# An Overview Report of Running Malware Analysis using a piece of Malware

*A culminating experience report*

*submitted in partial fulfillment of the*

*requirements for the award of the degree*

*Of*

MASTER OF SCIENCE

IN

CYBERSECURITY

AT

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Submitted in

Fall 2023

by

Anvesh Raju Vishwaraju

801321127

# Table of Contents

# 1. Project Summary

## 1.1 Project Overview

The project “An Overview Report of Running Malware Analysis using a piece of Malware” was taken up as a part of “ITIS 6330 - Malware Analysis” course during the semester of Spring 2023. This was an individual project, which has assigned to every student or individual and, I worked with the professor Dr. Jinpeng Wei to successfully complete the project.

This project is aimed to provide a great insight on how a malware behaves when downloaded accidentally or with intention it is deployed on the target system. This is somewhat a typical process we have employed and used few behavioral analysis tools to analyze the malware behaviorit has to do with. This project gives full insight of the technology we have to perform the analysis for educational purpose than any other.

In this project we use many static and dynamic analysis of the malware behavior in a basic and an advanced ways the tools used for.

## 1.2 Motivation

The motivation behind choosing this project was to make people understand how a malware can affect the target system and how to analyze this generally this is termed as Reverse Engineering to know the behavior of malware. It can make the people more curious and cautious on how to secure their system because for an unethical attacker there is no difference between good and bad. This system is designed to make this entire process simple and easy for all the users. It was a very interesting course and topic to be taken as project to undergo whole semester since it is the first time to take up such project ever before.

## 2. Project Description

In this Project, the individuals will analyze a piece of a malware which is recently found and a real world malware. This helps the real time analysis and behavior of the malware by using different tools which are discussed in the course over the entire semester or Spring 2023 Term. The main aim of the project is to get hands on experience of the different tools which are used for both static and dynamical behavior of malware. Our task is to analyze and reveal the behaviors of given piece of malware. We have to install VirtualBox to operate this project as we require Windows XP VM, Ubuntu VM and Windows 7 VM for the behavior analysis. Once it is done we have to write a report covering the analysis part and results of the tools.

We will examine a number of topics, including the malware's complex multi-stage activity, the main "mur.exe," the secondary "mur.exe," and the related "RegSvcs.exe-1" and "RegSvcs.exe-2" processes. These features include script execution support, system shutdown and reboot capabilities, virtualization/sandbox evasion tactics, and anti-debugging approaches. We will also examine the differences between the two instances of "RegSvcs.exe" and look at the spawning of numerous processes. In addition, we will examine components like domain names, data that has been exfiltrated, encryption, encoding, and keylogging capabilities.

### 2.1 Goals

The main goals of the project are

- It analyzes the malware behavior
- It focuses on different areas of malware programs and dissect into the sub processes or child processes.

## 2.2 Scope

The main intention behind this project is to give students advanced knowledge about malware analysis by giving some hands – on experience on analyzing real time recent malware specimen. Students will examine using static and dynamic analysis in detail, which enhances experience of hands – on.

It provides a chance to explore process interactions, anti-debugging techniques and muti-stage activities. Students will get a thorough understanding of malware analysis by concentrating on the primary and secondary malware components as well as related processes. This will ensure that they are equipped to handle cybersecurity concerns in the future.

Moreover, the creation of comprehensive and repeatable analysis reports is included in the project's scope. These reports improve knowledge sharing and cooperative learning by becoming an invaluable resource for the larger cybersecurity community as well as the students themselves. Furthermore, the project's component weighting stresses accuracy and thoroughness in the analysis process, which fosters the growth of critical analytical abilities and the capacity to effectively communicate findings in writing form.

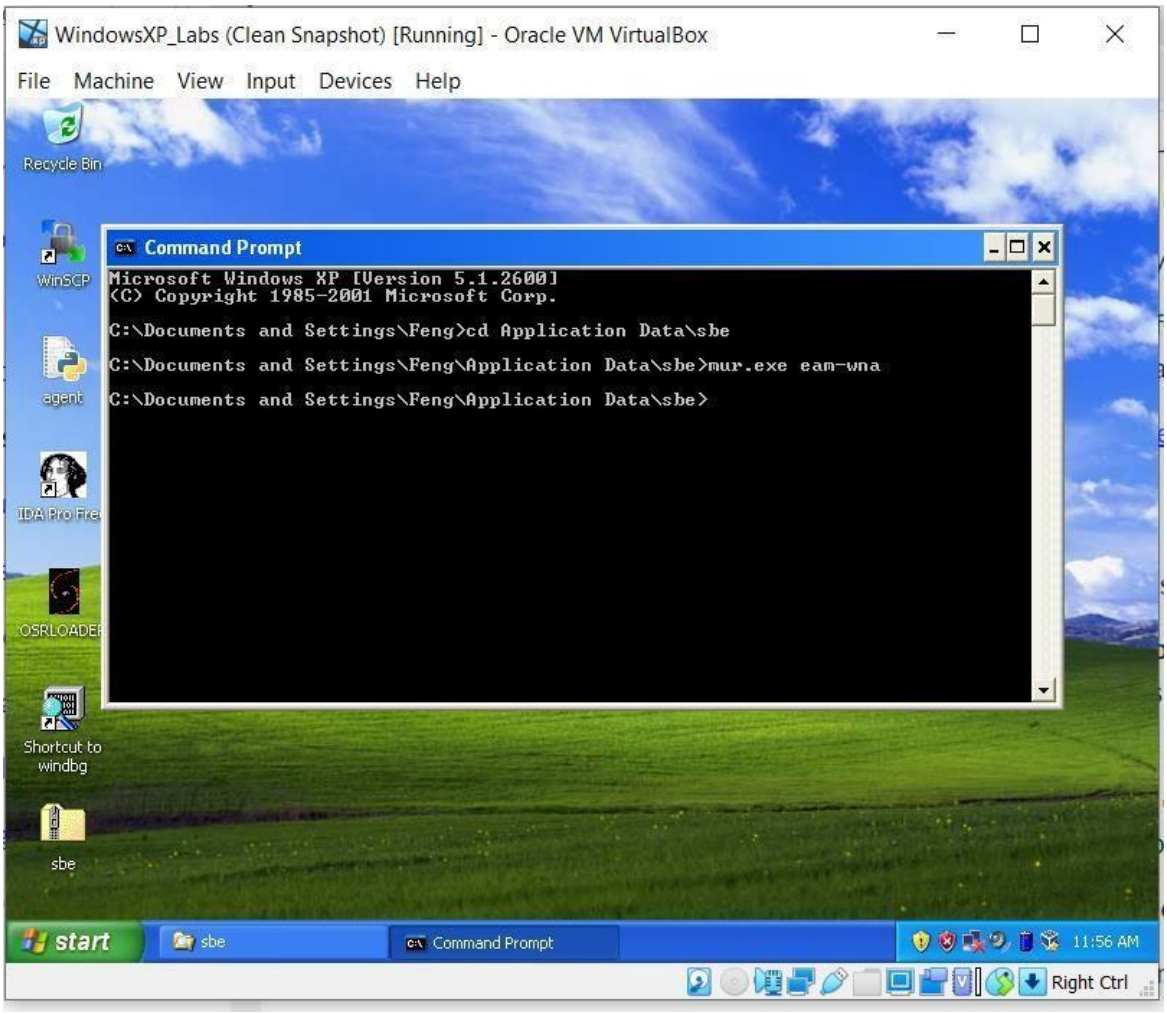
## 2.3 Environment Description

The following is the environment we have used :

- Virtual Environment Software : Oracle VirtualBox
- VM's installed with all the tools: Windows XP, Windows 7 and Ubuntu
- Report Writing Tool : MS – Word.

## 2.4 Malware Behavior Analysis :

I have downloaded the file sbe.zip in order to do malware analysis on it. Then, I simply extracted it by entering the password "infected," and then moved the file to C:\Documents and Settings\Feng\Application Data directory on Windows XP to carryout the analysis. The next step is performing the analysis part by running the malware by placing in the folder Application Data and then **mur.exe eam-wna** is executed in the command prompt and then malware just started running (Note: It may shutdown the VM but you should take snapshot of it and then run again from snapshot).




**We then run Basic Static Analysis of the malware behavior :**

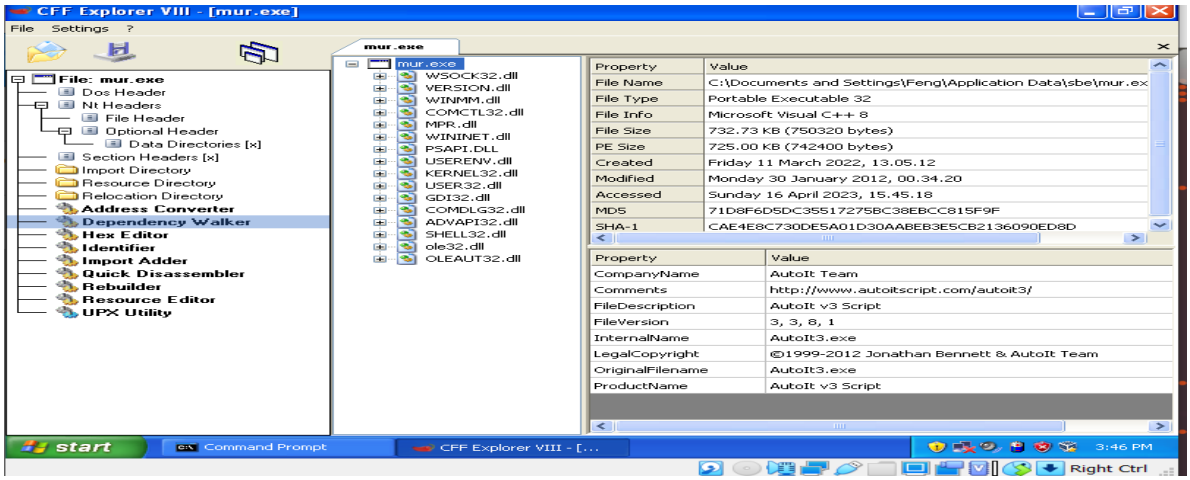


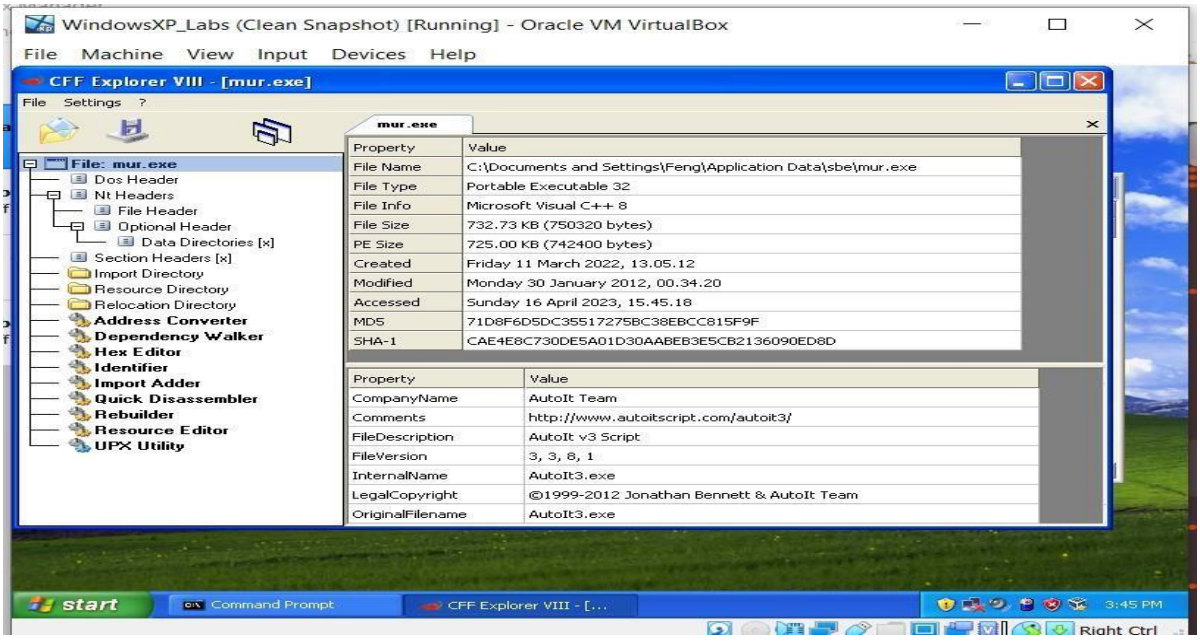
We used the static analysis tools such as: CFF Explorer, PEID etc., as you can check from the

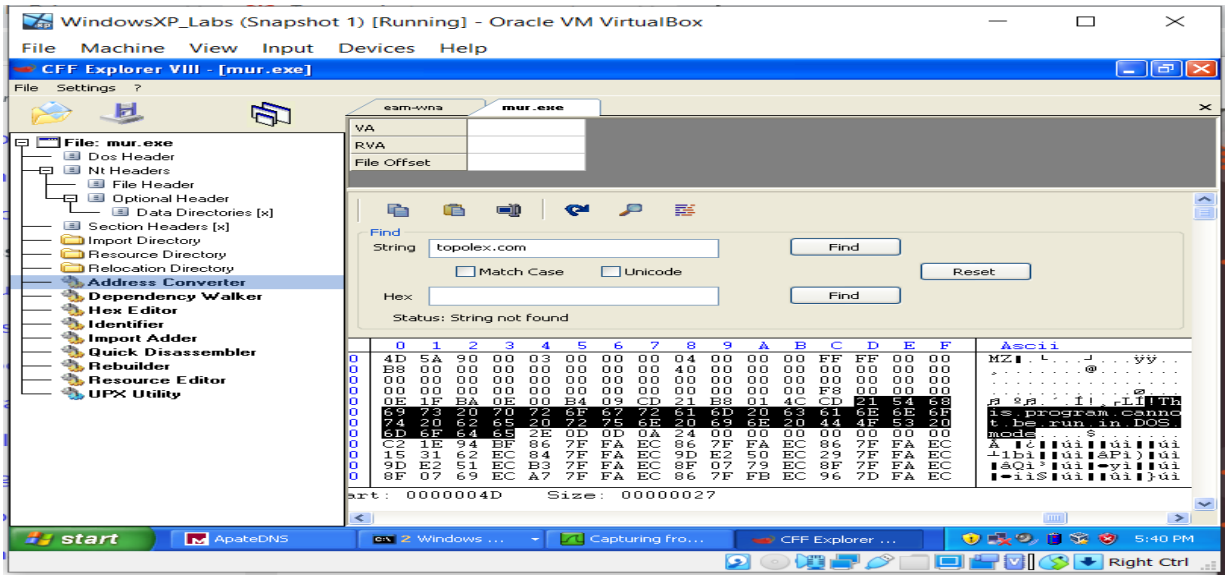
following steps :

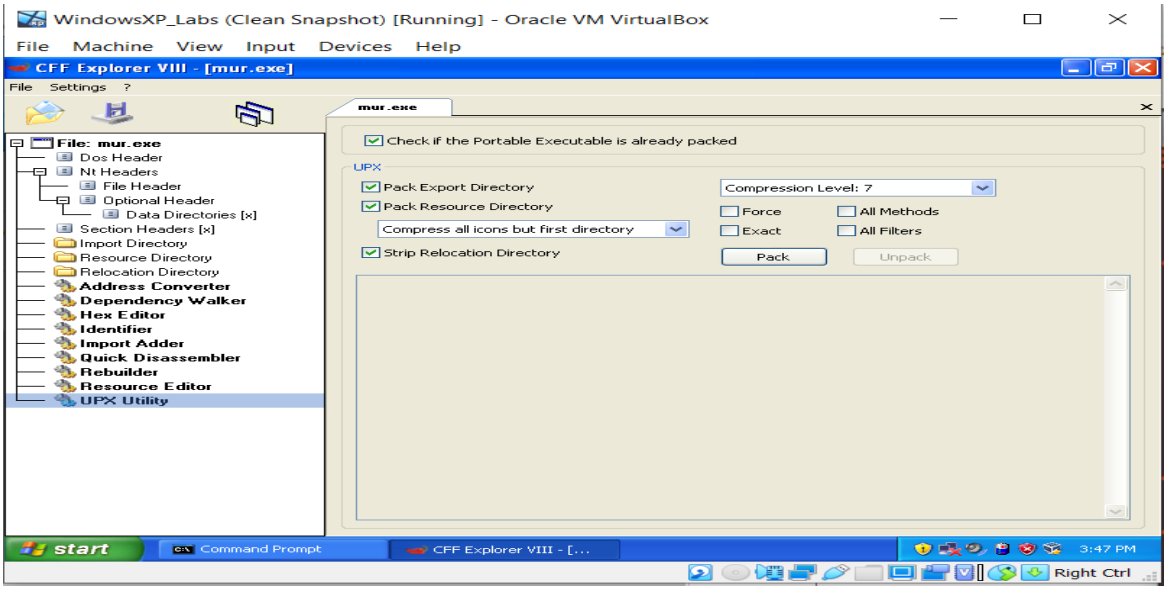
**Using CFF Explorer :**

With the hep of CFF Explorer, we have seen the Imports Directory, check whether malware is packed or not.



