Lab 3: Perform Static Malware Analysis

Lab Scenario

Attackers use sophisticated malware techniques as cyber weapons to steal sensitive data. Malware can inflict intellectual and financial losses on the target, be it an individual, a group of people, or an organization. The worst part is that it spreads from one system to another with ease and stealth.

Malware such as viruses, Trojans, worms, spyware, and rootkits allow an attacker to breach security defenses and subsequently launch attacks on target systems. Thus, to find and cure the existing infections and thwart future problems, it is necessary to perform malware analysis. Many tools and techniques exist to perform such tasks. Malware analysis provides an in-depth understanding of each individual sample and identifies emerging technology trends from large collections of malware samples without executing them. The samples of malware are mostly compatible with the Windows binary executable.

By performing malware analysis, detailed information regarding the malware can be extracted. This information includes items like the malicious intent of the malware, indicators of compromise, complexity level of the intruder, exploited vulnerability, extent of damage caused by the intrusion, perpetrator accountable for installing the malware, and system vulnerability the malware has exploited. An ethical hacker and pen tester must perform malware analysis to understand the workings of the malware and assess the damage that it may cause to the information system. Malware analysis is an integral part of any penetration testing process.

It is very dangerous to analyze malware on production devices connected to production networks. Therefore, one should always analyze malware samples in a testing environment on an isolated network.

Lab Objectives

- Perform malware scanning using Hybrid Analysis
- Analyze ELF executable file using Detect It Easy (DIE)
- Perform malware disassembly using IDA and OllyDbg

Overview of Static Malware Analysis

Static Malware Analysis, also known as code analysis, involves going through the executable binary code without executing it to gain a better understanding of the malware and its purpose. The process includes the use of different tools and techniques to determine the malicious part of the program or a file. It also gathers information about malware functionality and collects the technical pointers or simple signatures it generates. Such pointers include file name, MD5 checksums or hashes, file type, and file size. Analyzing the binary code provides information about the malware's functionality, network signatures, exploit packaging technique, dependencies involved, as well as other information.

Some of the static malware analysis techniques are:

- File fingerprinting
- Local and online malware scanning

- Performing strings search
- Identifying packing and obfuscation methods
- Finding portable executable (PE) information
- Identifying file dependencies
- Malware disassembly

Task 1: Perform Malware Scanning using Hybrid Analysis

Hybrid Analysis is a free service that analyzes suspicious files and URLs and facilitates the quick detection of unknown threats such as viruses, worms, Trojans, and other kinds of malware.

It helps ethical hackers and penetration testers to examine files and URLs, enabling the identification of viruses, worms, Trojans, and other malicious content detected by anti-virus engines and website scanners.

This task will demonstrate how to analyze malware using online Hybrid Analysis services.

