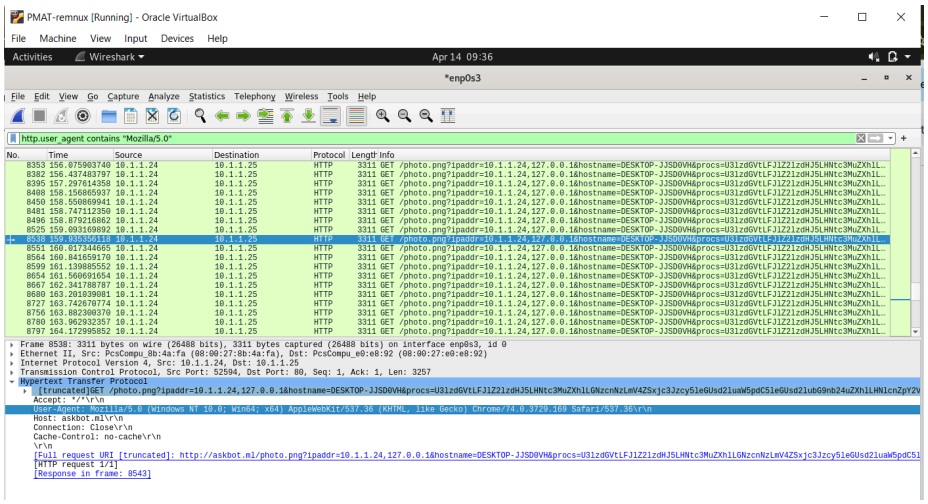

Figure 15

Summary

The malware attempted multiple outbound connections to suspicious IP addresses over HTTP.Unusual DNS queries were observed,

2.3 PROCMON Analysis

The purpose of using Process Monitor (Procmon) in this analysis is to observe and capture real-time system-level events generated by the execution of `ASKBot.exe`. This includes monitoring file system activity, registry modifications, and process behavior

Procmon Filter 1: (Process Name is ASKBot.exe)

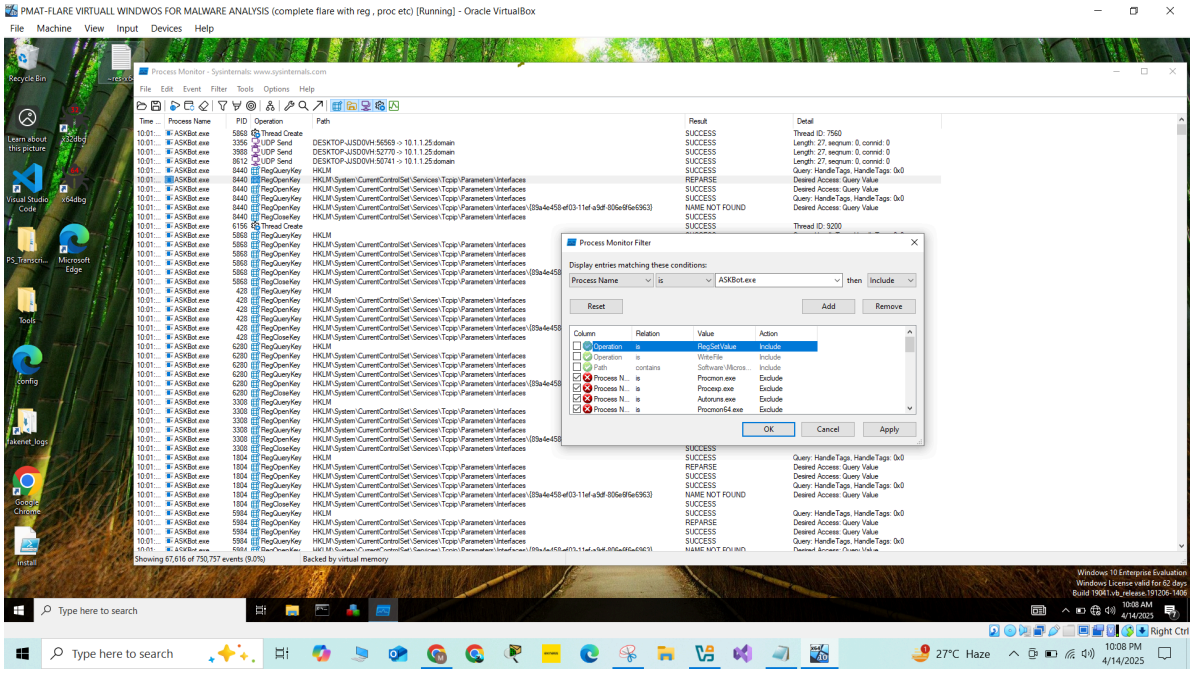

Figure 16

In Figure16 This filter isolates only the events generated by the malware process.It helps remove noise from other system processes.It ensures a focused view on the target executable's behavior