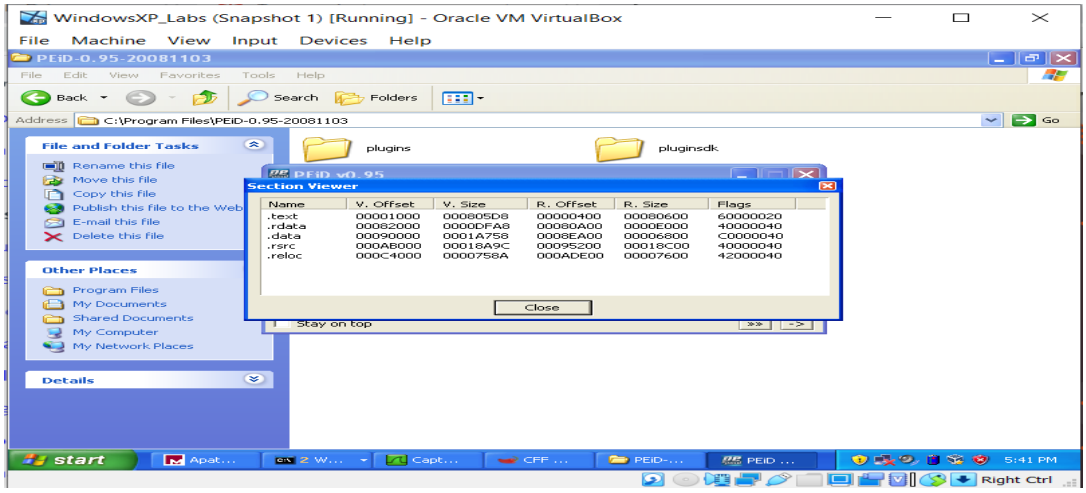


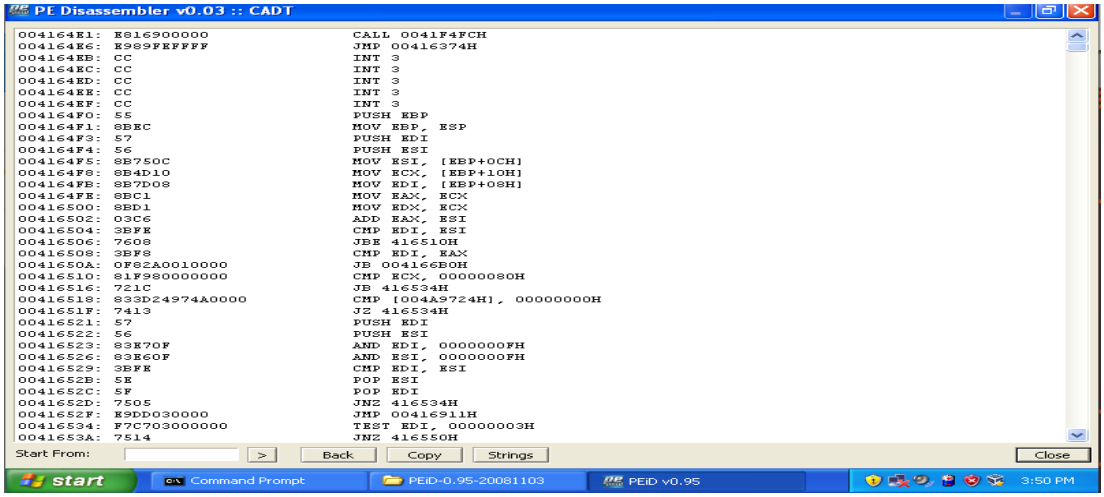



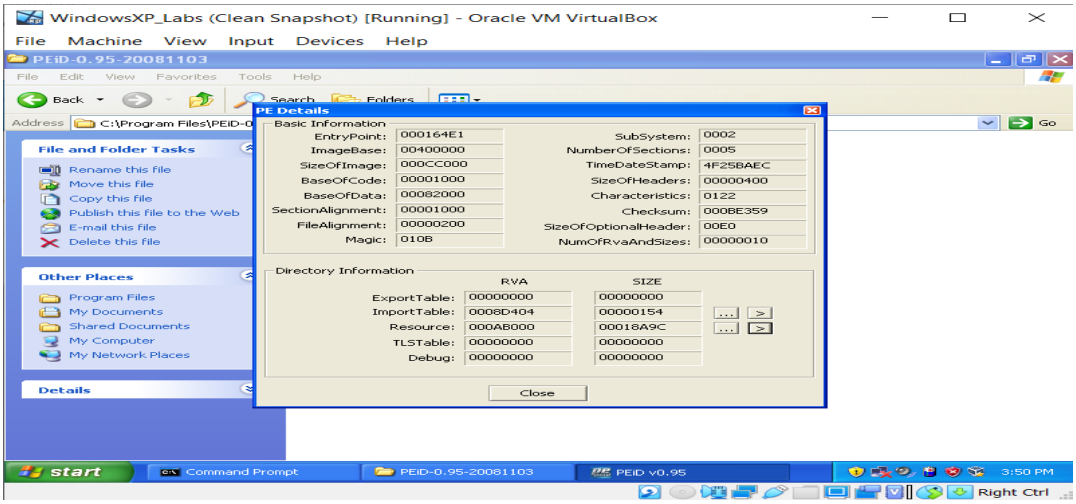

From this we can say malware is unpacked.

**Using PEID tool :**

It performs static analysis


WindowsXP_Labs (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
PEiD-0.95-20081103
Section Viewer
Name V. Offset V. Size R. Offset R. Size Flags
.text 00001000 000805D8 00000400 00080600 60000020
.rdata 00082000 0000DFA8 00080A00 0008E000 40000040
.data 00090000 0001A758 0008EA00 00006800 C0000040
.rsrc 000AB000 00018A9C 00095200 00018C00 40000040
.reloc 000C4000 0000758A 000ADE00 00007600 42000040
Close
Stay on top
plugins
pluginsdk
Right Ctrl

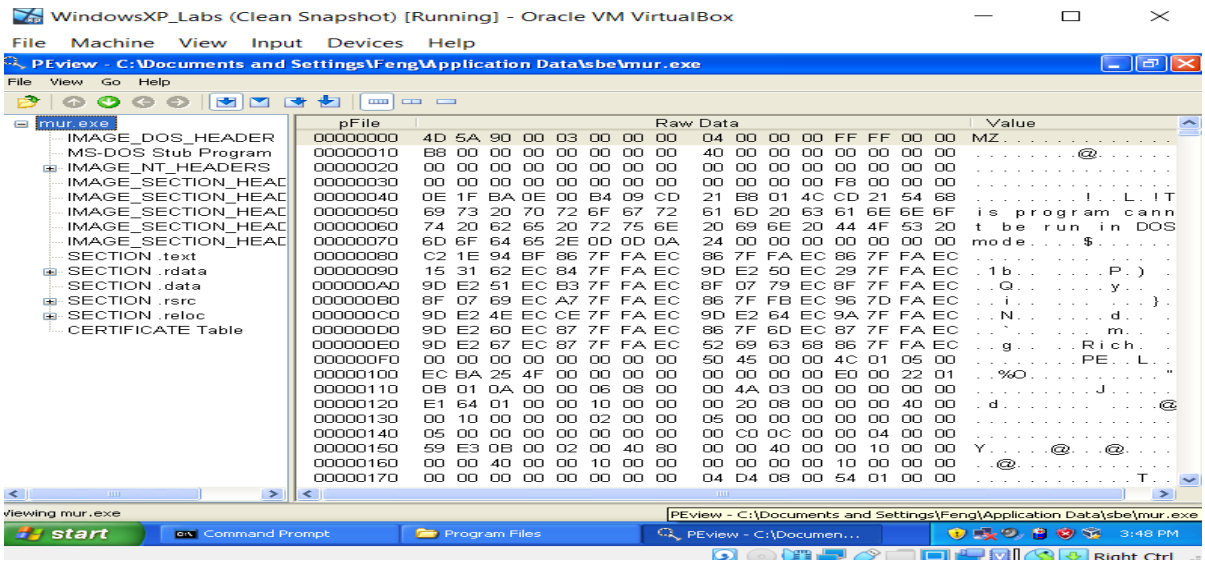

PE Disassembler v0.03 :: CADT
004164E1: E816900000 CALL 0041F4FCH
004164E6: E989FEFFFF JMP 00416374H
004164EB: CC INT 3
004164EC: CC INT 3
004164ED: CC INT 3
004164EE: CC INT 3
004164EF: CC INT 3
004164F0: 55 PUSH EBP
004164F1: 8BEC MOV EBP, ESP
004164F3: 57 PUSH EDI
004164F4: 56 PUSH ESI
004164F5: 8B750C MOV ESI, [EBP+0CH]
004164F8: 8B4D10 MOV ECX, [EBP+10H]
004164FB: 8B7D08 MOV EDI, [EBP+08H]
004164FE: 8BC1 MOV EAX, ECX
00416500: 8BD1 MOV EDX, ECX
00416502: 03C6 ADD EAX, ESI
00416504: 3BFE CMP EDI, ESI
00416506: 7608 JBE 416510H
00416508: 3BF8 CMP EDI, EAX
0041650A: 0F82A0010000 JB 004166B0H
00416510: 81F980000000 CMP ECX, 00000080H
00416516: 721C JB 416534H
00416518: 833D24974A0000 CMP [004A9724H], 00000000H
0041651F: 7413 JZ 416534H
00416521: 57 PUSH EDI
00416522: 56 PUSH ESI
00416523: 83E70F AND EDI, 0000000FH
00416526: 83E60F AND ESI, 0000000FH
00416529: 3BFE CMP EDI, ESI
0041652B: 5E POP ESI
0041652C: 5F POP EDI
0041652D: 7505 JNZ 416534H
0041652F: E9DD030000 JMP 00416911H
00416534: F7C703000000 TEST EDI, 00000003H
0041653A: 7514 JNZ 416550H
Start From:
Back
Copy
Strings
Close
Command Prompt
PEiD-0.95-20081103
PEiD v0.95
3:50 PM


WindowsXP_Labs (Clean Snapshot) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
PEiD-0.95-20081103
PE Details
Basic Information
EntryPoint: 000164E1
ImageBase: 00400000
SizeOfImage: 000CC000
BaseOfCode: 00001000
BaseOfData: 00082000
SectionAlignment: 00001000
FileAlignment: 00000200
Magic: 010B
SubSystem: 0002
NumberOfSections: 0005
TimeDateStamp: 4F25BAEC
SizeOfHeaders: 00000400
Characteristics: 0122
Checksum: 000BE359
SizeOfOptionalHeader: 00E0
NumOfRvaAndSizes: 00000010
Directory Information
RVA SIZE
ExportTable: 00000000 00000000
ImportTable: 0008D404 00000154
Resource: 000AB000 00018A9C
TLSTable: 00000000 00000000
Debug: 00000000 00000000
Close
Command Prompt
PEiD-0.95-20081103
PEiD v0.95
3:50 PM
Right Ctrl

**Using PEView Tool :**

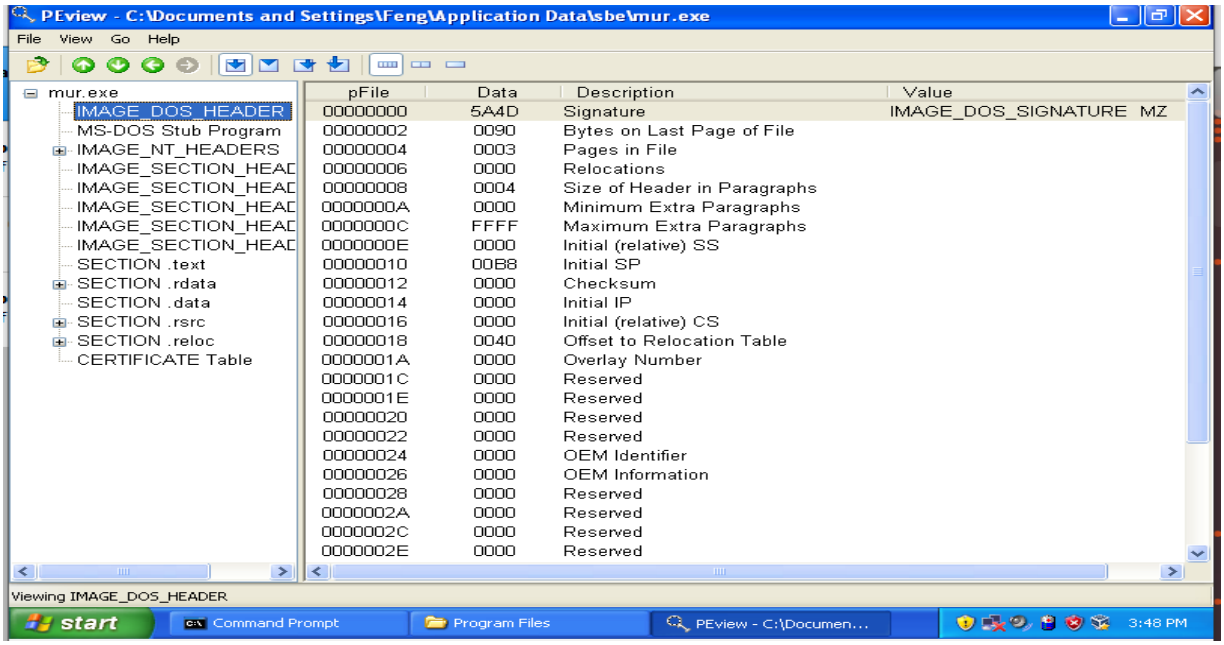
This is the another static analysis tool for the analysis of Malware Behavior and we got the following results after performing the analysis :

We generally interpreted the information in Hex View in the PEView tool:



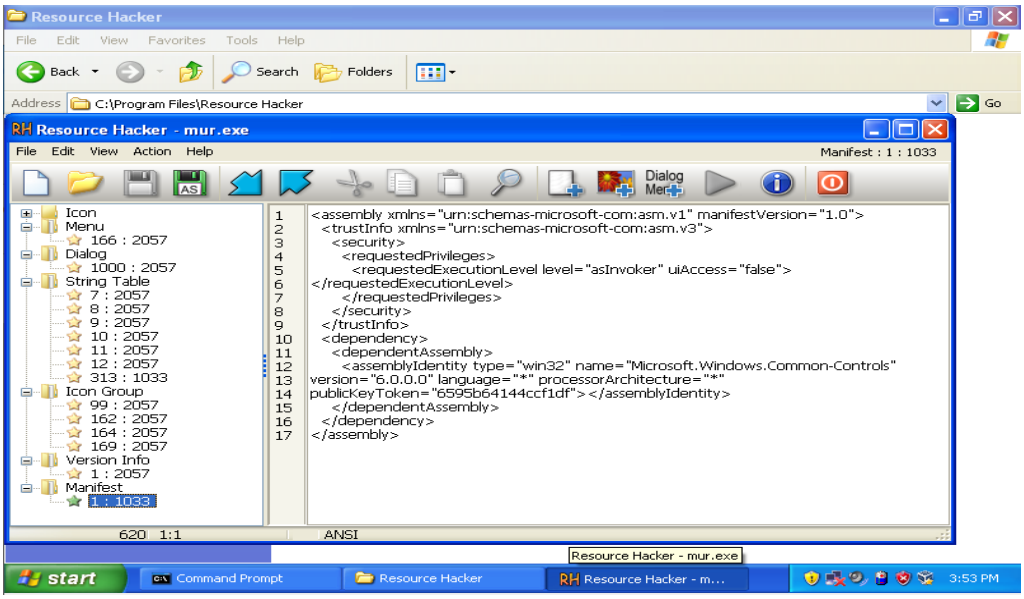

From the following image, we can see that the DOS Header is MZ which is interpreted as 5A4D in Hexadecimal format which represents the signature of the file and we can see the other details as follows in the image:

1. Bytes on Last Page of File
2. Pages in File
3. Size of Headers in Paragraphs
4. CheckSum
5. Offset to Relocation Tables
6. OEM Identifier, Information And also we can see some are Reserved




**Use of Resource Hacker tool :**

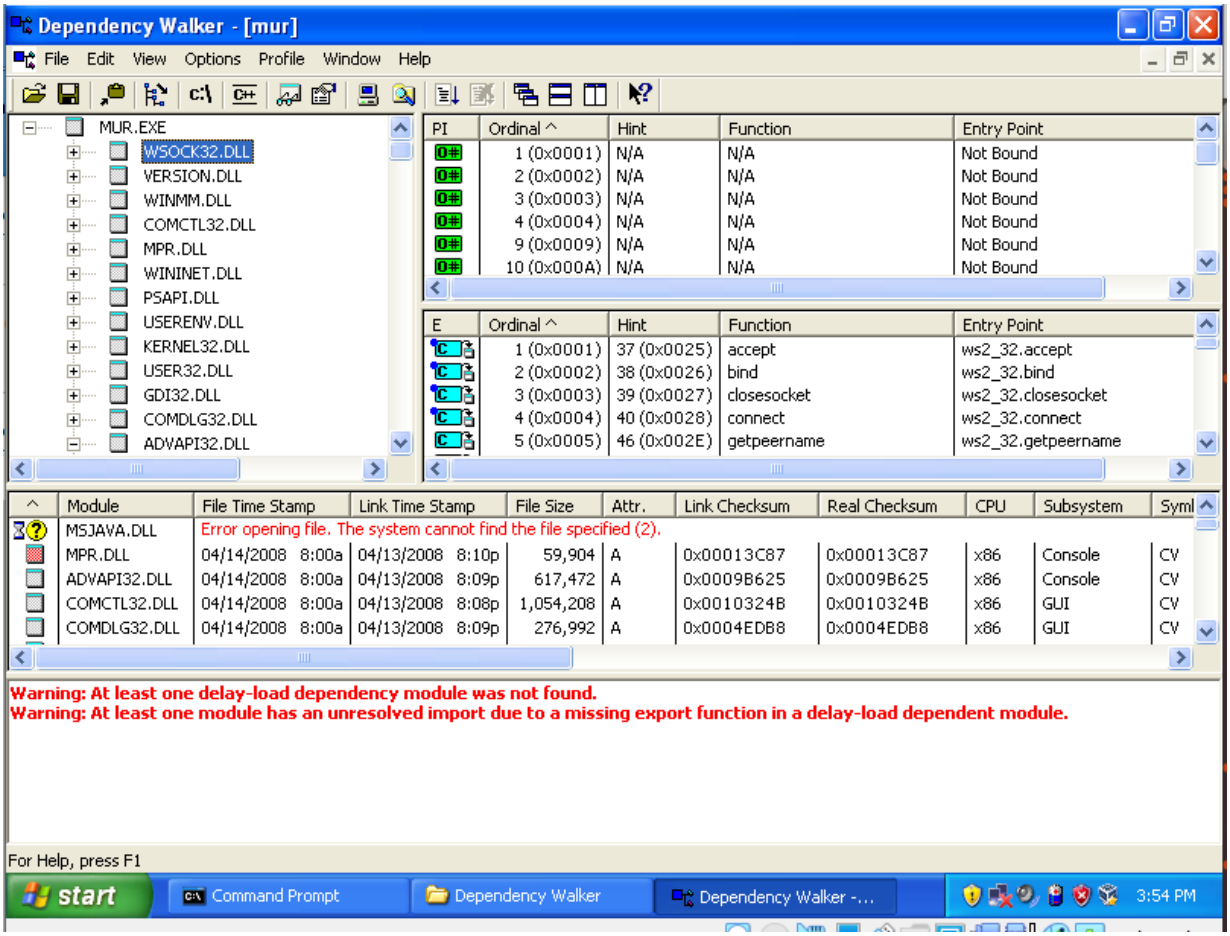
We open the mur.exe in resource hacker and get to know the resources the malware is utilizing to cause problems:




And then we save as .res file which we use to open in dependency walker.