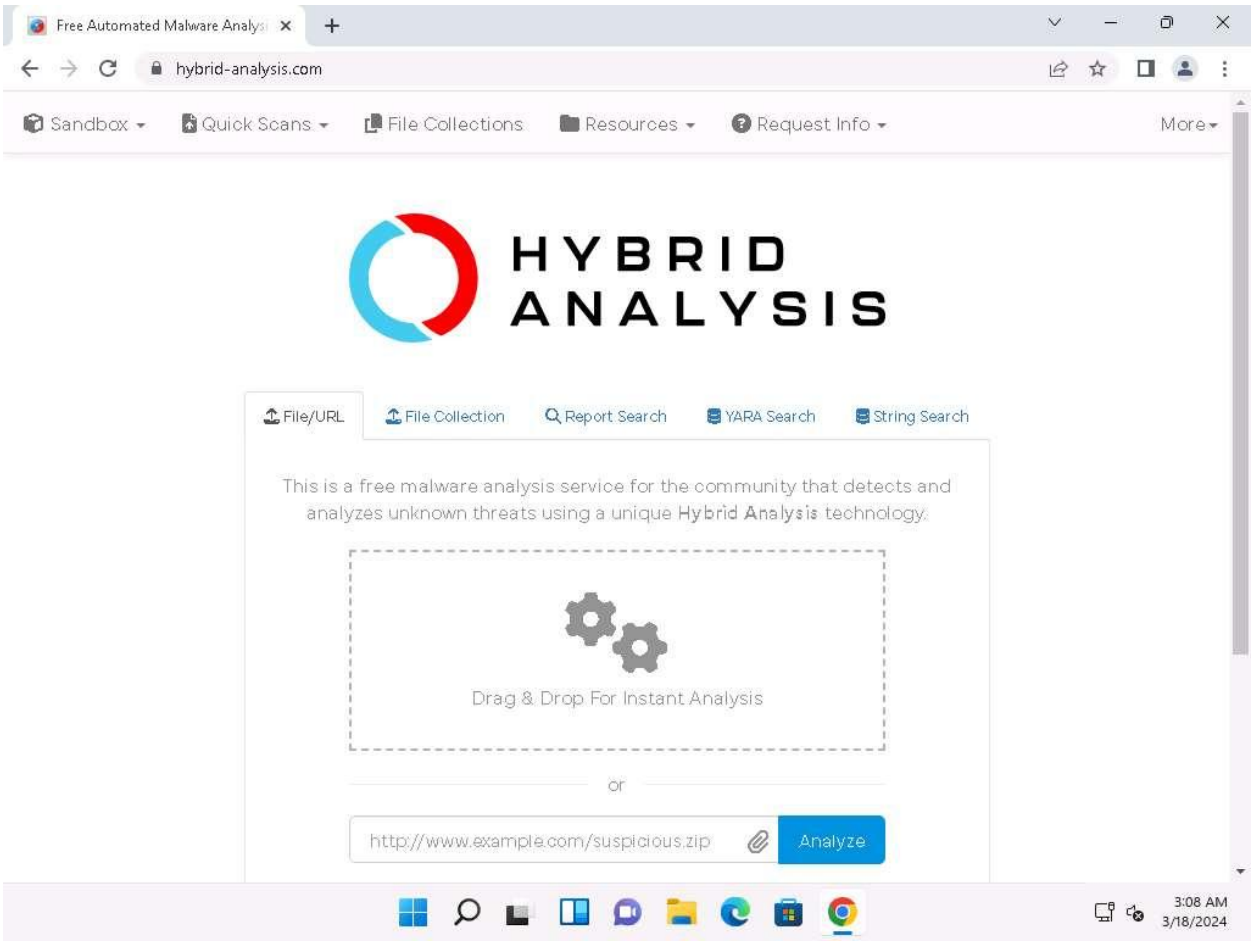
1. By default, **Windows 11** machine selected, click Ctrl+Alt+Delete. Login with **Admin/Pa$$w0rd**.

Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.