Procmon Filter 2: (Operation is CreateFile)

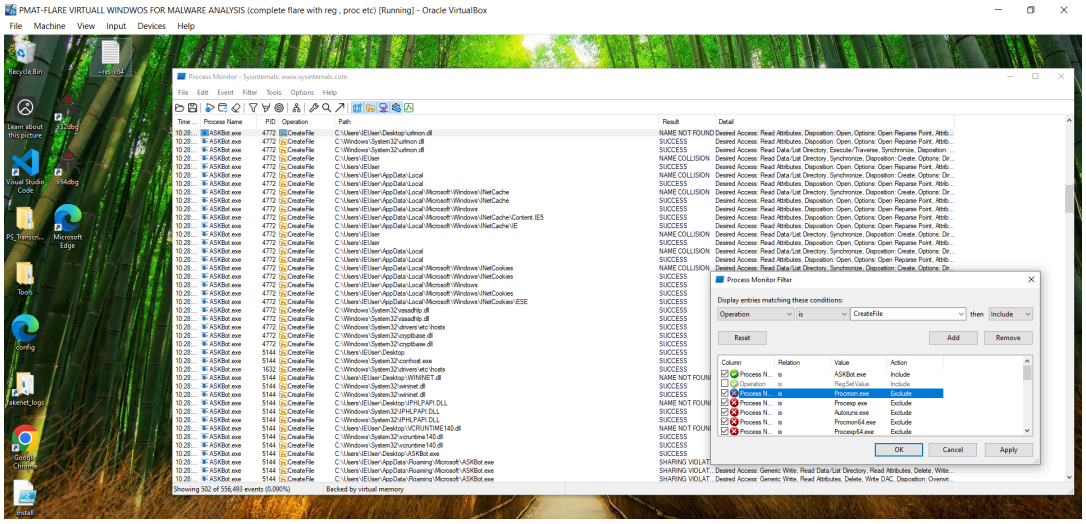

Figure 17

As you can see above **Figure 17** Filtering by **`CreateFile`** helps us **identify which files and locations the malware tries to access or manipulate**, including DLLs, config files, user data, or logs.

In screenshot, ASKBot.exe is attempting to access:

→System DLLs like **urmon.dll, cryptbase.dll, user32.dll**
→Internet cache directories like **INetCache, IE, Cookies**

This indicates that the malware is:

→**Loading dependencies dynamically**
→**Checking or modifying user-specific cache or cookie data, persistence**
→possibly creating **malicious file** and save into image

Procmon Filter 3: (Operation is RegSetValue)

below **Figure 18** Detects registry modifications — a key persistence method.Reveals changes to startup keys or malware configs.Helps identify how malware maintains control after reboot.