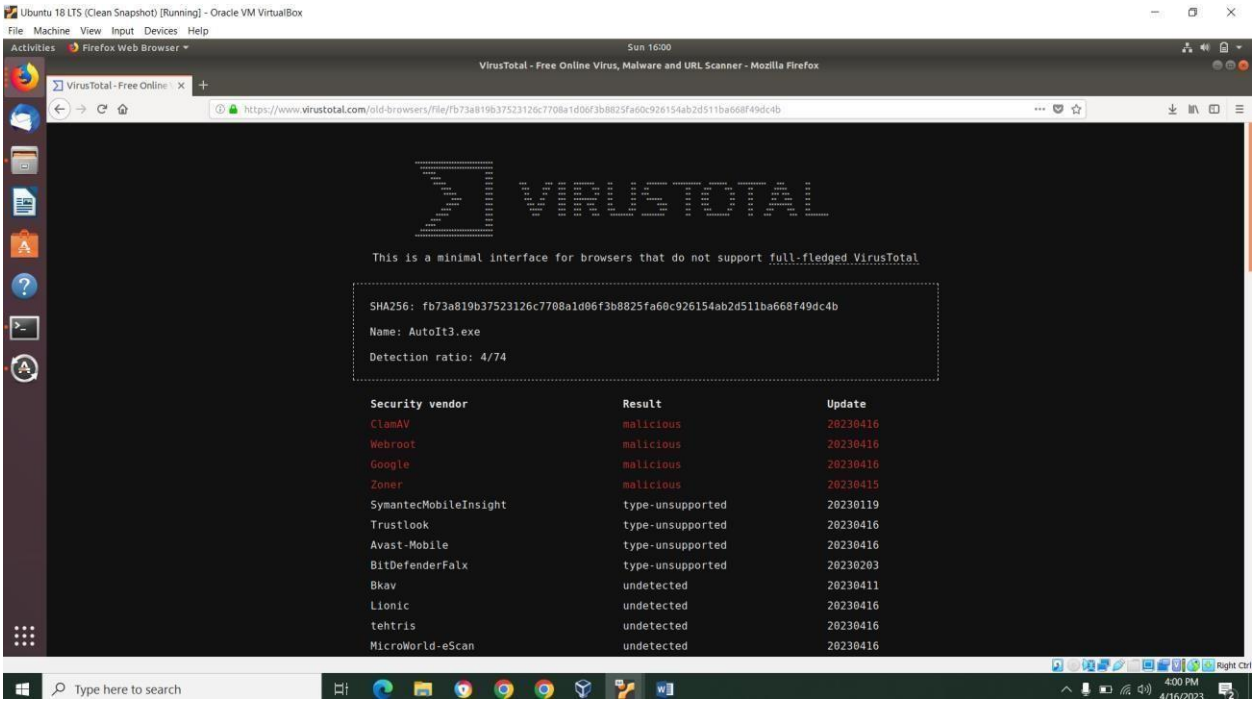
## Use of Dependency Walker Tool:

We use that .res file and open in the dependency walker to list the DLL's and check for the bytes and different modules, functions.



## Use of VirusTotal.com website:

With the help of the website , we can see that the Detection Ratio is 4/74 and the name as AutoIt3.exe

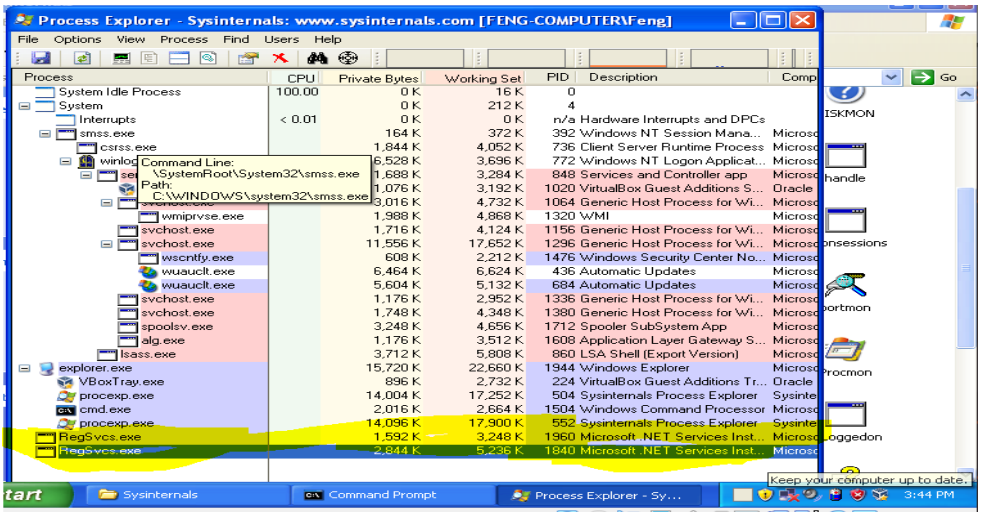



## Running the dynamic analysis of malware:

We use the following tools to perform the dynamic analysis of malware behavior and as follows:

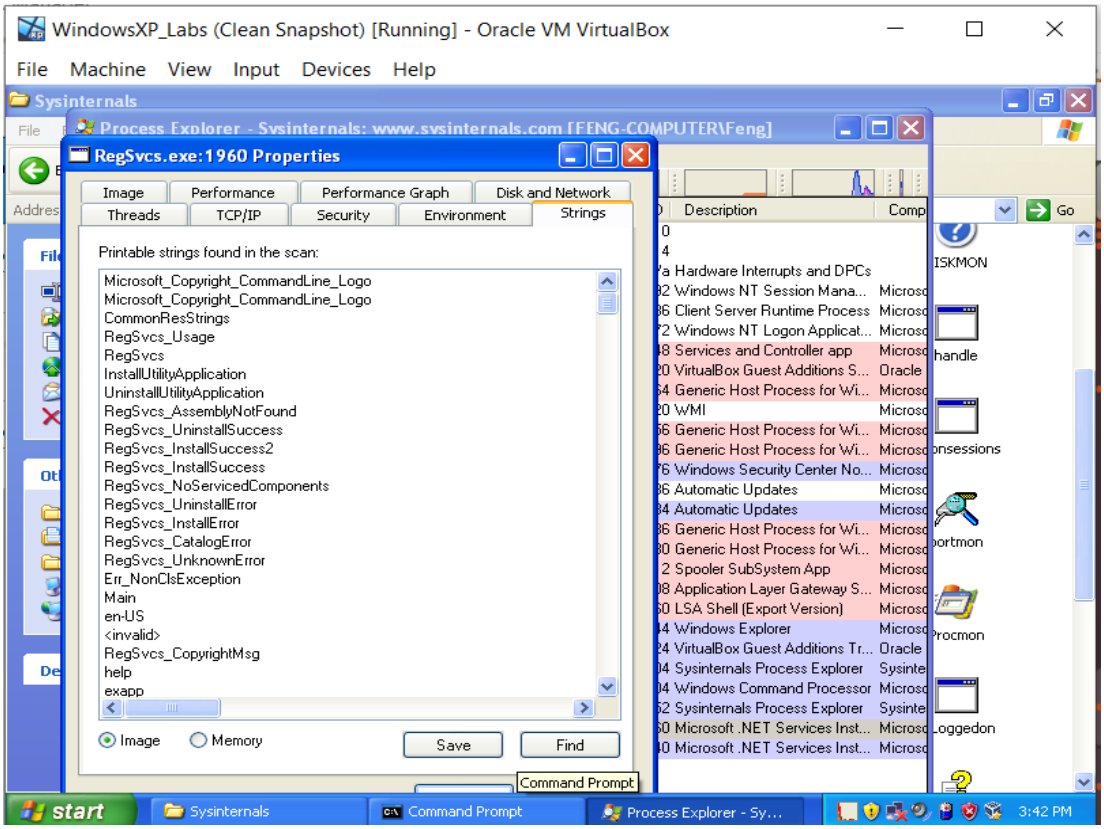
## Use of Process Explorer for the analysis :

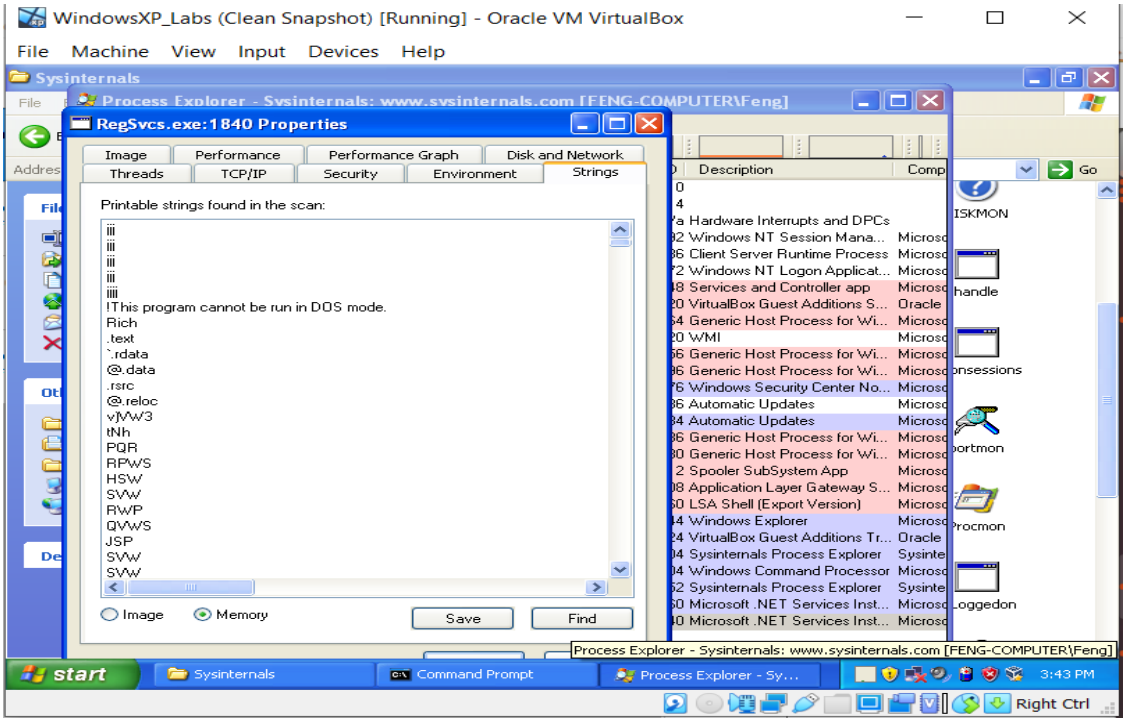
We have used this tool to perform the analysis and get the information of the processes and its child processes , strings from the image and memory and other details as follows from the following the screenshots :




We can see that it propagated the following child processes which are highlighted along with the process ID's 1960 and 1840 respectively.

We see the some properties, strings by double clicking on them both the processes:

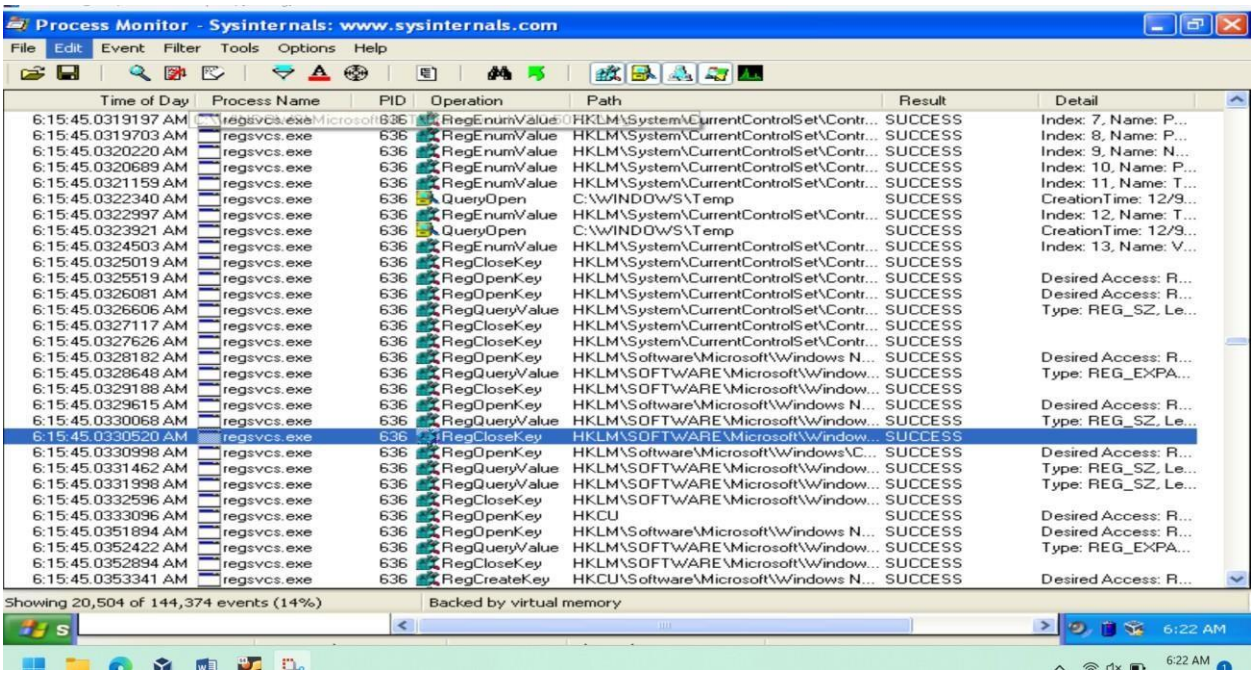


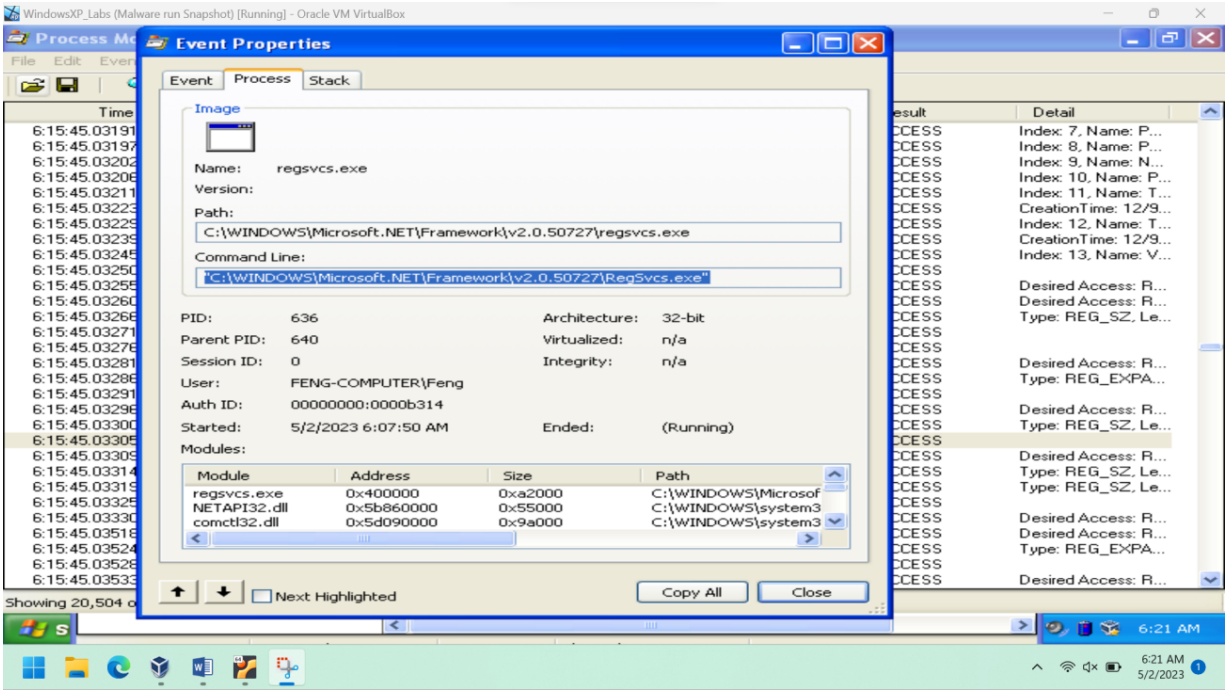



## Use of Process Monitor Tool :

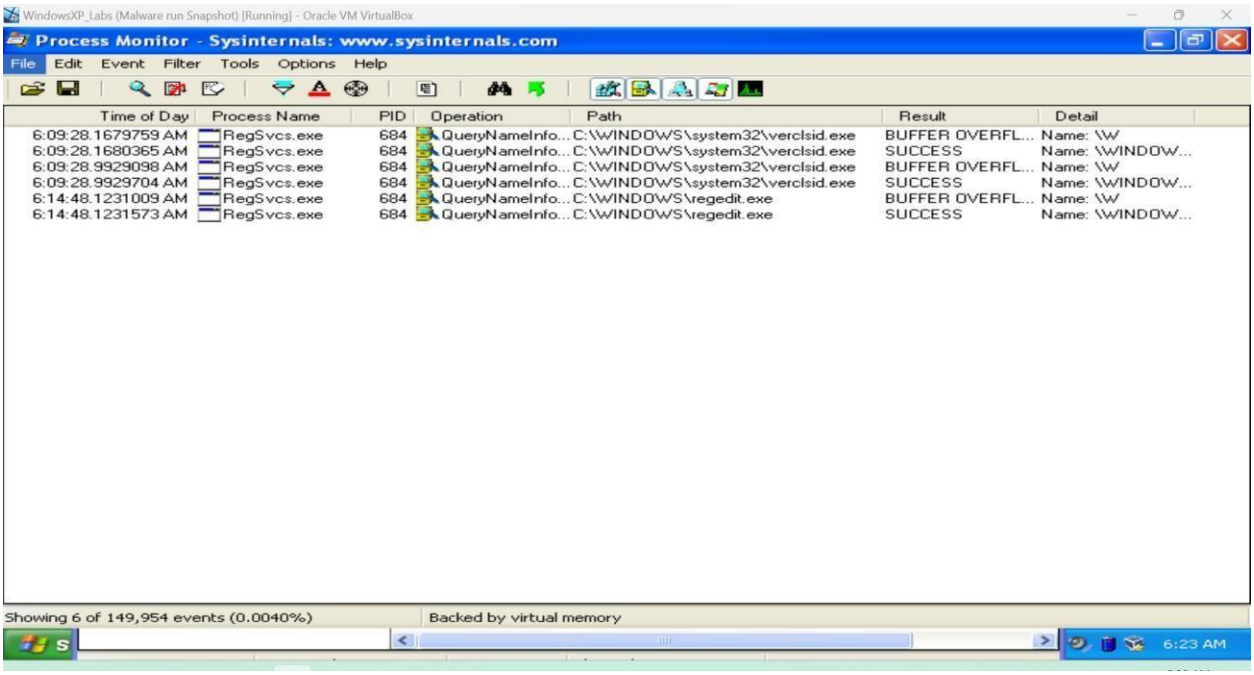
We used this tool to monitor the processes with the help of filtering the results by PID (Process ID).