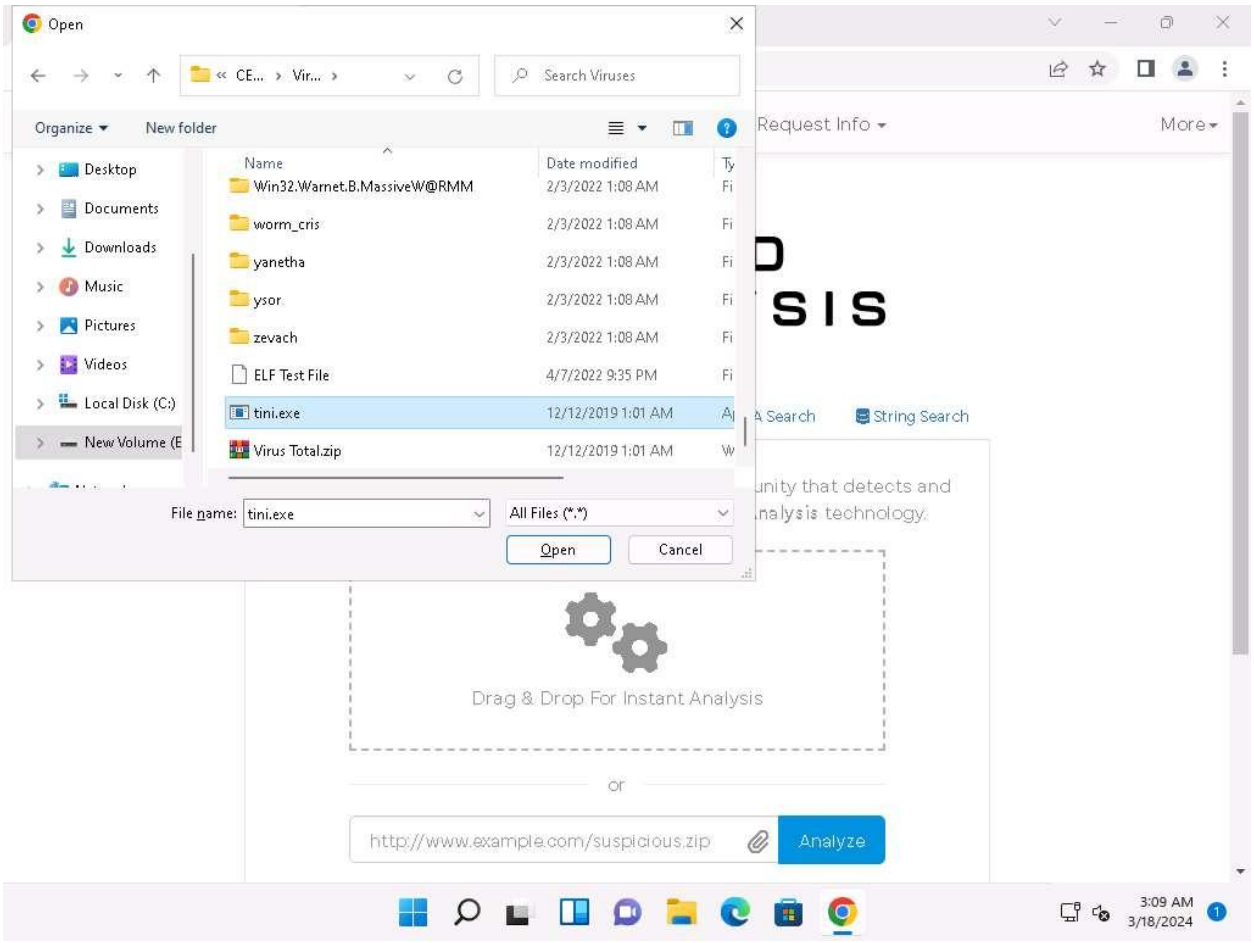
2. Open any web browser (here, **Google Chrome**) and go to **https://www.hybrid-analysis.com** and press **Enter**.

If a cookie notification appears in the lower section of the page, then click **ACCEPT**.

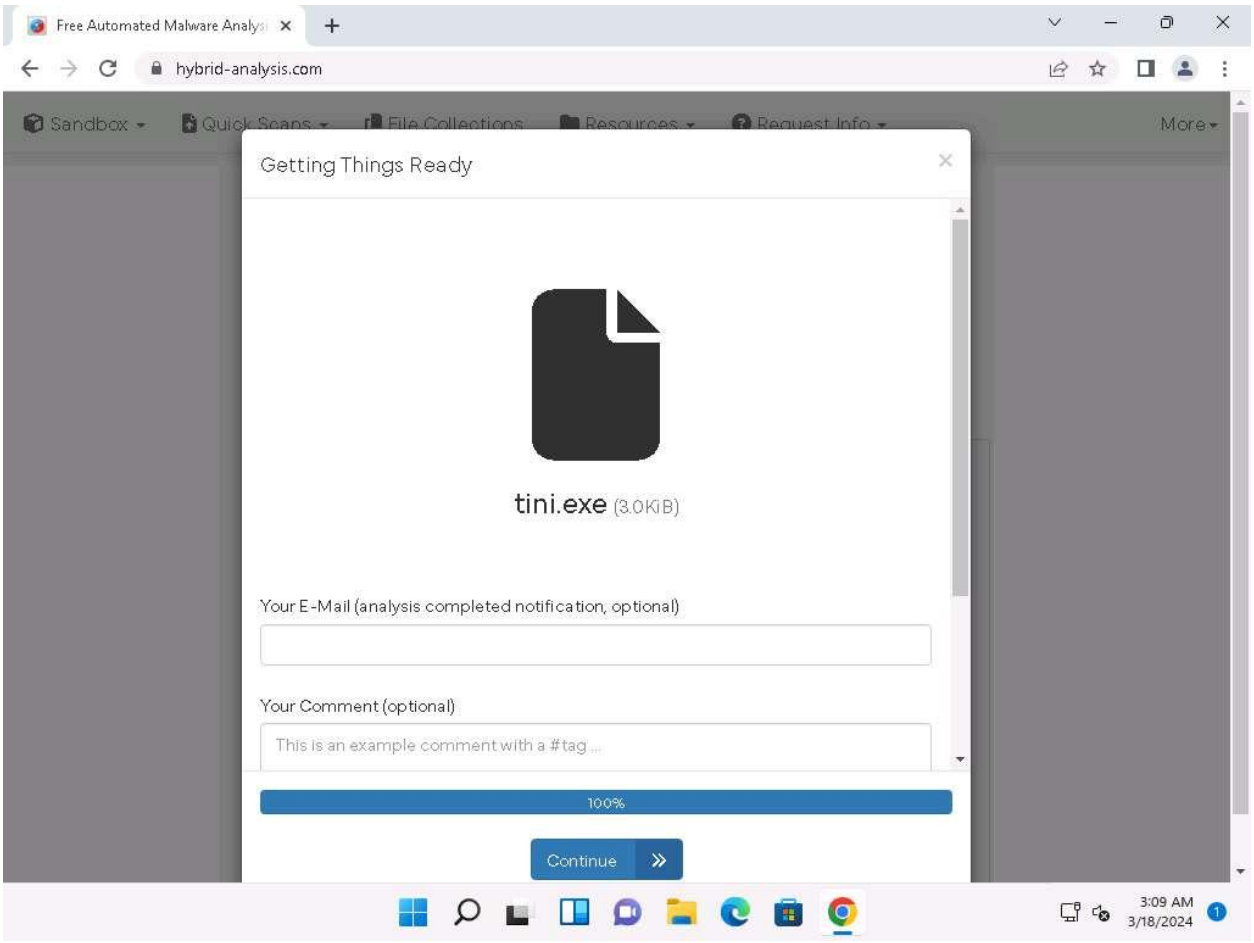
3. The **HYBRID ANALYSIS** main page appears; click **Drag & Drop For Instant Analysis** section to upload a virus file.