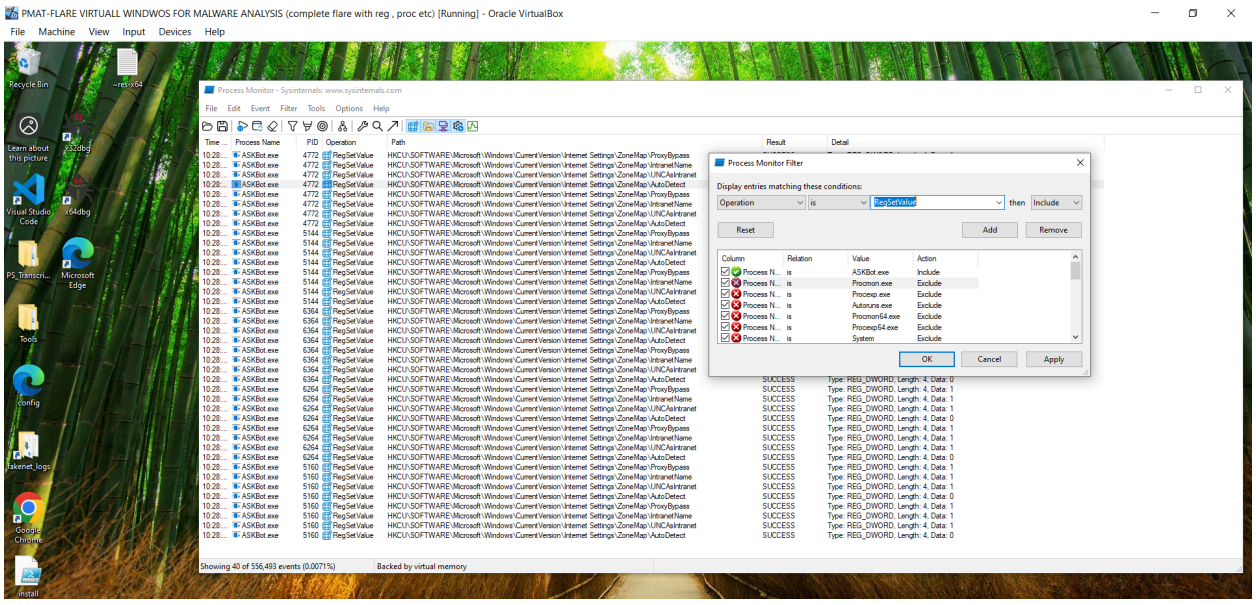

Figure 18

3.Task 3.2 – Debugging with x64dbg

Purpose of Debugger Analysis:

The purpose of using a debugger like x64dbg is to observe the real-time execution flow of the malware and understand its internal logic. It allows us to step through each instruction, analyze function calls, and monitor how the malware interacts with system components such as files, processes, and the registry.. Overall, debugger-based inspection provides deeper behavioral insights for reverse engineering and incident reporting.

Open in x64dbg:

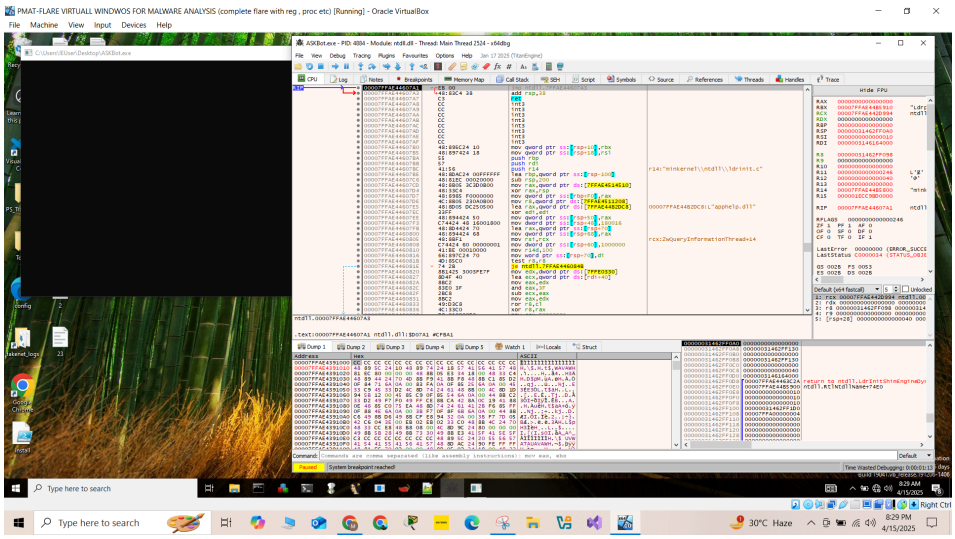

Figure 19

So after running and lot of the struggle i found the “entry point” was packed because there some reason

Checked in the x64dbg:

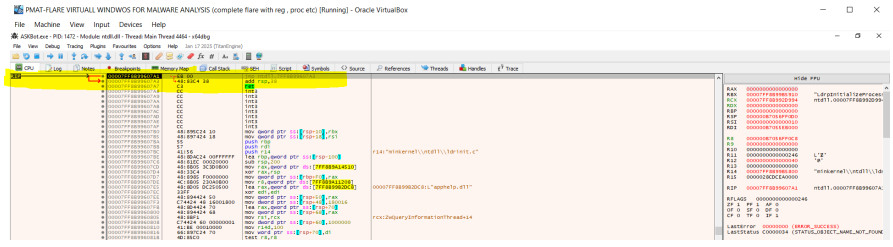

Figure 20

So in **Figure 20** when i open this sample the debugger automatically stop in this point and In this screenshot we can clearly see that program **jump** to the some function and then addition in the register then finally **(ret)**, so after analysing i see in the the particular address where the program want to jump.