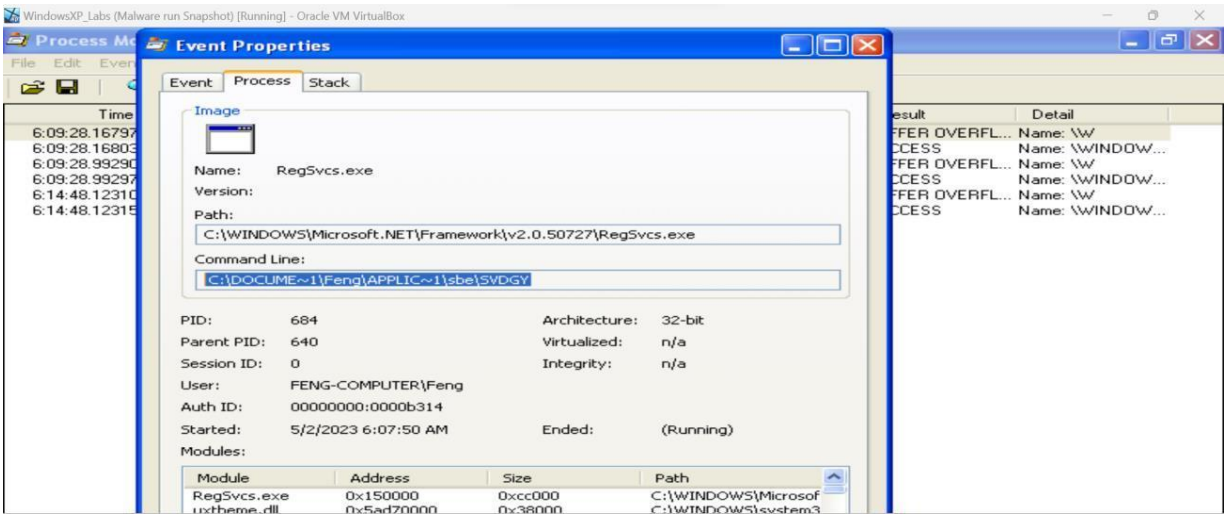
**RegSvcs.exe with the PID:** 636

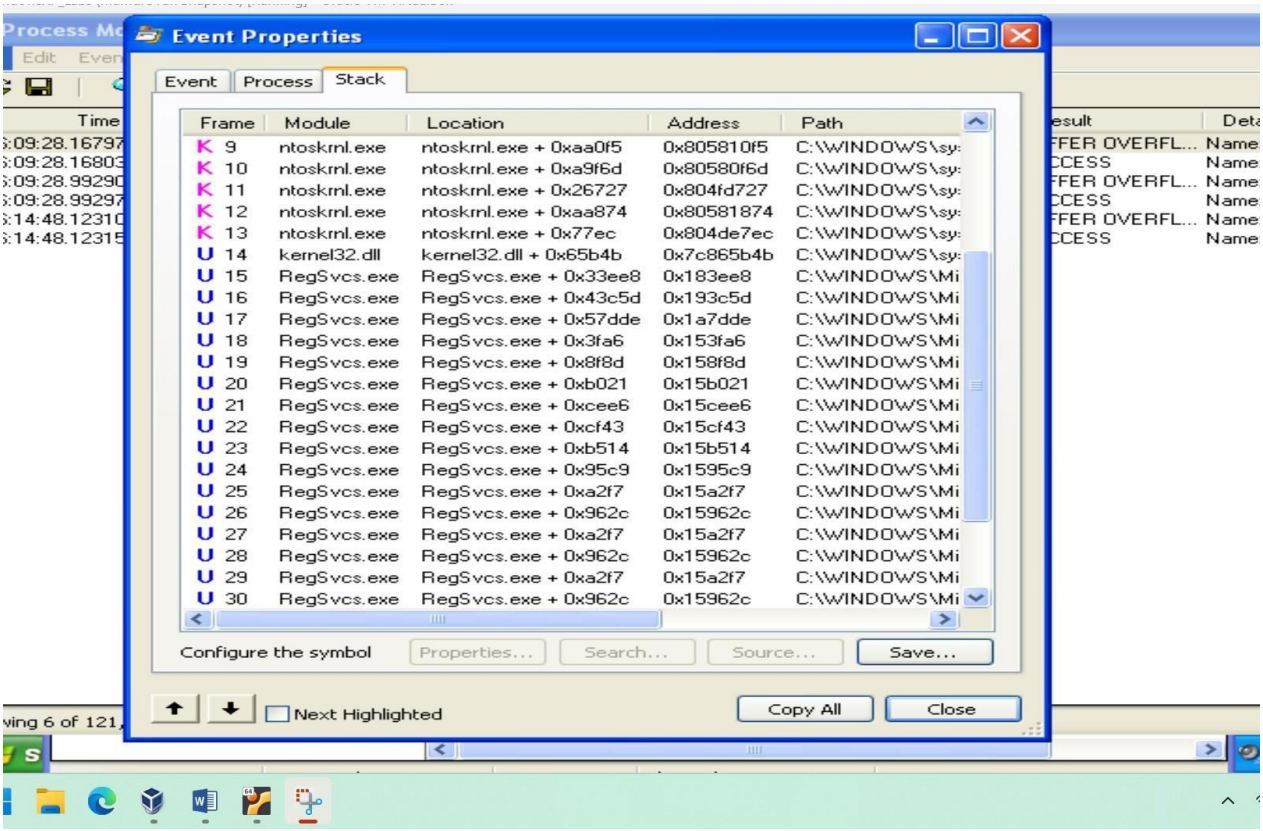

Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Time of Day | Process Name | PID | Operation | Path | Result | Detail
6:15:45.0319197 AM regsvcs.exe 636 RegEnumValue HKLM\System\CurrentControlSet\Contr... SUCCESS Index: 7, Name: P...
6:15:45.0319703 AM regsvcs.exe 636 RegEnumValue HKLM\System\CurrentControlSet\Contr... SUCCESS Index: 8, Name: P...
6:15:45.0320220 AM regsvcs.exe 636 RegEnumValue HKLM\System\CurrentControlSet\Contr... SUCCESS Index: 9, Name: N...
6:15:45.0320689 AM regsvcs.exe 636 RegEnumValue HKLM\System\CurrentControlSet\Contr... SUCCESS Index: 10, Name: P...
6:15:45.0321159 AM regsvcs.exe 636 RegEnumValue HKLM\System\CurrentControlSet\Contr... SUCCESS Index: 11, Name: T...
6:15:45.0322340 AM regsvcs.exe 636 QueryOpen C:\WINDOWS\Temp SUCCESS CreationTime: 12/9...
6:15:45.0322997 AM regsvcs.exe 636 RegEnumValue HKLM\System\CurrentControlSet\Contr... SUCCESS Index: 12, Name: T...
6:15:45.0323921 AM regsvcs.exe 636 QueryOpen C:\WINDOWS\Temp SUCCESS CreationTime: 12/9...
6:15:45.0324503 AM regsvcs.exe 636 RegEnumValue HKLM\System\CurrentControlSet\Contr... SUCCESS Index: 13, Name: V...
6:15:45.0325019 AM regsvcs.exe 636 RegCloseKey HKLM\System\CurrentControlSet\Contr... SUCCESS
6:15:45.0325519 AM regsvcs.exe 636 RegOpenKey HKLM\System\CurrentControlSet\Contr... SUCCESS Desired Access: R...
6:15:45.0326081 AM regsvcs.exe 636 RegOpenKey HKLM\System\CurrentControlSet\Contr... SUCCESS Desired Access: R...
6:15:45.0326606 AM regsvcs.exe 636 RegQueryValue HKLM\System\CurrentControlSet\Contr... SUCCESS Type: REG_SZ, Le...
6:15:45.0327117 AM regsvcs.exe 636 RegCloseKey HKLM\System\CurrentControlSet\Contr... SUCCESS
6:15:45.0327626 AM regsvcs.exe 636 RegCloseKey HKLM\System\CurrentControlSet\Contr... SUCCESS
6:15:45.0328182 AM regsvcs.exe 636 RegOpenKey HKLM\Software\Microsoft\Windows N... SUCCESS Desired Access: R...
6:15:45.0328648 AM regsvcs.exe 636 RegQueryValue HKLM\SOFTWARE\Microsoft\Window... SUCCESS Type: REG_EXPA...
6:15:45.0329188 AM regsvcs.exe 636 RegCloseKey HKLM\SOFTWARE\Microsoft\Window... SUCCESS
6:15:45.0329615 AM regsvcs.exe 636 RegOpenKey HKLM\Software\Microsoft\Windows N... SUCCESS Desired Access: R...
6:15:45.0330068 AM regsvcs.exe 636 RegQueryValue HKLM\SOFTWARE\Microsoft\Window... SUCCESS Type: REG_SZ, Le...
6:15:45.0330520 AM regsvcs.exe 636 RegCloseKey HKLM\SOFTWARE\Microsoft\Window... SUCCESS
6:15:45.0330998 AM regsvcs.exe 636 RegOpenKey HKLM\Software\Microsoft\Windows\C... SUCCESS Desired Access: R...
6:15:45.0331462 AM regsvcs.exe 636 RegQueryValue HKLM\SOFTWARE\Microsoft\Window... SUCCESS Type: REG_SZ, Le...
6:15:45.0331998 AM regsvcs.exe 636 RegQueryValue HKLM\SOFTWARE\Microsoft\Window... SUCCESS Type: REG_SZ, Le...
6:15:45.0332596 AM regsvcs.exe 636 RegCloseKey HKLM\SOFTWARE\Microsoft\Window... SUCCESS
6:15:45.0333096 AM regsvcs.exe 636 RegOpenKey HKCU SUCCESS Desired Access: R...
6:15:45.0351894 AM regsvcs.exe 636 RegOpenKey HKLM\Software\Microsoft\Windows N... SUCCESS Desired Access: R...
6:15:45.0352422 AM regsvcs.exe 636 RegQueryValue HKLM\SOFTWARE\Microsoft\Window... SUCCESS Type: REG_EXPA...
6:15:45.0352894 AM regsvcs.exe 636 RegCloseKey HKLM\SOFTWARE\Microsoft\Window... SUCCESS
6:15:45.0353341 AM regsvcs.exe 636 RegCreateKey HKCU\Software\Microsoft\Windows N... SUCCESS Desired Access: R...
Showing 20,504 of 144,374 events (14%)
Backed by virtual memory
6:22 AM
6:22 AM


WindowsXP_Labs (Malware run Snapshot) [Running] - Oracle VM VirtualBox
Event Properties
Event Process Stack
Image
Name: regsvcs.exe
Version:
Path:
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\regsvcs.exe
Command Line:
"C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\RegSvcs.exe"
PID: 636
Parent PID: 640
Session ID: 0
User: FENG-COMPUTER\Feng
Auth ID: 00000000:0000b314
Started: 5/2/2023 6:07:50 AM
Architecture: 32-bit
Virtualized: n/a
Integrity: n/a
Ended: (Running)
Modules:
Module | Address | Size | Path
regsvcs.exe 0x400000 0xa2000 C:\WINDOWS\Microsof
NETAPI32.dll 0x5b860000 0x55000 C:\WINDOWS\system3
comctl32.dll 0x5d090000 0x9a000 C:\WINDOWS\system3
Next Highlighted
Copy All
Close
6:21 AM
6:21 AM
5/2/2023

**RegSvcs.exe with the PID:** 684

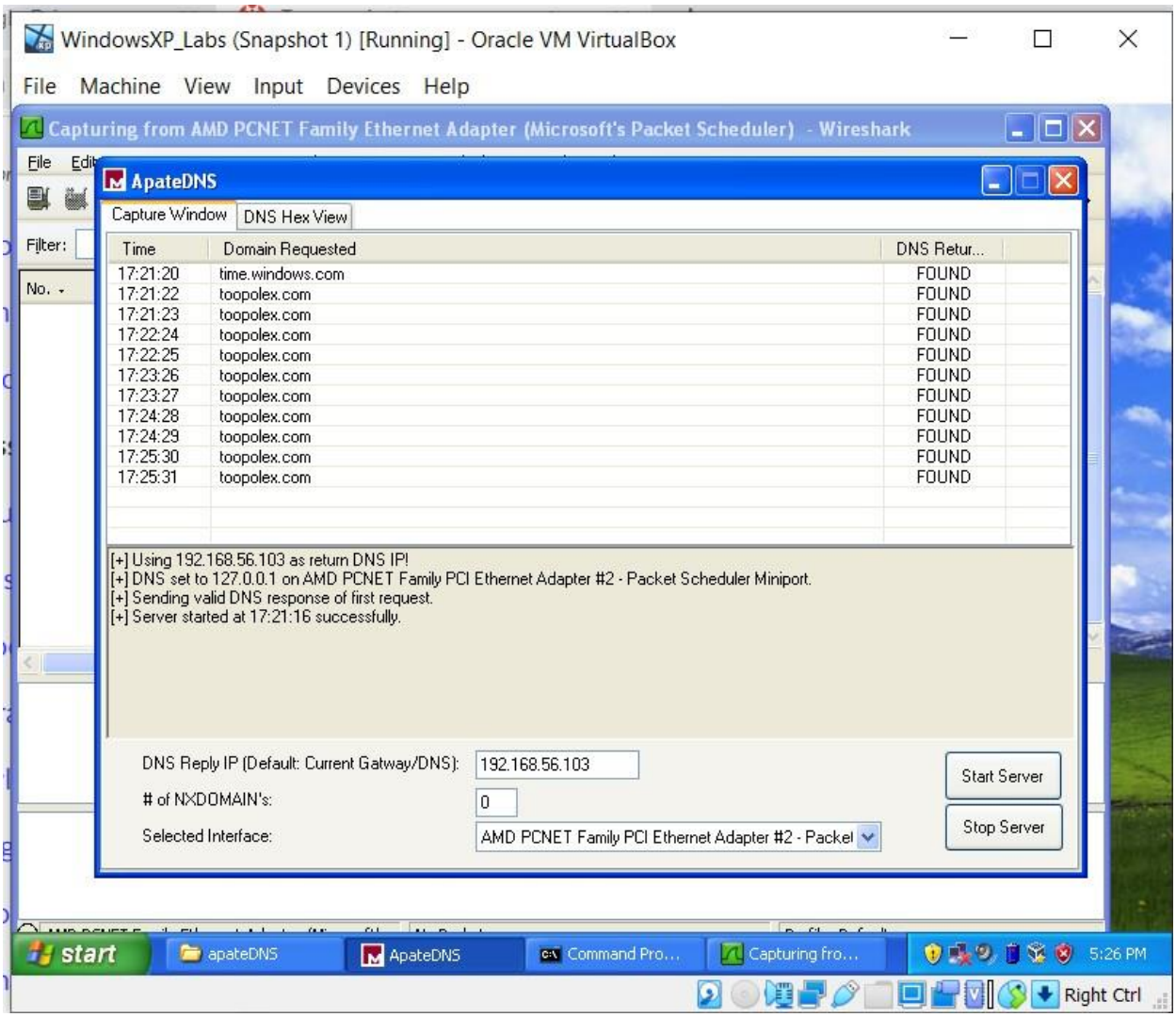





## Use of ApateDNS tool :

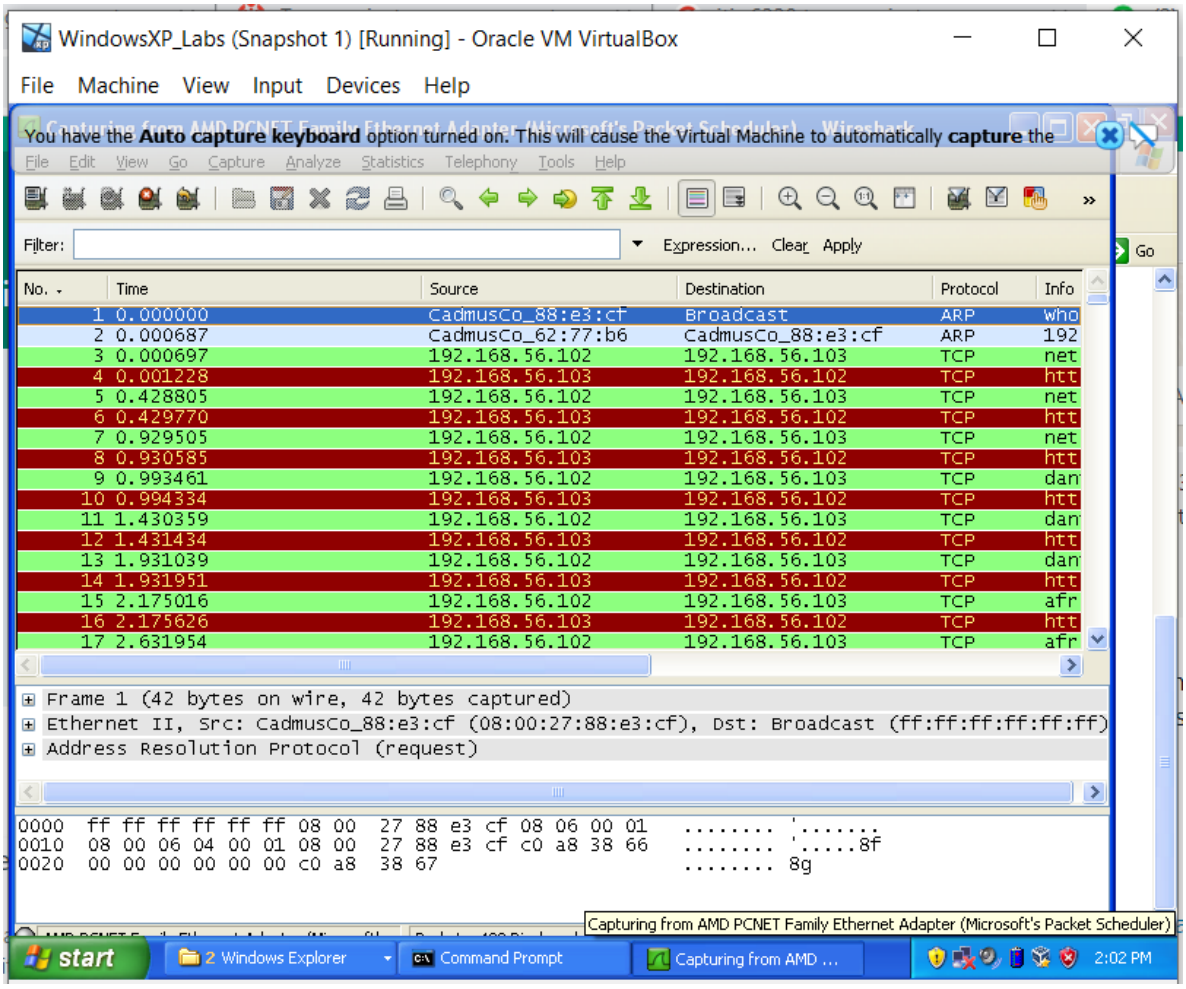
We enter the IP address of Ubuntu VM i.e., 192.168.56.103 and choose Host only Adapter and then start the server after running the malware:



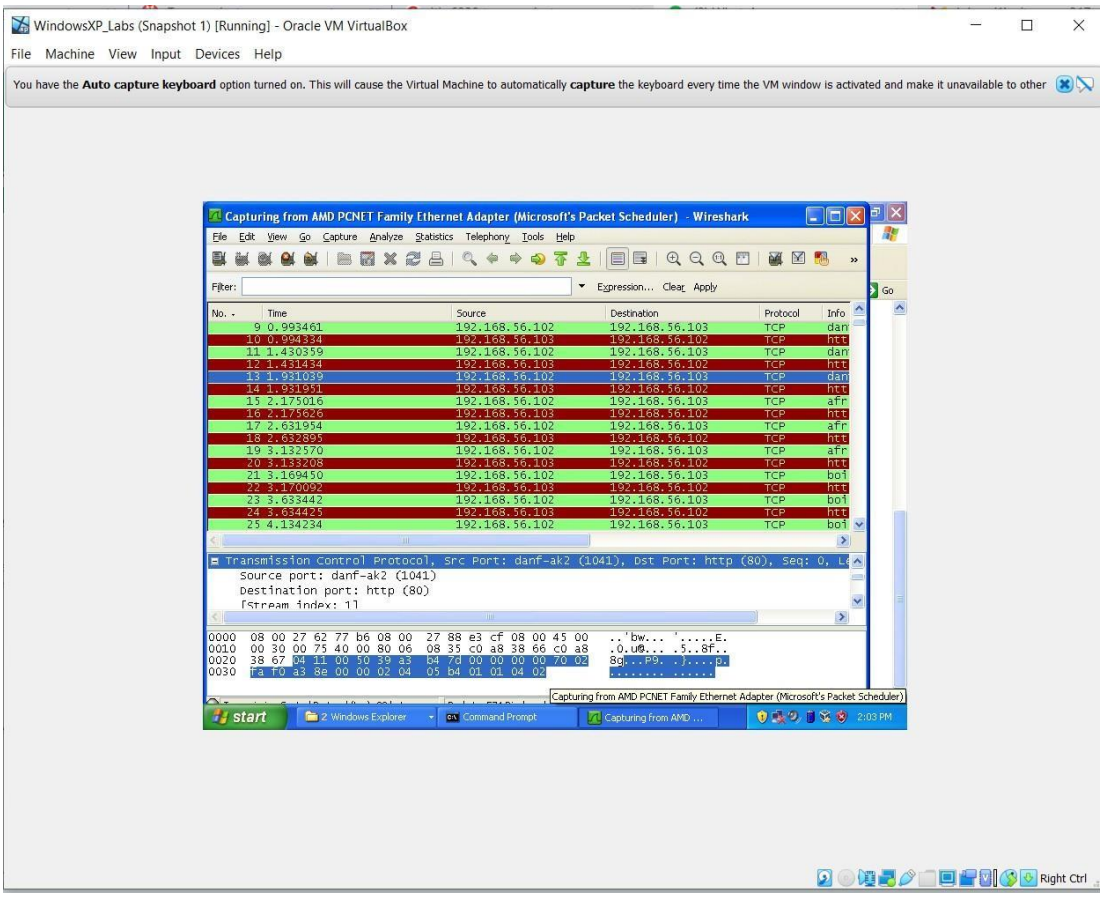

And we can see that malware is trying to reach the website: **toopolex.com**

## Use of Wireshark tool :

We use this tool to capture the traffic which malware is trying to reach the internet:



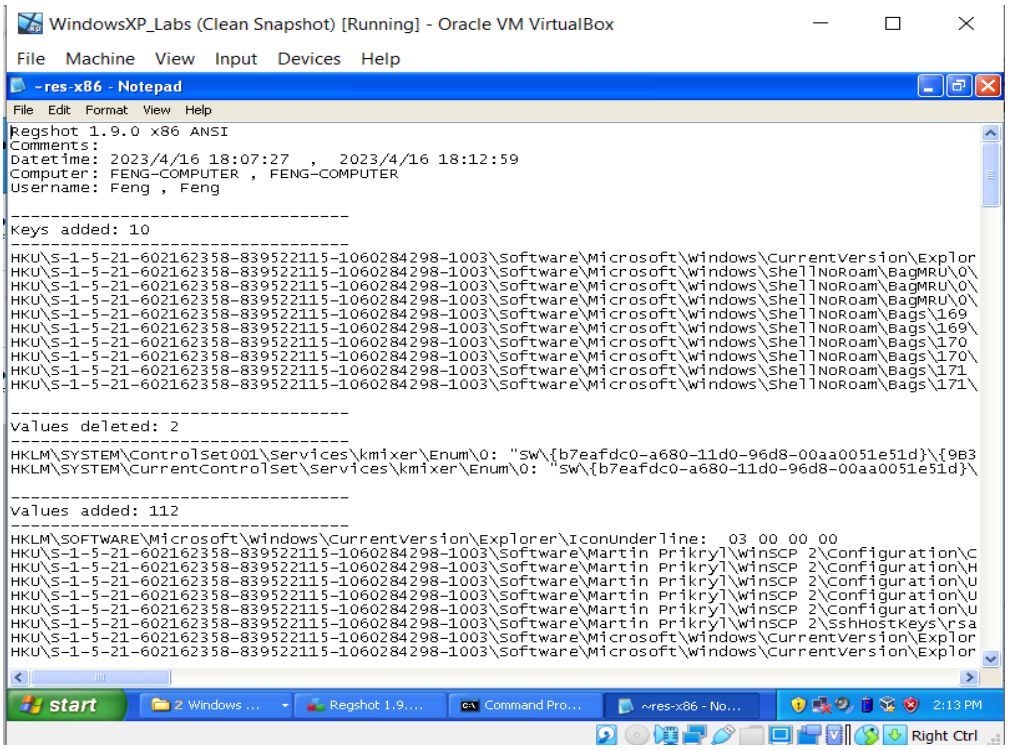

We can see that TCP traffic and the source and destination IP Addresses, this shows that it is trying to reach the internet and communicate through it.

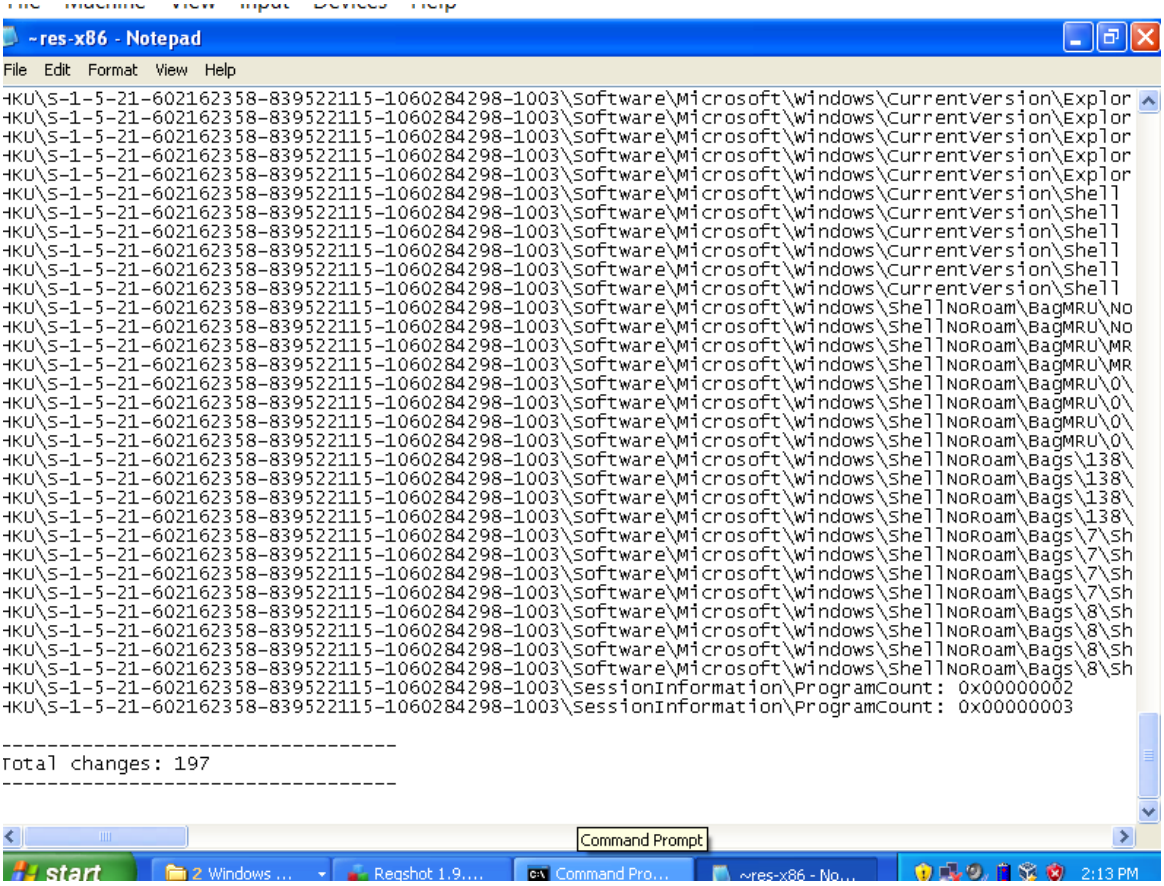

WindowsXP_Labs (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM window is activated and make it unavailable to other
Capturing from AMD PCNET Family Ethernet Adapter (Microsoft's Packet Scheduler) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
9 0.993461 192.168.56.102 192.168.56.103 TCP dan
10 0.994334 192.168.56.103 192.168.56.102 TCP htt
11 1.430359 192.168.56.102 192.168.56.103 TCP dan
12 1.431434 192.168.56.103 192.168.56.102 TCP htt
13 1.931039 192.168.56.102 192.168.56.103 TCP dan
14 1.931951 192.168.56.103 192.168.56.102 TCP htt
15 2.175016 192.168.56.102 192.168.56.103 TCP afr
16 2.175836 192.168.56.103 192.168.56.102 TCP htt
17 2.631954 192.168.56.102 192.168.56.103 TCP afr
18 2.632895 192.168.56.103 192.168.56.102 TCP htt
19 3.132570 192.168.56.102 192.168.56.103 TCP afr
20 3.133208 192.168.56.103 192.168.56.102 TCP htt
21 3.169450 192.168.56.102 192.168.56.103 TCP boi
22 3.170092 192.168.56.103 192.168.56.102 TCP htt
23 3.633442 192.168.56.102 192.168.56.103 TCP boi
24 3.634425 192.168.56.103 192.168.56.102 TCP htt
25 4.134234 192.168.56.102 192.168.56.103 TCP boi
Transmission Control Protocol, Src Port: danf-ak2 (1041), Dst Port: http (80), Seq: 0, L
Source port: danf-ak2 (1041)
Destination port: http (80)
[Stream index: 1]
0000 08 00 27 62 77 b6 08 00 27 88 e3 cf 08 00 45 00 ..'bw... '.....E.
0010 00 30 00 75 40 00 80 06 08 35 c0 a8 38 66 c0 a8 .0.u@... .5..8f..
0020 38 67 04 11 00 50 39 a3 b4 7d 00 00 00 00 70 02 8g...P9. .}....p.
0030 fa f0 a3 8e 00 00 02 04 05 b4 01 01 04 02 ........ ........
Capturing from AMD PCNET Family Ethernet Adapter (Microsoft's Packet Scheduler)
start 2 Windows Explorer Command Prompt Capturing from AMD ... 2:03 PM
Right Ctrl

## Use of RegShot Tool :

We have performed the RegShot Analysis, in which we take two shots one before running malware and the other after running. We see the total changes made to the Registry files.

**Total Changes :** 197





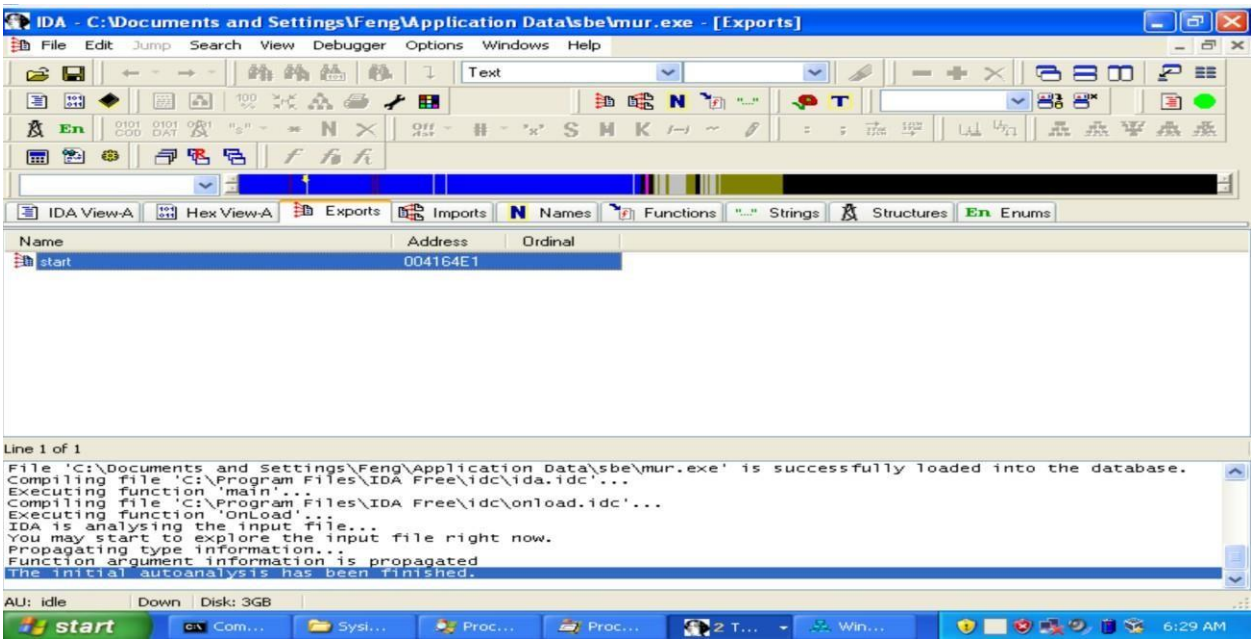

Now we have to perform advanced analysis of both the behavior analysis in static as well as dynamic analysis.

## Advanced Static Analysis on malware behavior :

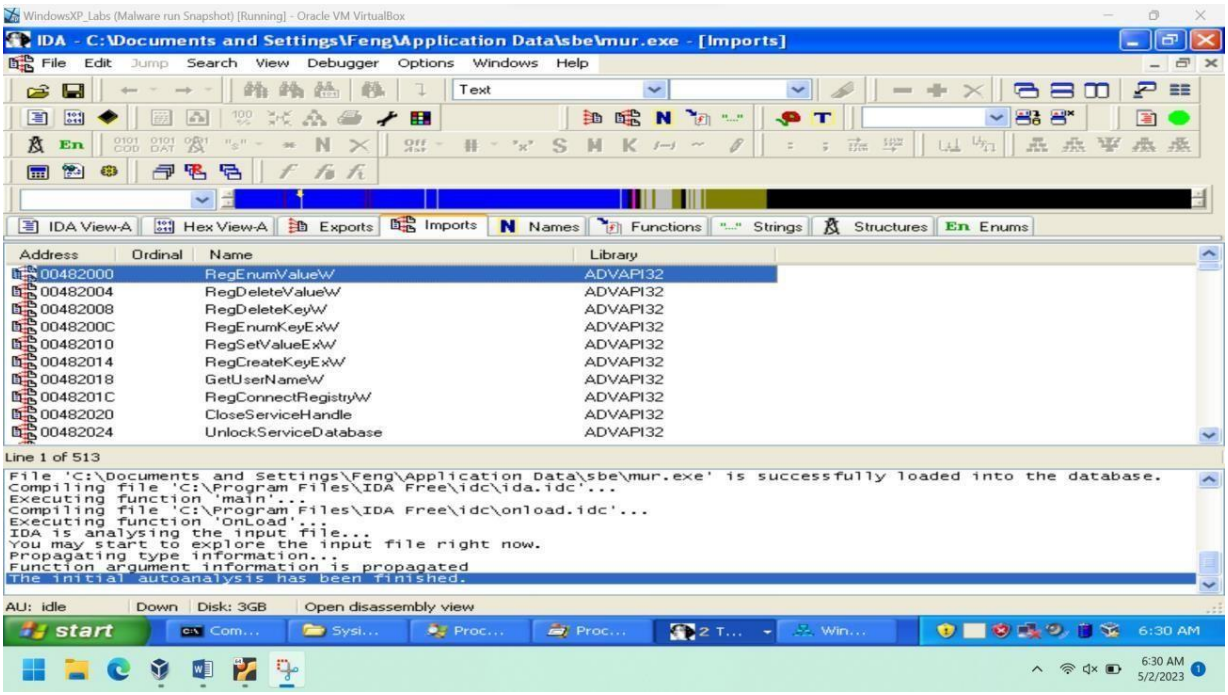
In this section, we use some advanced tools for the static analysis such as IDA Pro and Ghidra software tools. These tools are used for disassembling the malware either in assembly language or C language or both.