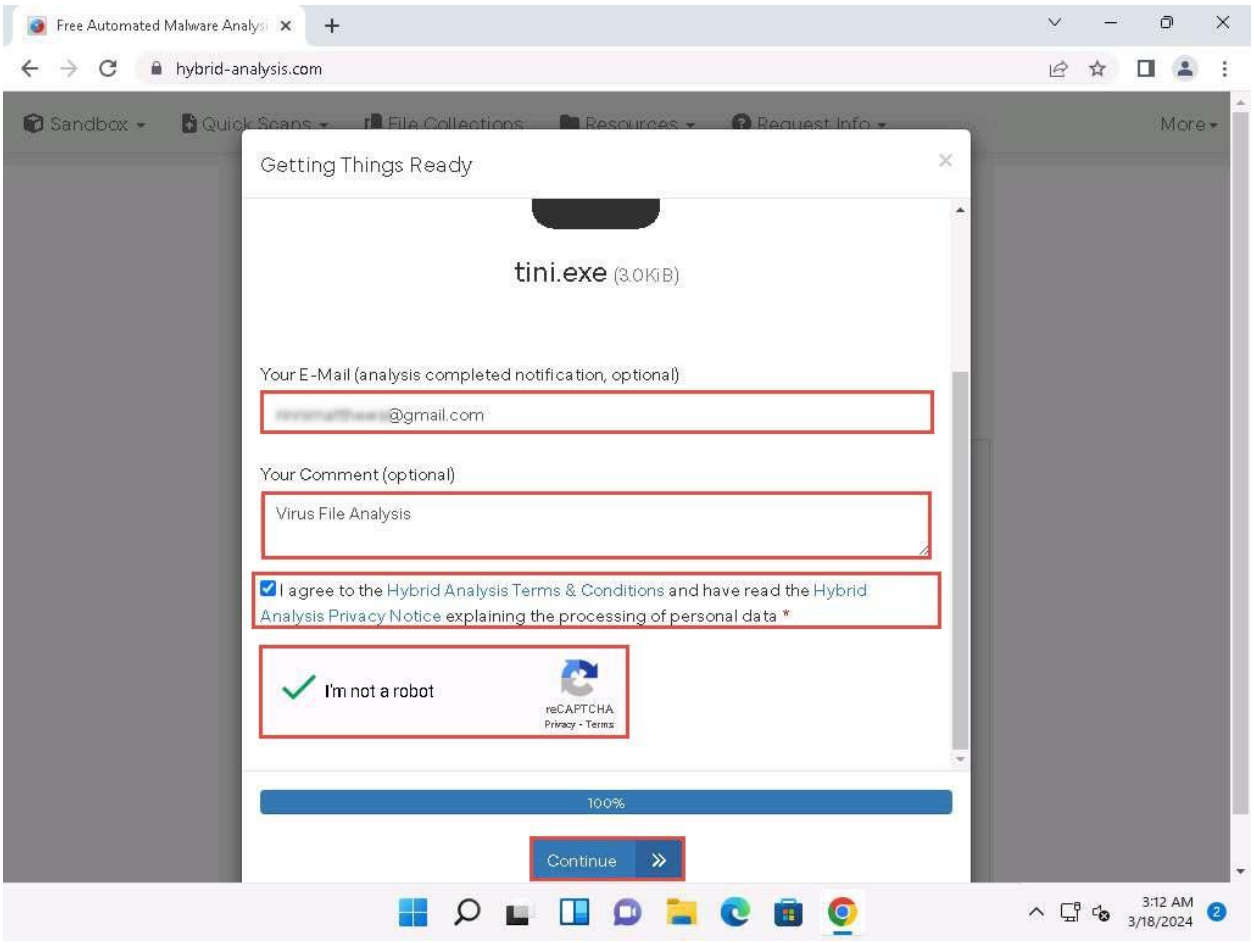
4. The **Open** window appears; navigate to **E:\CEH-Tools\CEHv13 Module 07 Malware Threats\Viruses**, select **tini.exe**, and click **Open**.