Network indicators:

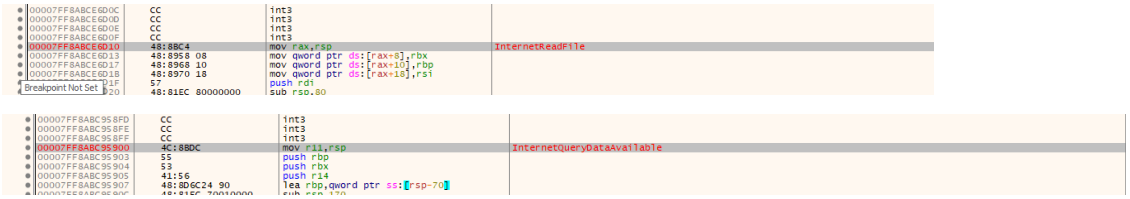

Figure 21

Above **Figure 21** These functions belong to the WinINet library, commonly used by malware for establishing connections to remote servers, sending **HTTP** requests. These api function related to **WinINet API calls** confirms that **ASKBot.exe** exhibits behavior consistent with **network-aware malware**

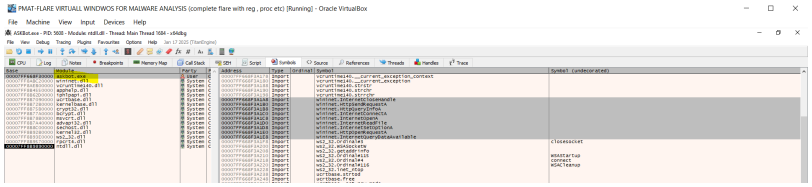

Figure 22

We should further see the **WinINet** for more network base call

WinINet

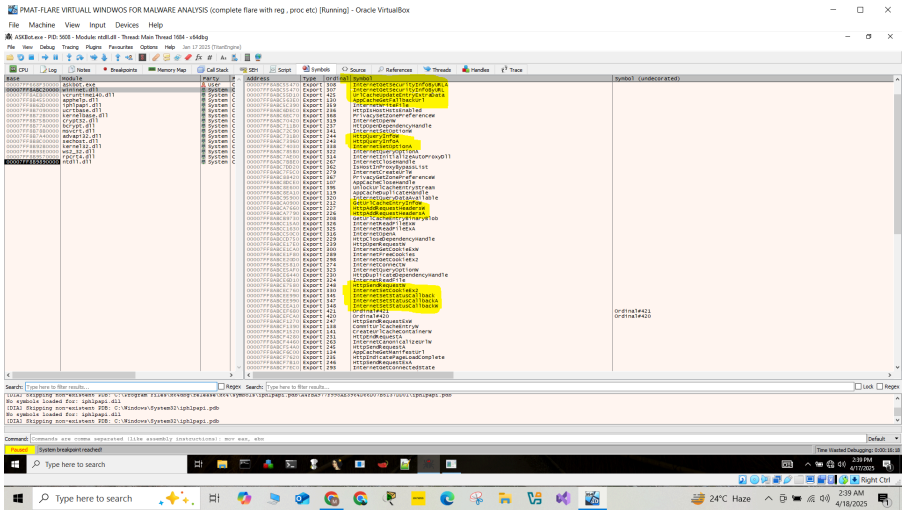

Figure 23

Like as you seen in **Figure 23** this we can clearly see in this that temption of the connect for internet happend

presence of these WinINet API calls confirms that **ASKBot.exe** exhibits behavior consistent with network-aware malware such as:

Registory keys modification found :

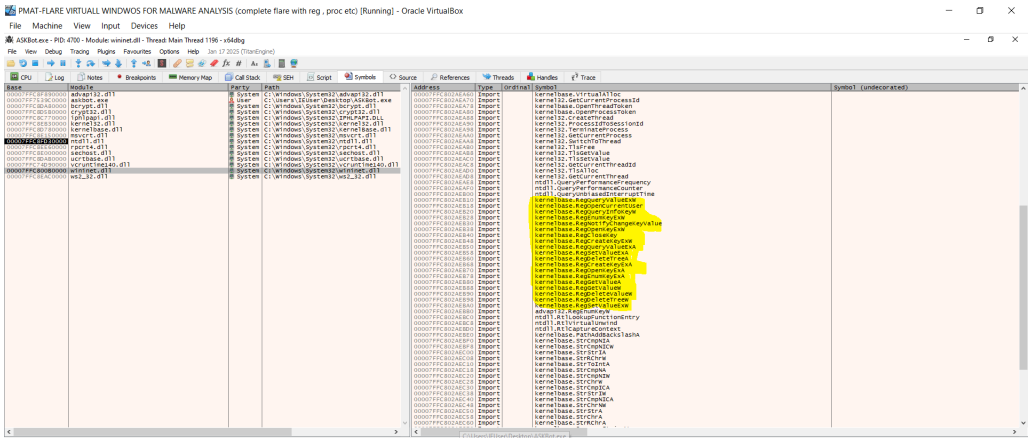

Figure 24