## Use of IDA Pro tool :

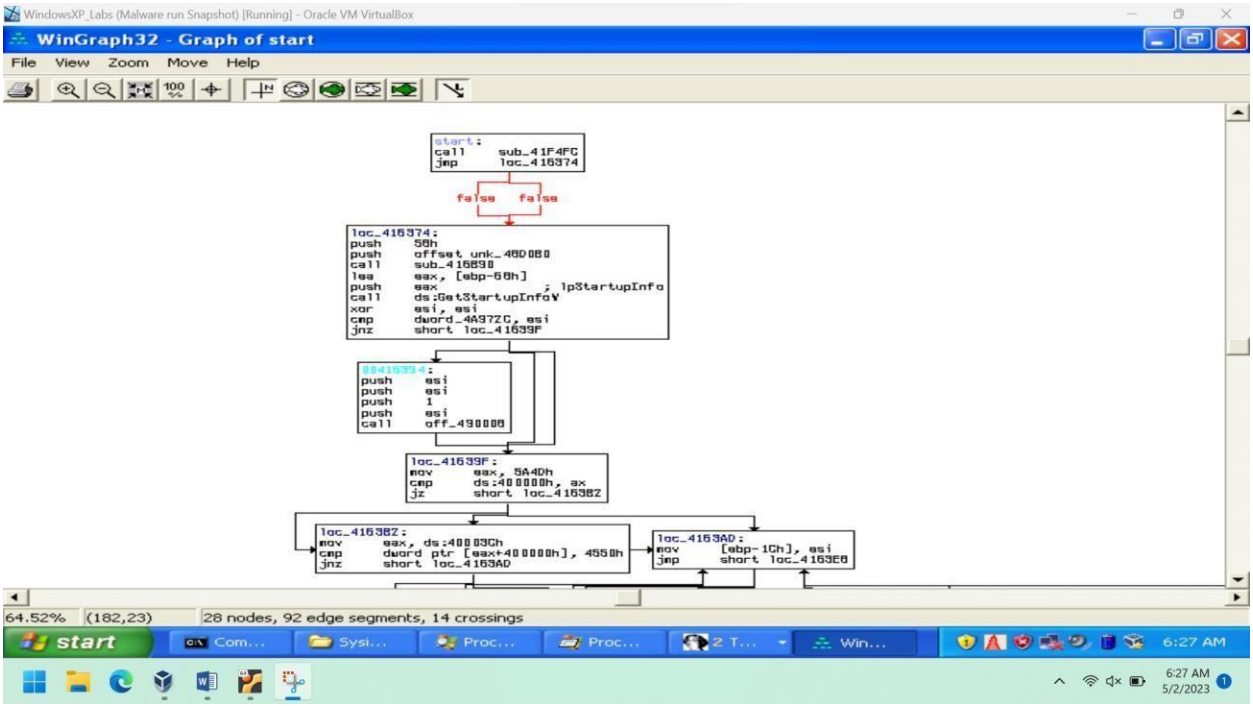
Now we see list of Exports :

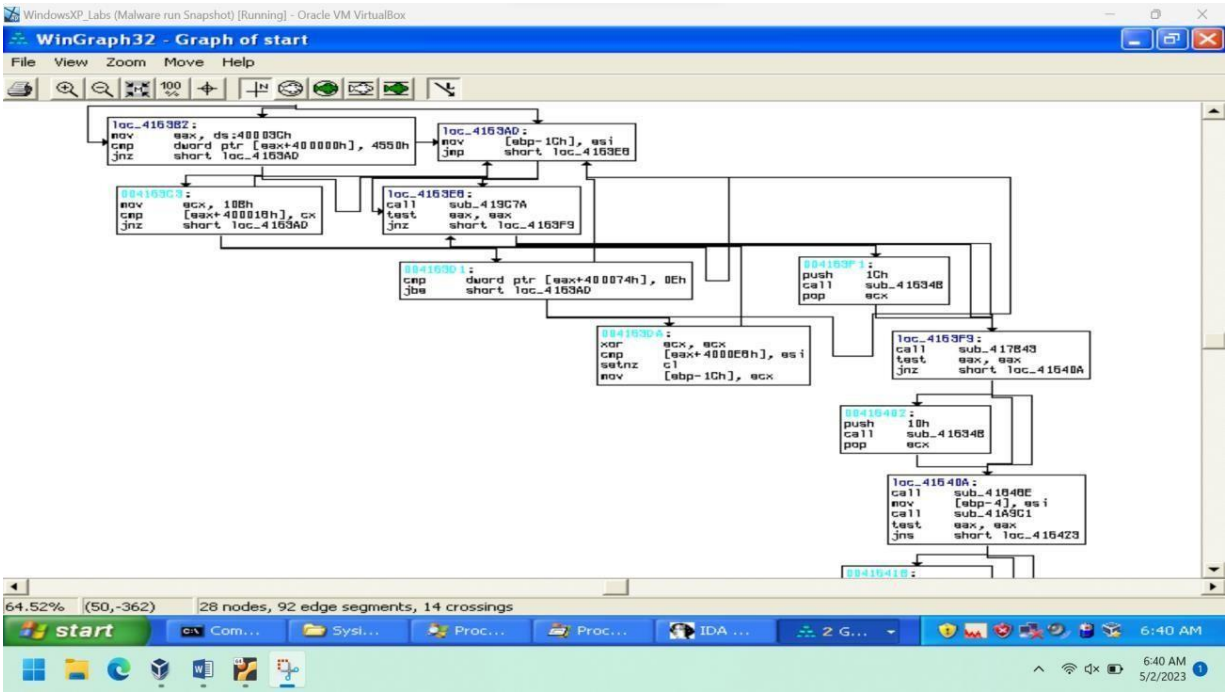


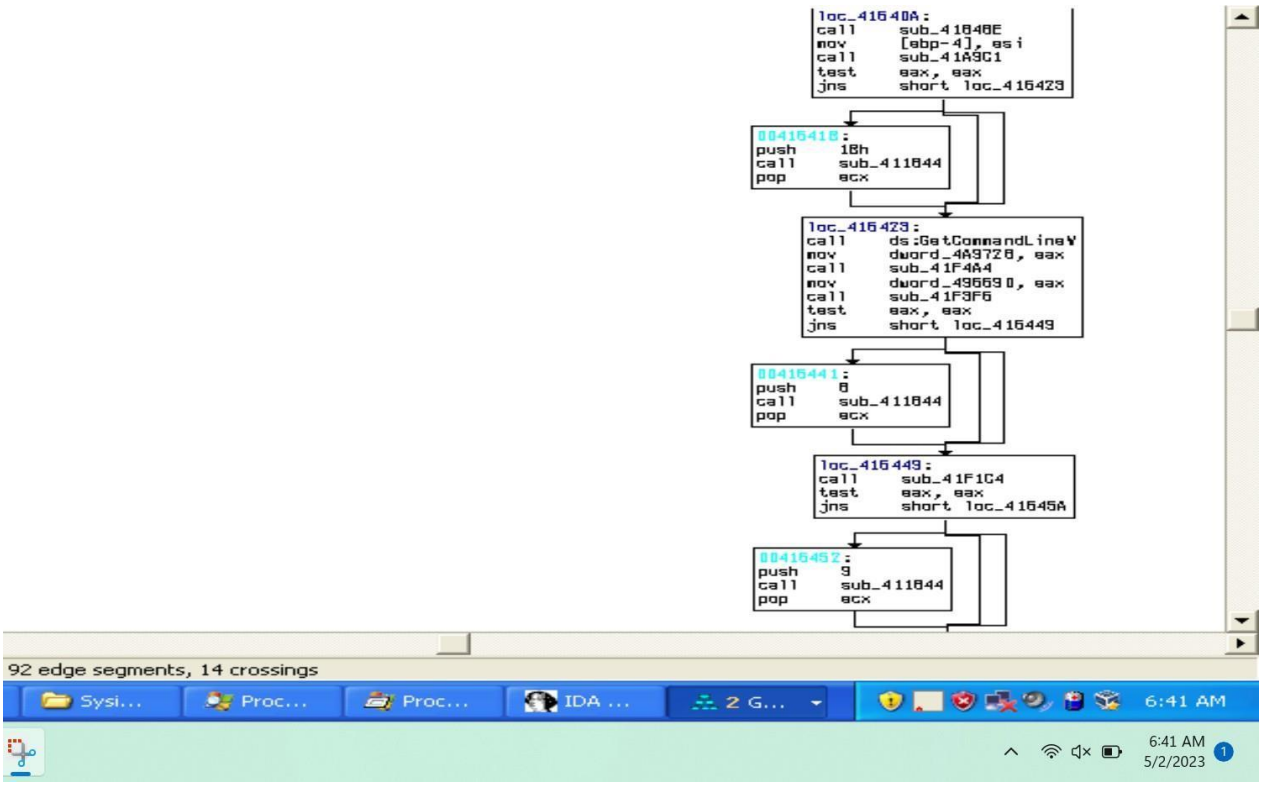

Now the list of imports does the malware in assessing it's behavior :