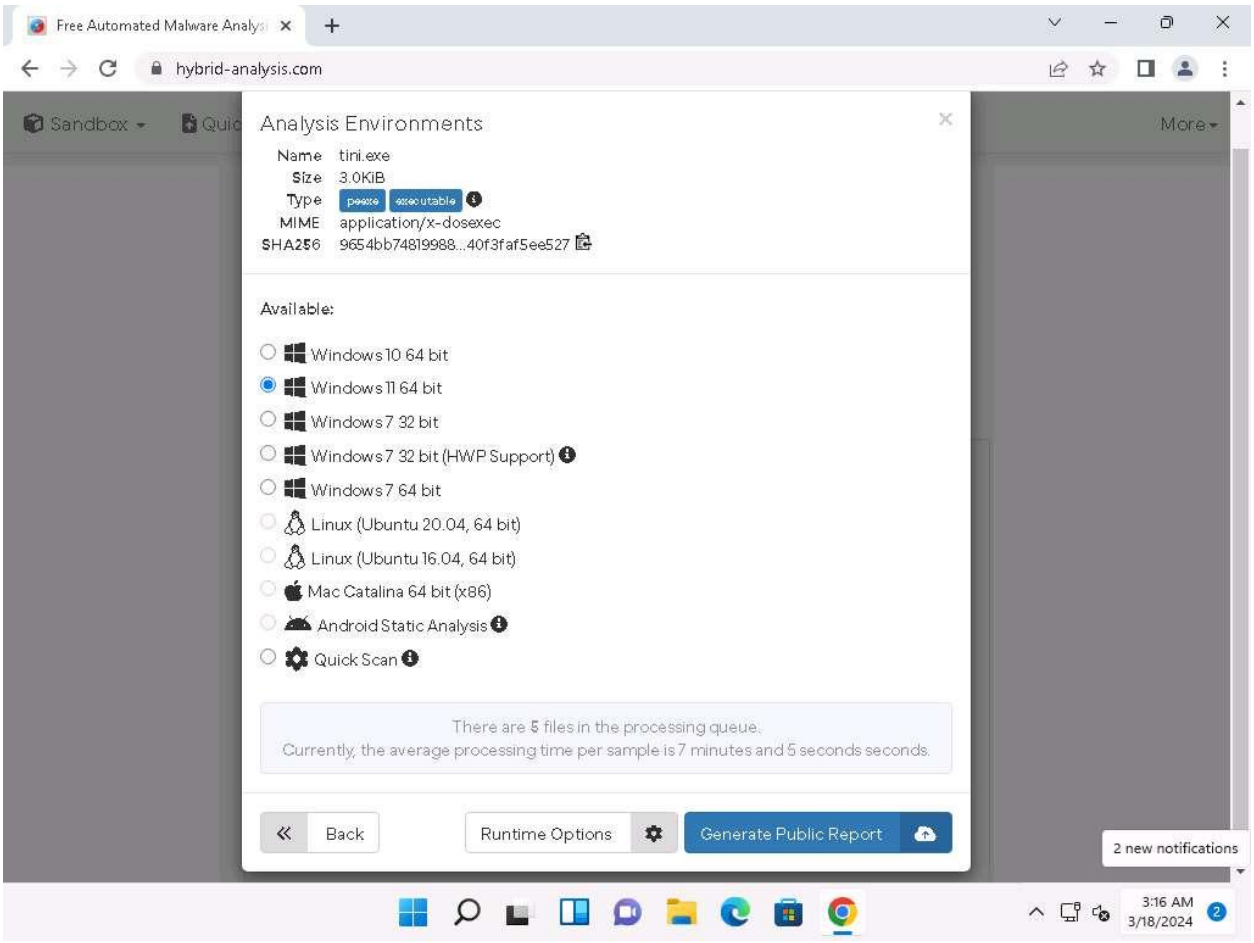
5. **Getting Things Ready** page appears and the virus file begins to upload. Once it is uploaded, the status bar reaches **100%**, as shown in the screenshot.



6. Now, enter your personal mail in **Your E-mail** field and enter a comment in **Your Comment** field. Scroll-down to check the **I agree to the Hybrid Analysis Terms & Conditions and have read the Hybrid Analysis Privacy Notice explaining the processsing of personal data** checkbox and **I'm not a robot** checkbox. Click **Continue**.



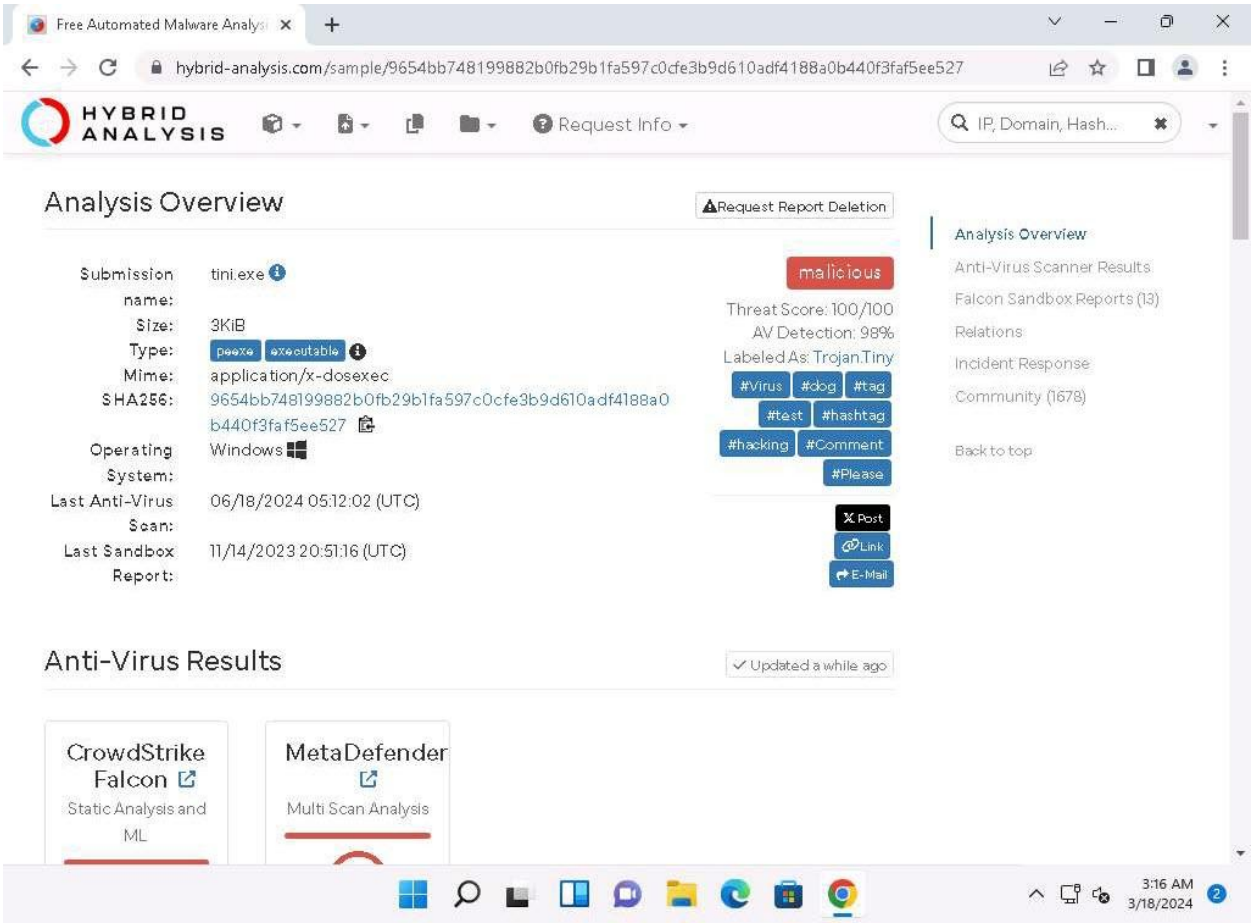
7. **Analysis Environments** page appears, select **Windows 11 64 bit** radio-button and click **Generate Public Report**.