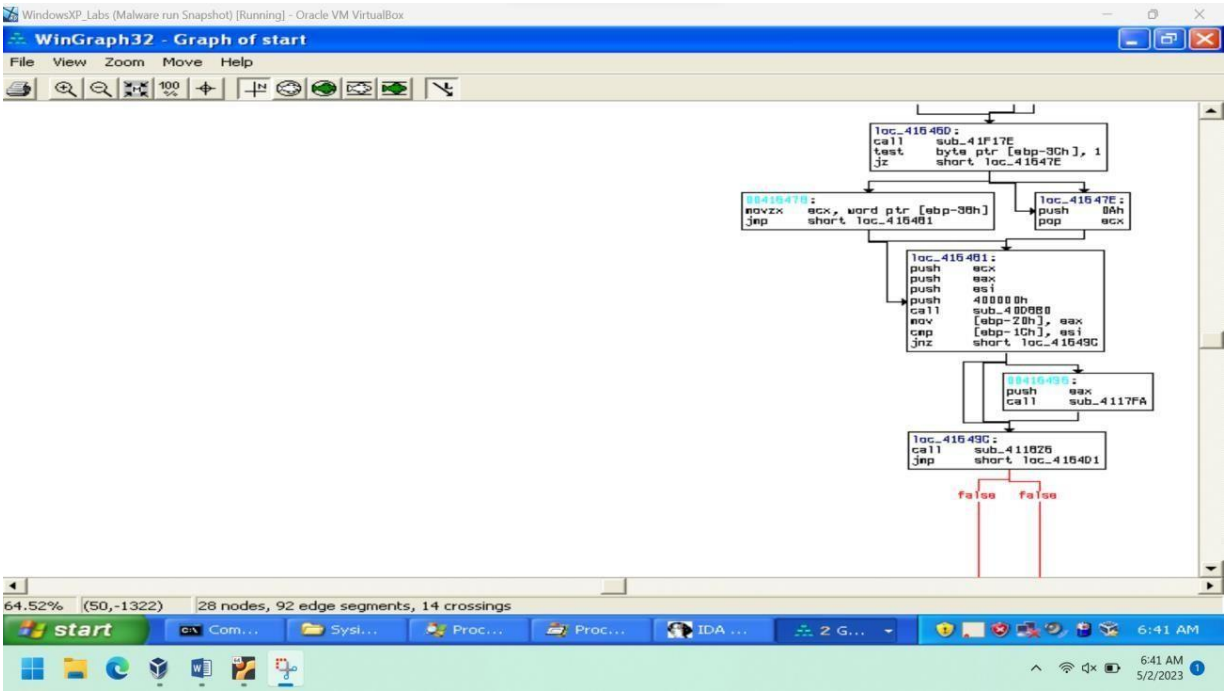


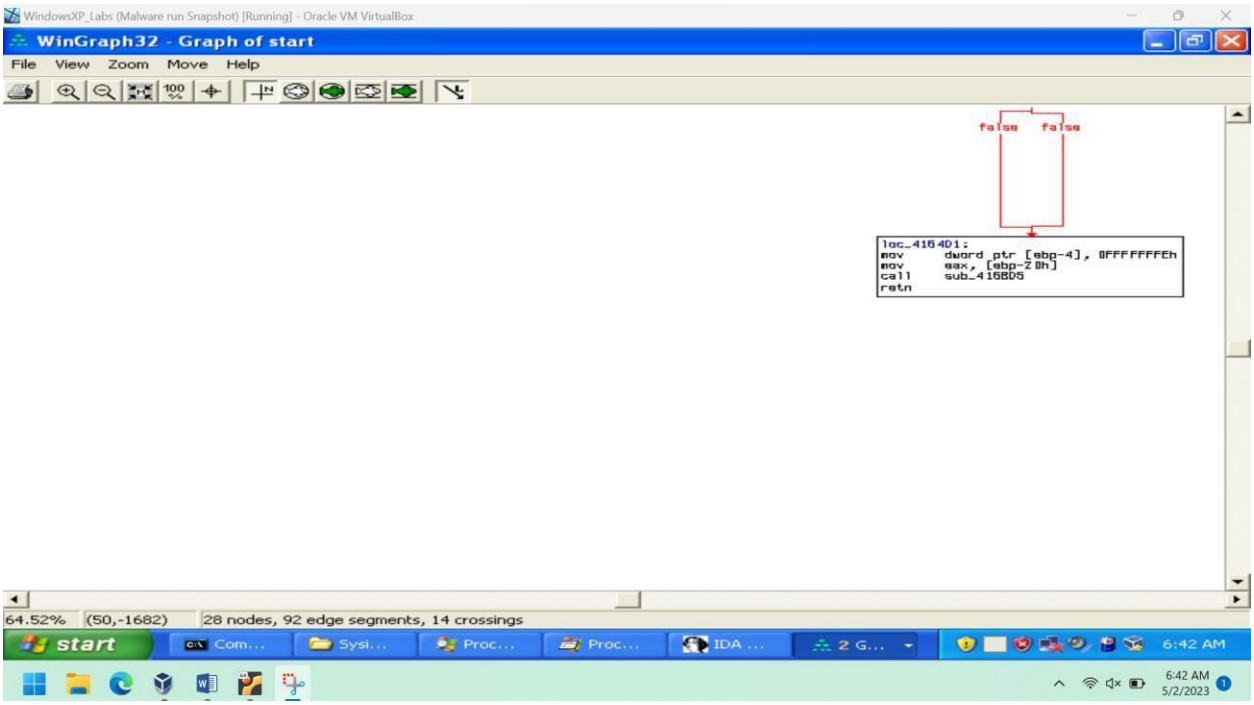
Taking WinGraph32 – Graph of start



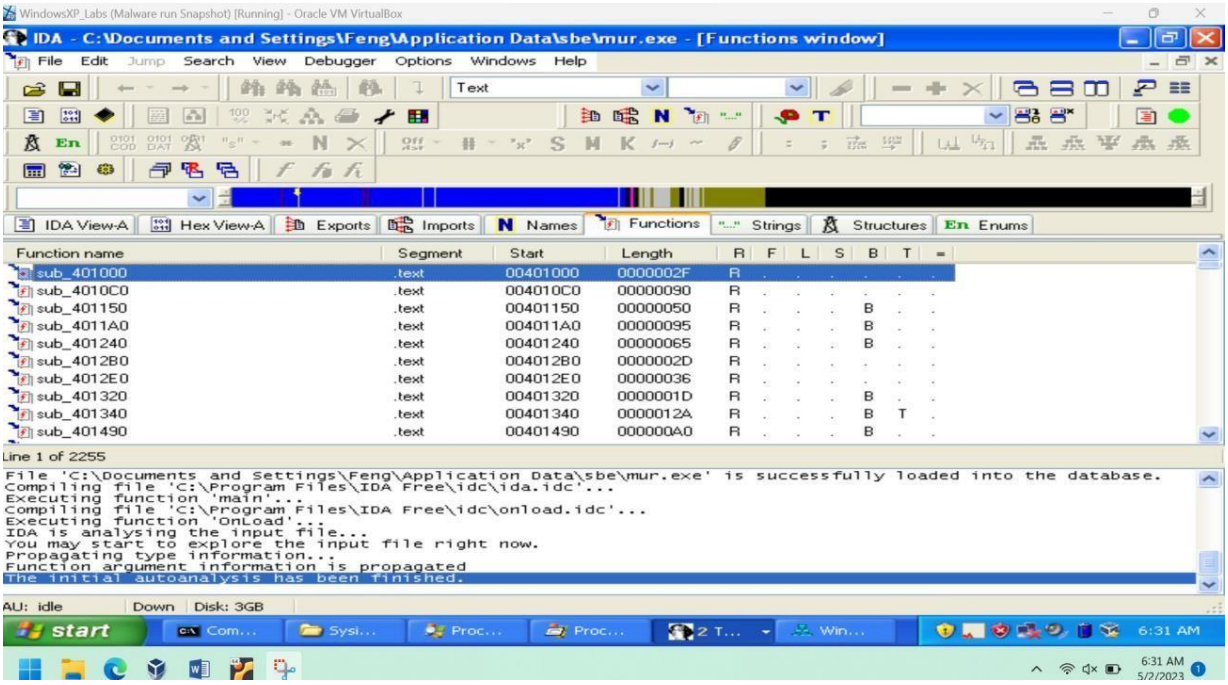



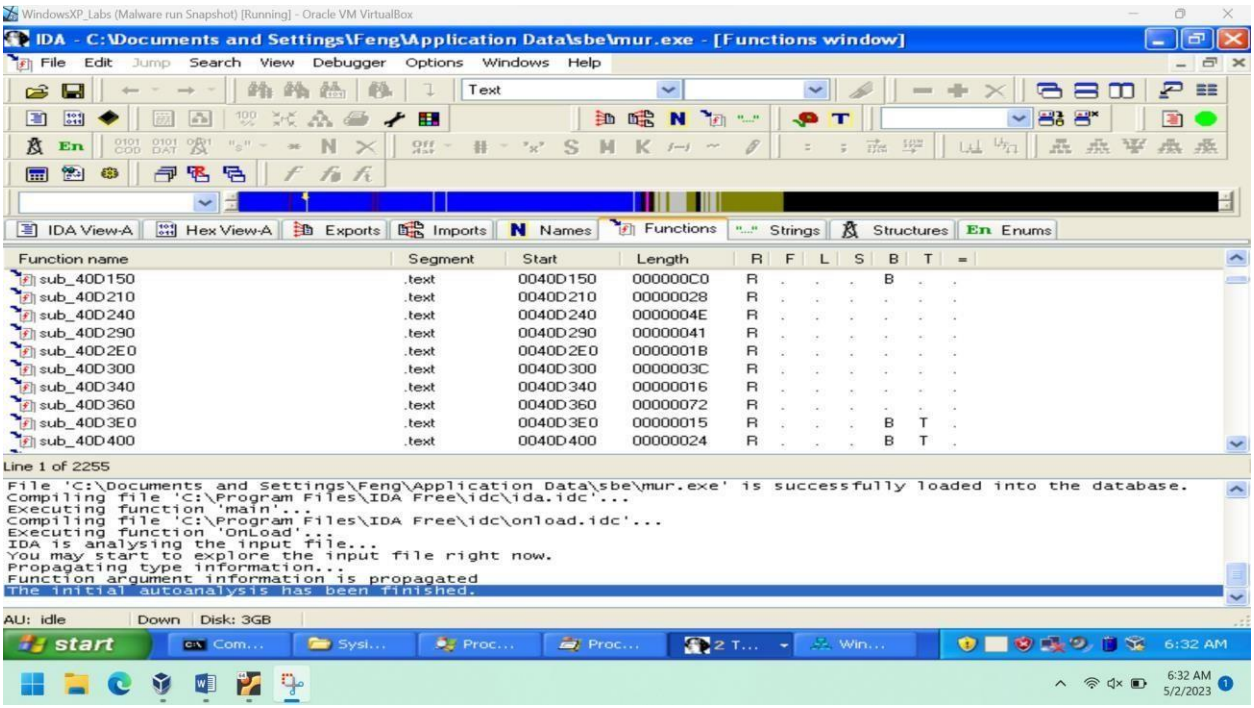

loc_41640A:
call sub_41646E
mov [ebp-4], esi
call sub_41A9C1
test eax, eax
jns short loc_416423
00416418:
push 18h
call sub_411844
pop ecx
loc_416423:
call ds:GetCommandLineW
mov dword_4A97Z8, eax
call sub_41F4A4
mov dword_495690, eax
call sub_41F3F6
test eax, eax
jns short loc_416449
00416441:
push 8
call sub_411844
pop ecx
loc_416449:
call sub_41F1C4
test eax, eax
jns short loc_41645A
00416452:
push 9
call sub_411844
pop ecx
92 edge segments, 14 crossings
Sysi...
Proc...
Proc...
IDA ...
2 G...
6:41 AM
6:41 AM
5/2/2023


WindowsXP_Labs (Malware run Snapshot) [Running] - Oracle VM VirtualBox
WinGraph32 - Graph of start
File View Zoom Move Help
loc_41646D:
call sub_41F17E
test byte ptr [ebp-3Ch], 1
jz short loc_41647E
00416478:
movzx ecx, word ptr [ebp-38h]
jmp short loc_416481
loc_41647E:
push 0Ah
pop ecx
loc_416481:
push ecx
push eax
push esi
push 400000h
call sub_400880
mov [ebp-20h], eax
cmp [ebp-1Ch], esi
jnz short loc_41649C
00416495:
push eax
call sub_4117FA
loc_41649C:
call sub_411826
jmp short loc_4164D1
false
false
64.52%
(50,-1322)
28 nodes, 92 edge segments, 14 crossings
start
Com...
Sysi...
Proc...
Proc...
IDA ...
2 G...
6:41 AM
6:41 AM
5/2/2023

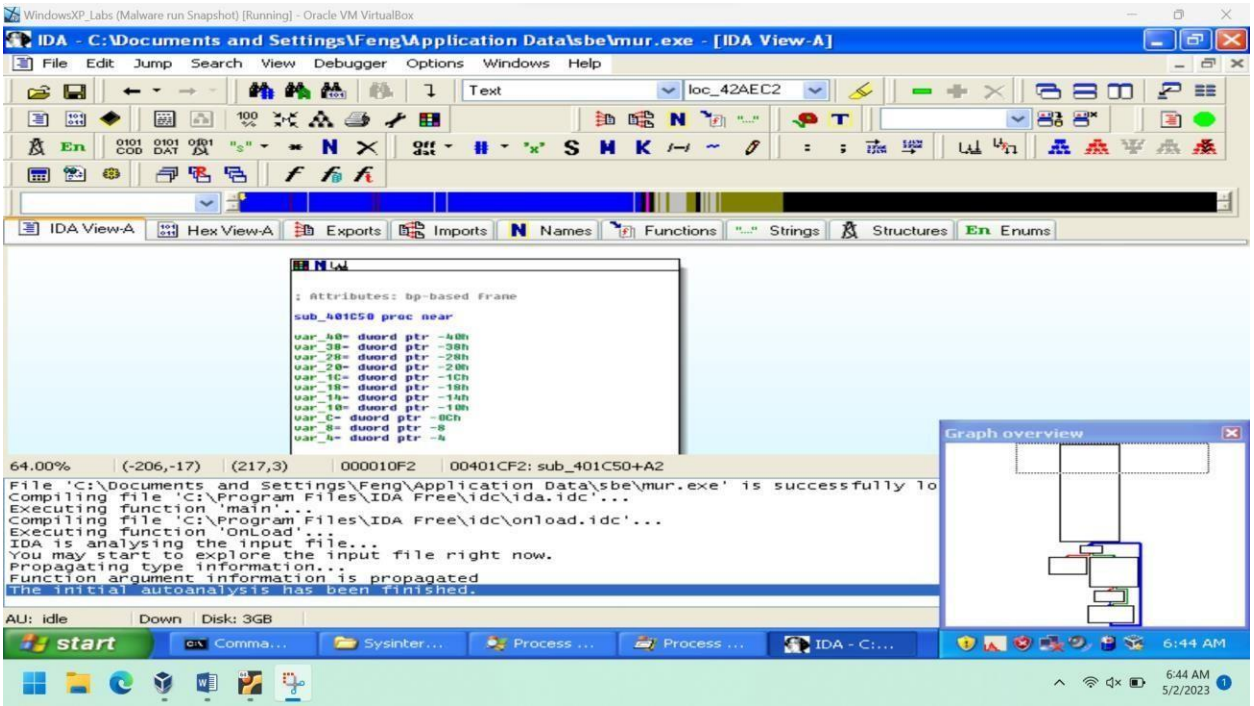



The functions associated in the analysis of mur.exe :

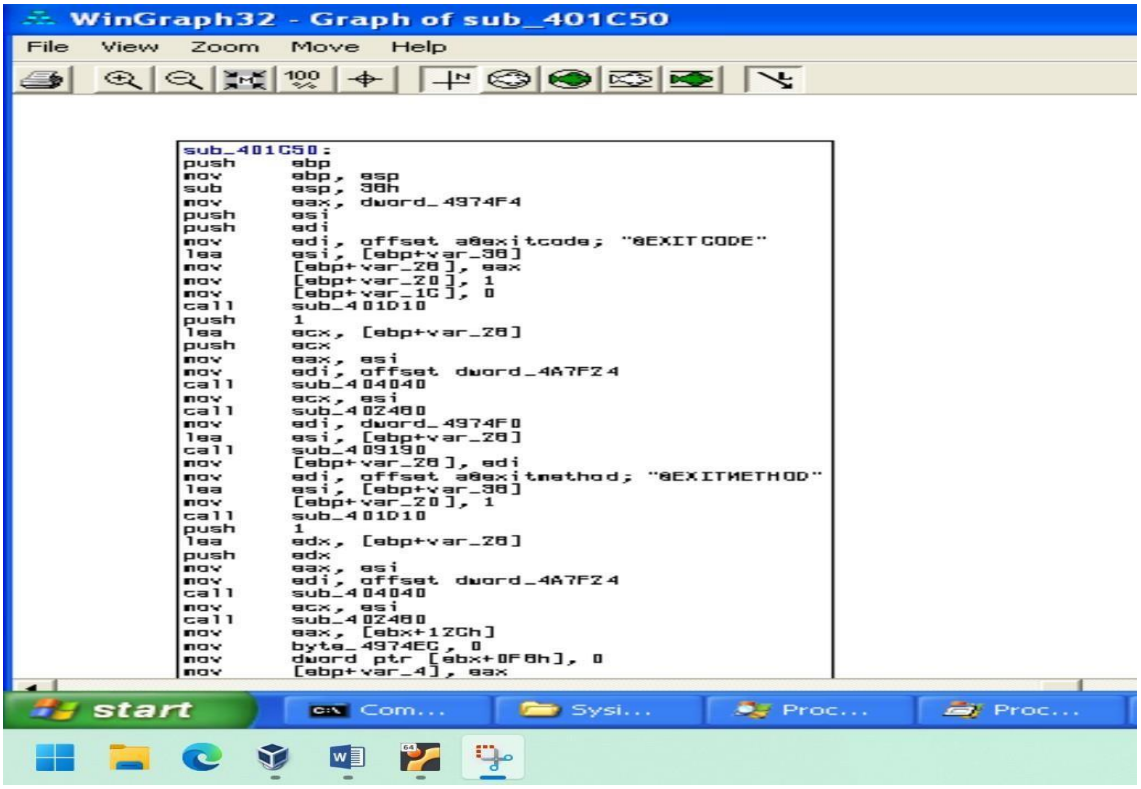


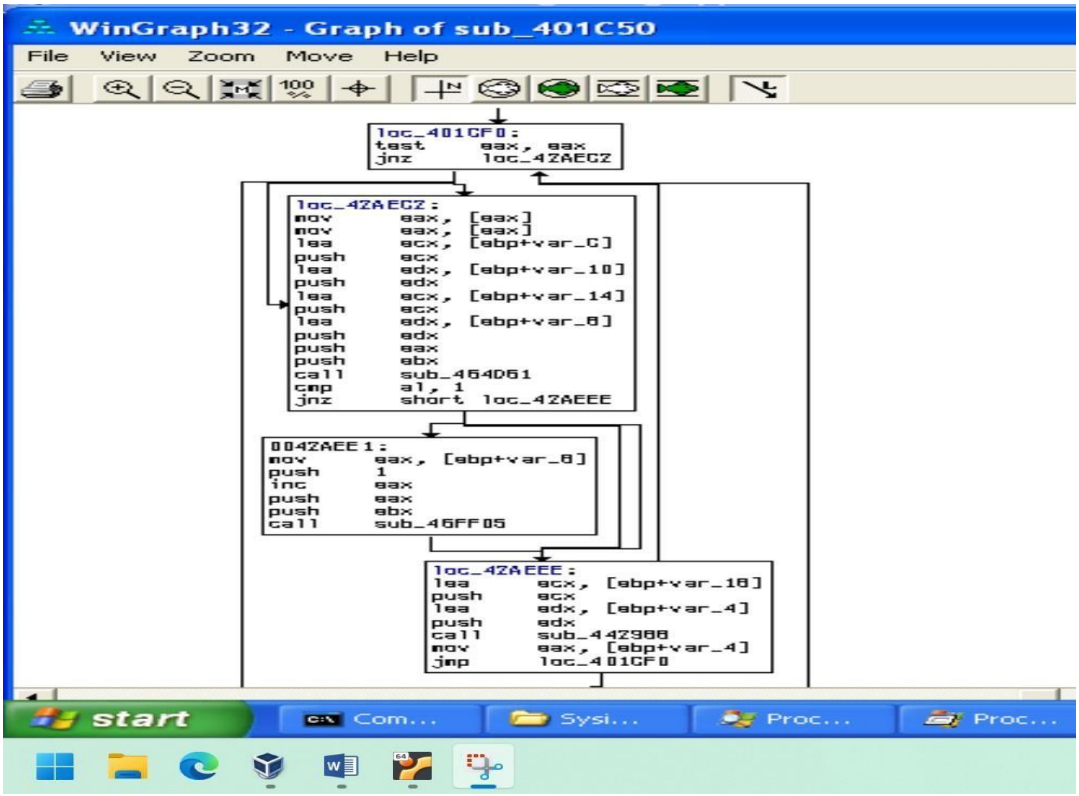


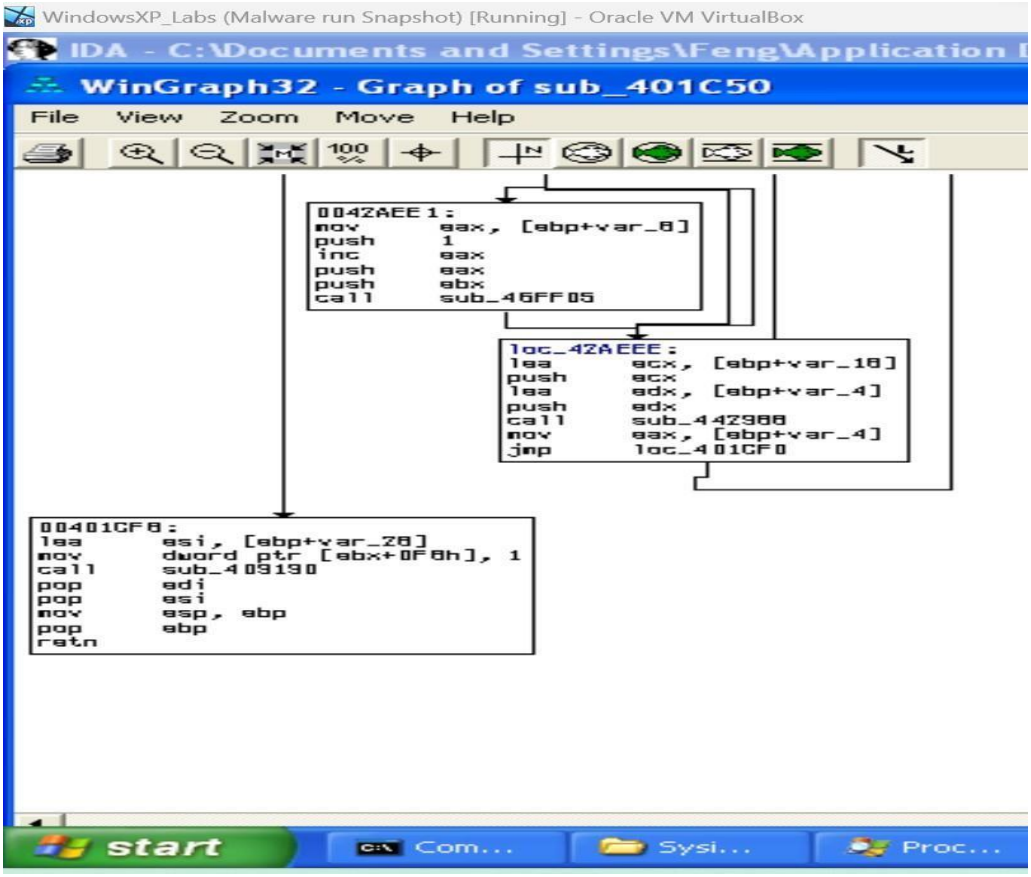

One of the function sub_401C50 :




And it’s WinGraph32 :


WinGraph32 - Graph of sub_401C50
File View Zoom Move Help
sub_401C50:
push ebp
mov ebp, esp
sub esp, 30h
mov eax, dword_4974F4
push esi
push edi
mov edi, offset a0exitcode; "0EXITCODE"
lea esi, [ebp+var_30]
mov [ebp+var_28], eax
mov [ebp+var_20], 1
mov [ebp+var_1C], 0
call sub_401D10
push 1
lea ecx, [ebp+var_28]
push ecx
mov eax, esi
mov edi, offset dword_4A7FZ4
call sub_404040
mov ecx, esi
call sub_402480
mov edi, dword_4974F0
lea esi, [ebp+var_28]
call sub_409190
mov [ebp+var_28], edi
mov edi, offset a0exitmethod; "0EXITMETHOD"
lea esi, [ebp+var_30]
mov [ebp+var_20], 1
call sub_401D10
push 1
lea edx, [ebp+var_28]
push edx
mov eax, esi
mov edi, offset dword_4A7FZ4
call sub_404040
mov ecx, esi
call sub_402480
mov eax, [ebx+1ZGh]
mov byte_4974EC, 0
mov dword ptr [ebx+0F8h], 0
mov [ebp+var_4], eax
start
Com...
Sysi...
Proc...
Proc...