8. The report generation process initializes and after it completes, **Analysis Overview** page appears.

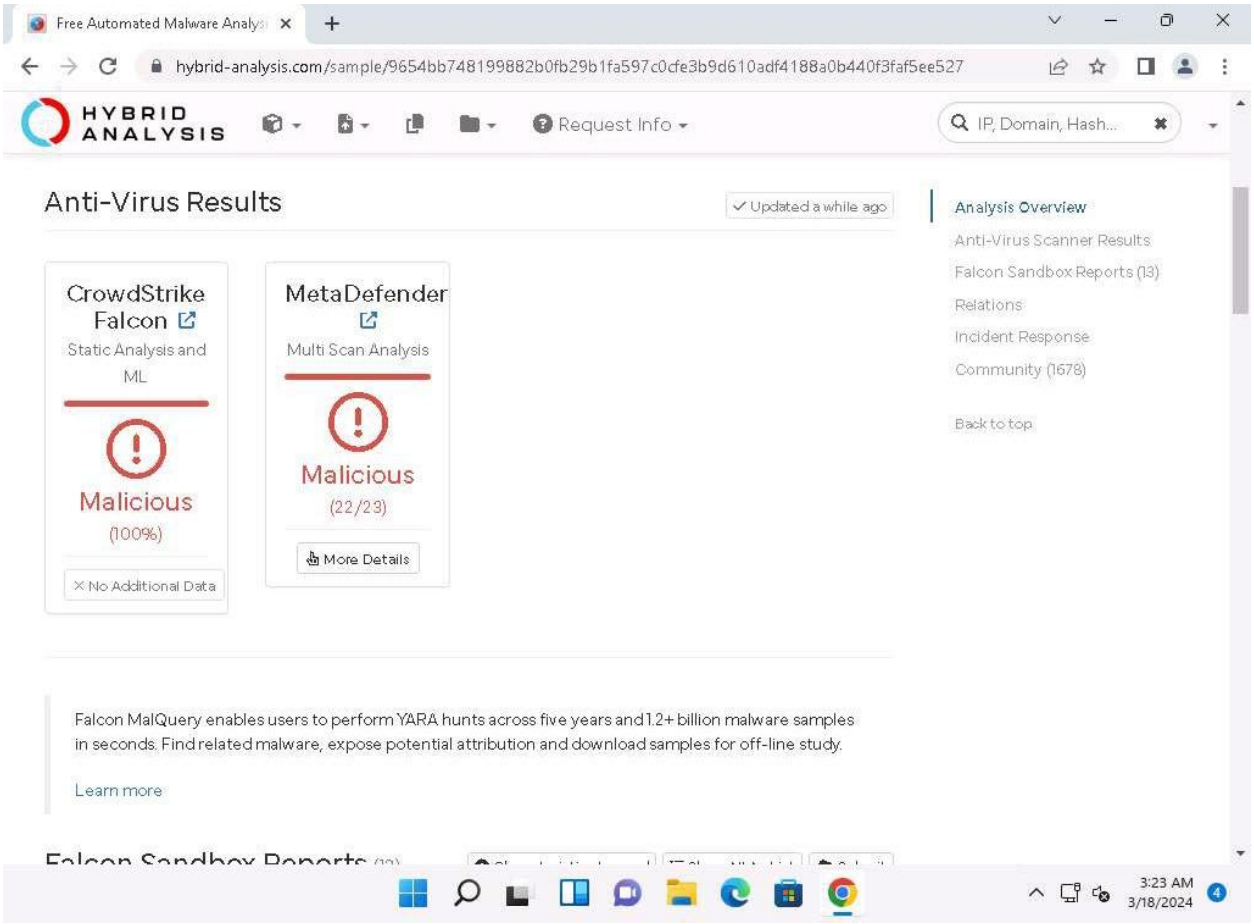
If you receive an error in the webpage, then reload the page to obtain the result.