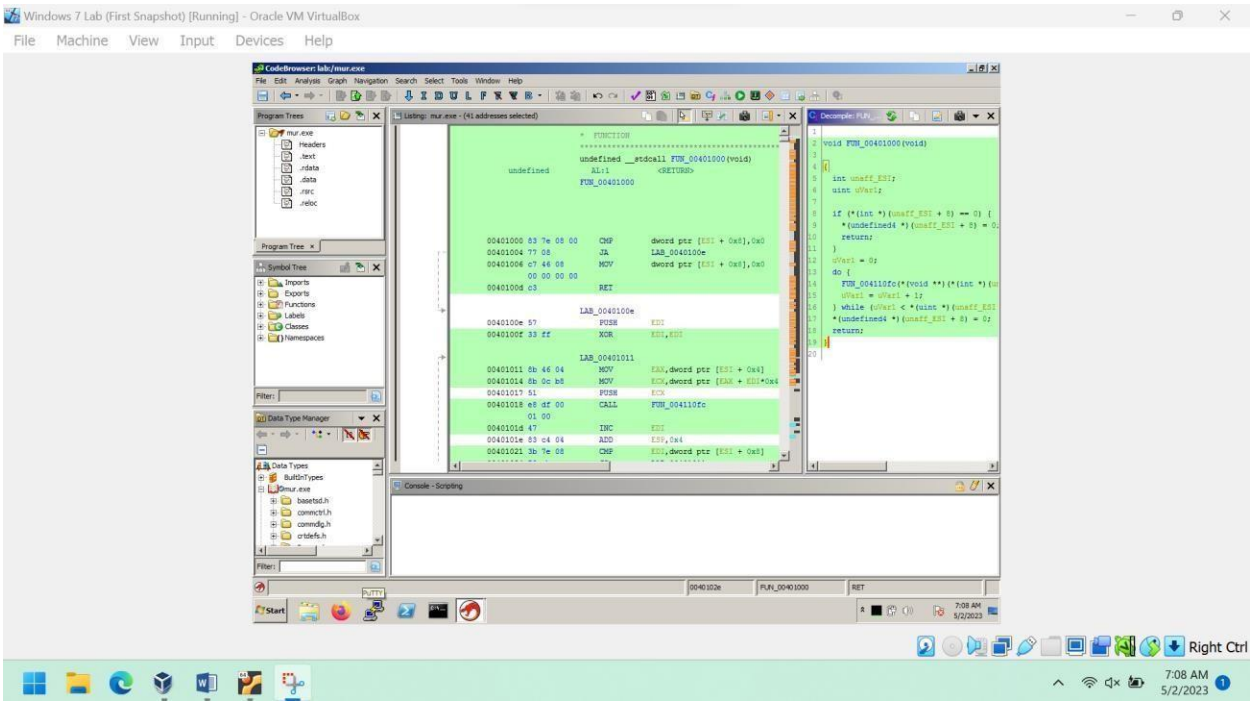

WinGraph32 - Graph of sub_401C50
File View Zoom Move Help
loc_401CF0:
test eax, eax
jnz loc_4ZAECZ
loc_4ZAECZ:
mov eax, [eax]
mov eax, [eax]
lea ecx, [ebp+var_C]
push ecx
lea edx, [ebp+var_10]
push edx
lea ecx, [ebp+var_14]
push ecx
lea edx, [ebp+var_8]
push edx
push eax
push ebx
call sub_464D61
cmp al, 1
jnz short loc_4ZAEEE
0042AEE1:
mov eax, [ebp+var_8]
push 1
inc eax
push eax
push ebx
call sub_46FF05
loc_4ZAEEE:
lea ecx, [ebp+var_18]
push ecx
lea edx, [ebp+var_4]
push edx
call sub_44Z980
mov eax, [ebp+var_4]
jmp loc_401CF0
start
Com...
Sysi...
Proc...
Proc...



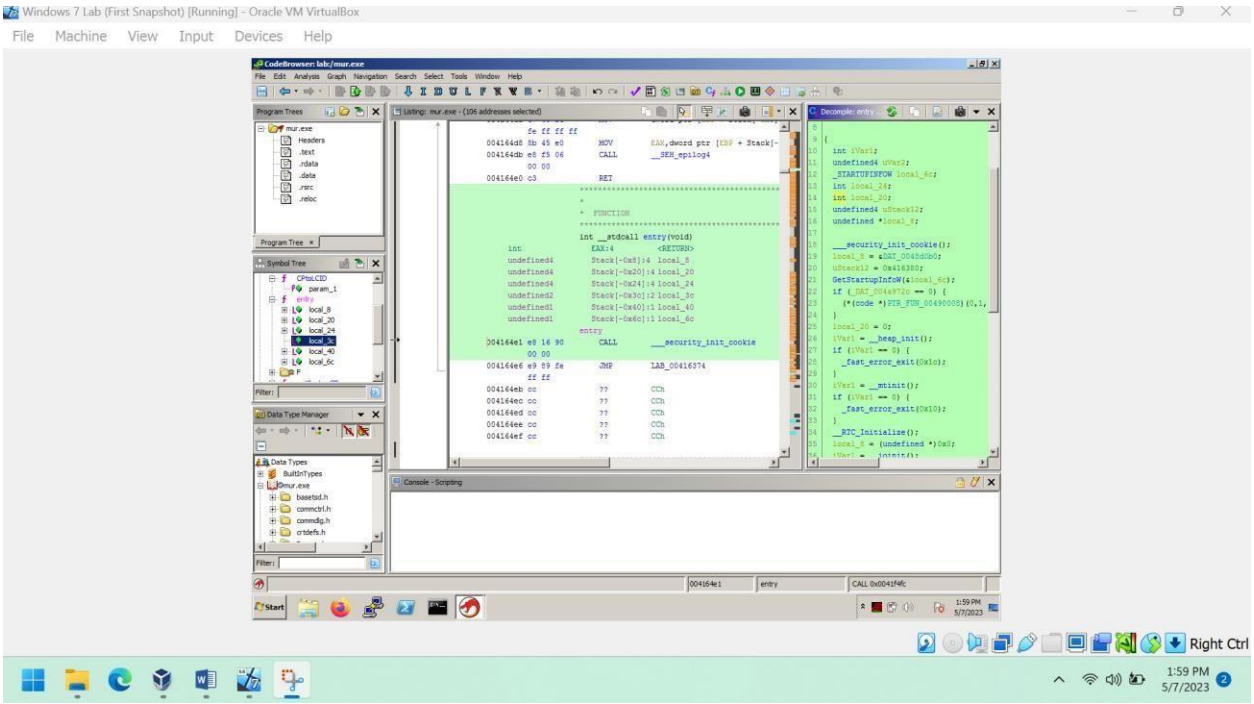

These are the findings I have found so far with the help of IDA Pro tool and now we see the results using Ghidra tool.

## Use of Ghidra Software tool :

This tool shows the disassembly in both the assembly language and C language in the CodeViewer browser. As you can see in the following figure it is clearly evident that we can somehow understand in C language rather than Assembly language which is the best feature of Ghidra tool.



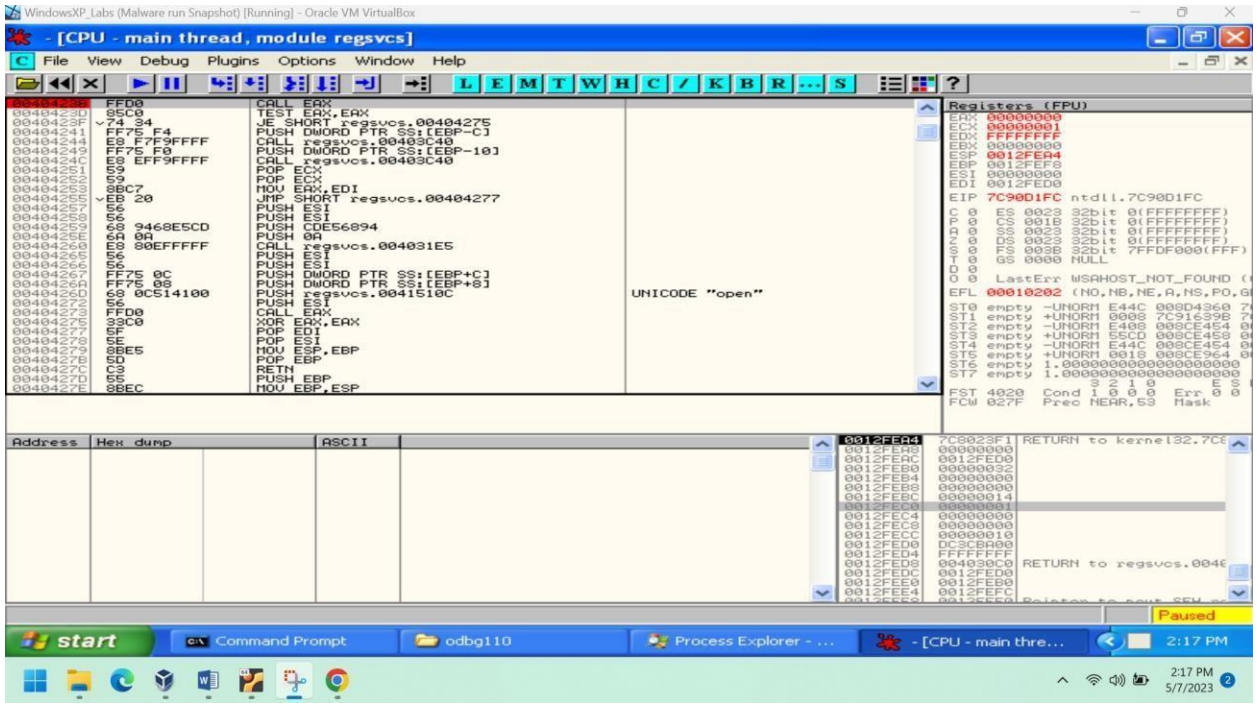

In this you can see C language constructs on the right side in the decompiler section.

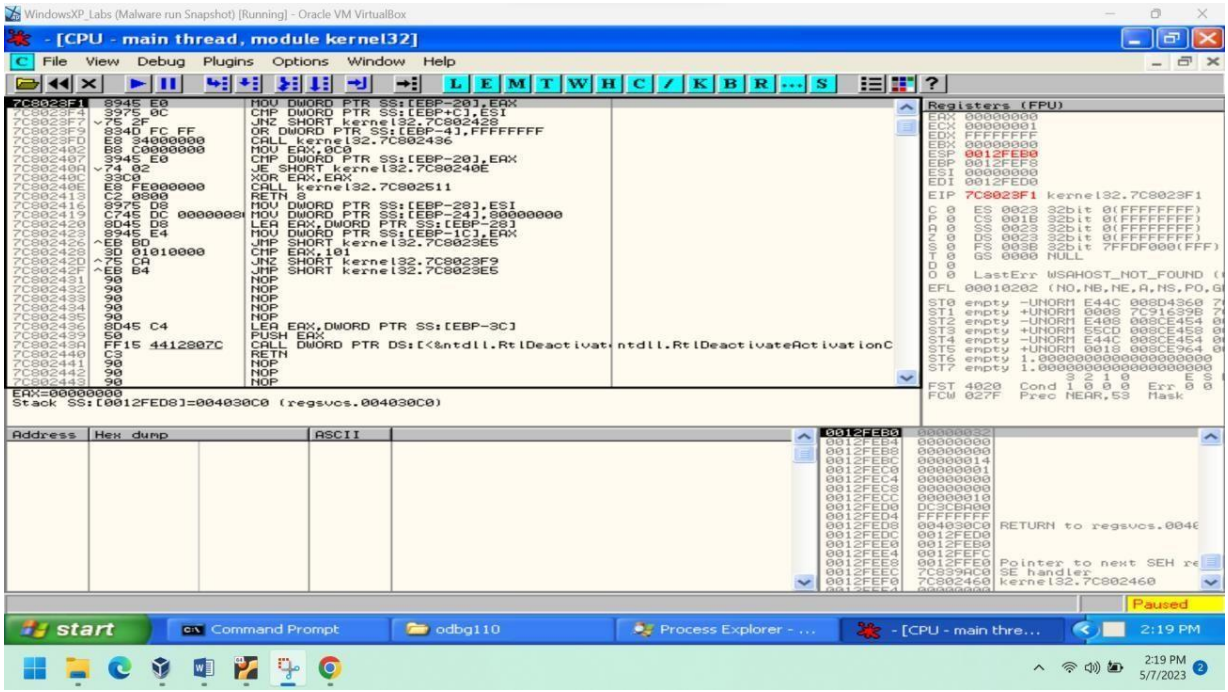


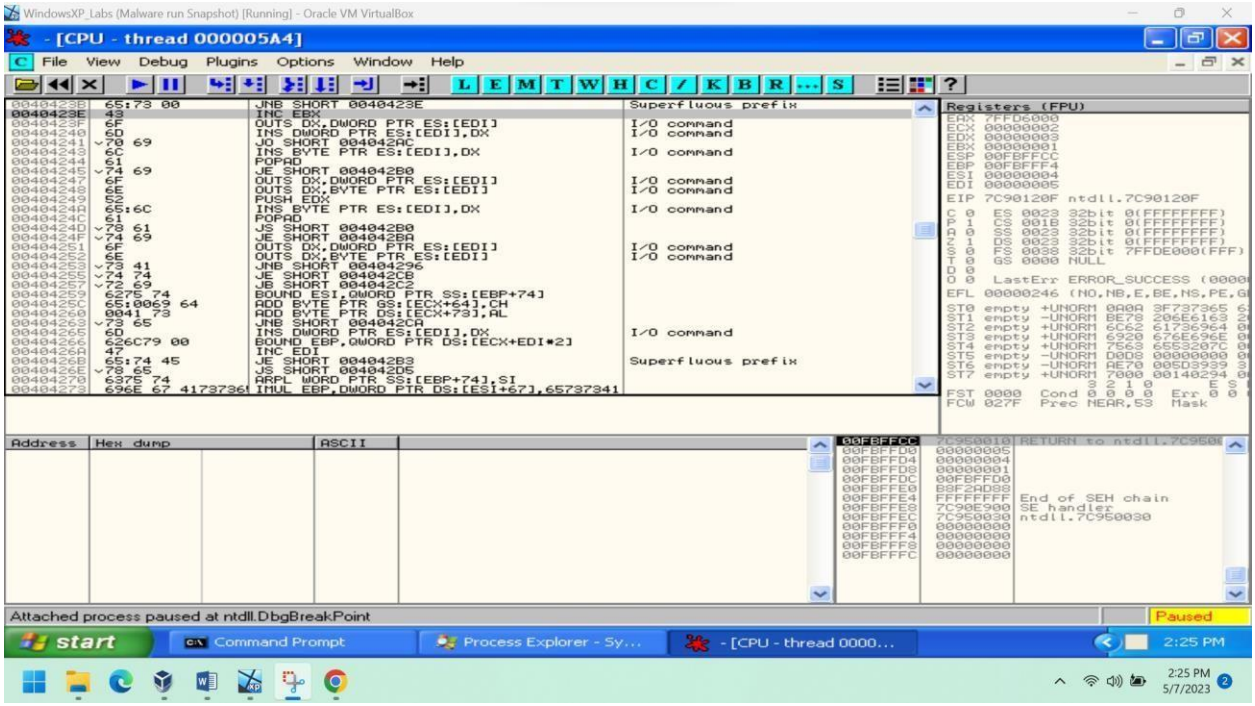

This is another function where we can see that it is working with some cookies related and also we can see some manipulations. There are few other important functions which we can see on the left side in the Functions under symbol tree.

Now we see the analysis performed using OllyDbg 1.10 tool as following as below

**Use of Ollydbg 1.10 analysis :**




WindowsXP_Labs (Malware run Snapshot) [Running] - Oracle VM VirtualBox
- [CPU - thread 000005A4]
File View Debug Plugins Options Window Help
Registers (FPU)
Address Hex dump ASCII
Attached process paused at ntdll.DbgBreakPoint
Paused
start
Command Prompt
Process Explorer - Sy...
- [CPU - thread 0000...
2:25 PM
5/7/2023

**3. Reflections :**

During the course of the project, I had gained a lot of knowledge and my takeaways in this project to analyze the malware and get some real time hands-on experience in the way how to analyze, reverse engineer the code, how the malware installs on the system, how can we identify it. I had performed various roles like testing the malware, reverse engineer it, how systems can be protected against malware. It was my responsibility to use programs like PEiD, upx, PEview, CFF Explorer, and other static analysis tools to learn more about the features and architecture of the malware. I also contributed to the documentation and report-writing phase, where my job was to explain the results of the static analysis in great detail, providing information on headers, file formats, and perhaps obfuscated code.

I used a wide range of technical expertise for this endeavor, especially in the areas of malware analysis, reverse engineering, and cybersecurity. My knowledge in MS Cybersecurity program topics like "Malware Analysis" was quite helpful. I was able to utilize the numerous malware analysis tools and methodologies that I gained a strong understanding of from these classes to this project. In addition, the network security and digital forensics courses I took gave me a strong basis for understanding the complex facets of malware behavior, including command and control, data exfiltration, and evasion strategies.

This experience has led me to pursue new research directions in malware analysis and cybersecurity. I came to understand how critical it is to keep up with the most recent malware trends and analysis tools in order to continue being productive in this ever changing industry. I'm also inspired to investigate more sophisticated reverse engineering methods and resources, as well as go further into threat intelligence and behavioral analysis. My dedication to lifelong learning and cybersecurity research has been strengthened by this project, and I can't wait to use these newfound insights in my future work as a cybersecurity professional.

The project's scope includes anti-debugging, methods of system evasion, process spawning, and important malware functionalities. It also entails looking into keylogging capabilities, encryption, exfiltrated data, and domain names. The final study will prove to be a useful tool for researchers and cybersecurity professionals in comprehending the malware's functioning.

### 3.1 Challenges:

There were a number of difficulties ran into while working on this project. A significant obstacle was the intricacy of the malware's multi-phase operation. A thorough examination of the malware's behavior and code was necessary to comprehend the complexities of how it changed over time and interacted with outside parties. Finding the malware's anti-debugging and sandbox evasion strategies required a great deal of investigation and testing, which made the job difficult.

### 3.2 Learnings:

My learning goals for this assignment were to improve my report-writing skills, strengthen my grasp of malware analysis methods, and obtain a thorough understanding of reverse engineering. Using IDA Pro and Ghidra, among other tools, I learnt how to maneuver through complex code structures during the project and honed my ability to spot evasion techniques. Through the preparation of the analytical report, the project also enhanced my capacity to communicate complicated technical findings in an accessible and straightforward manner.