9. You can observe that the file is detected as **malicious** with threat score at 100 along with the additional information such as SHA value.

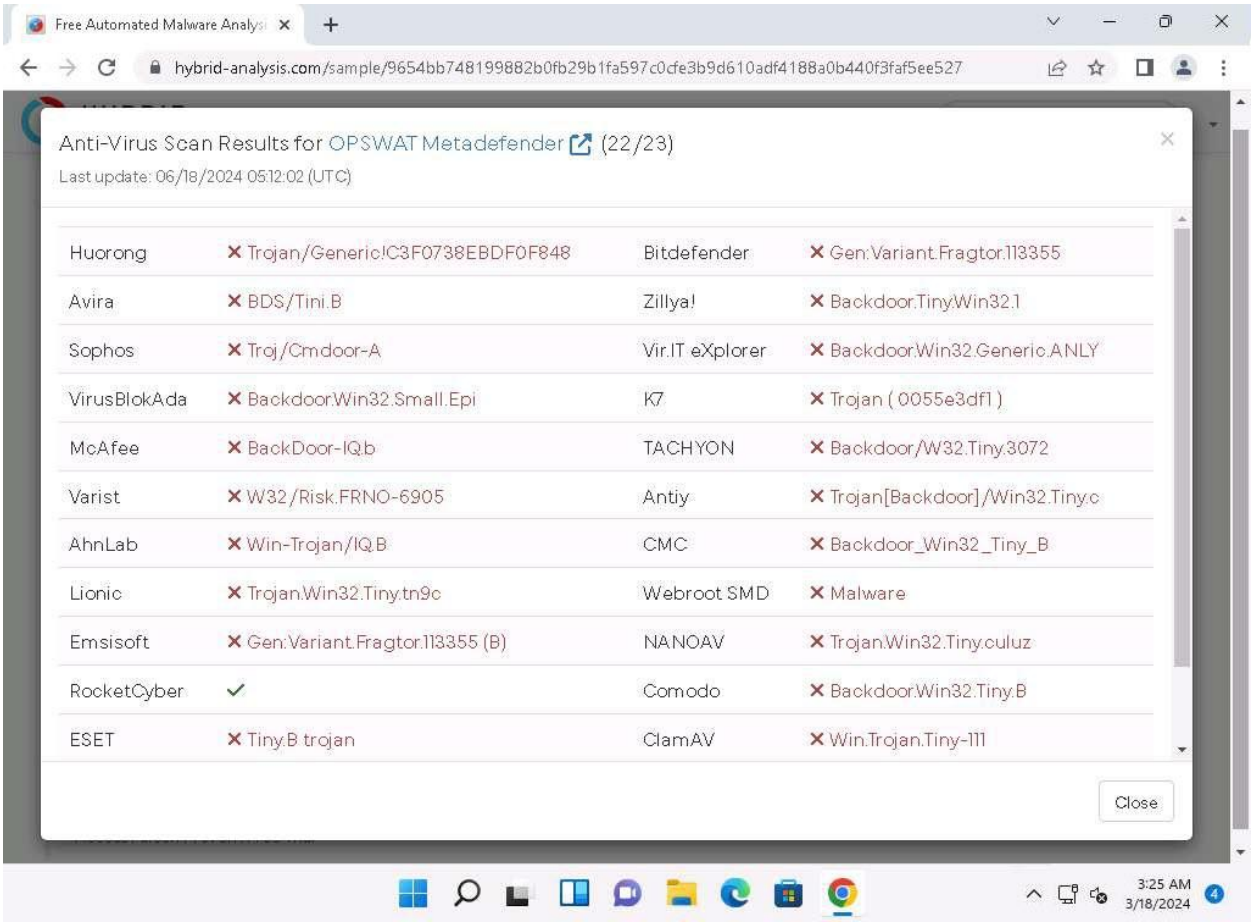


10. In the **Anti-Virus Results** section, you can observe the AV results obtained from different online resources such as **CrowdStrike Falcon** and **MetaDefender**.

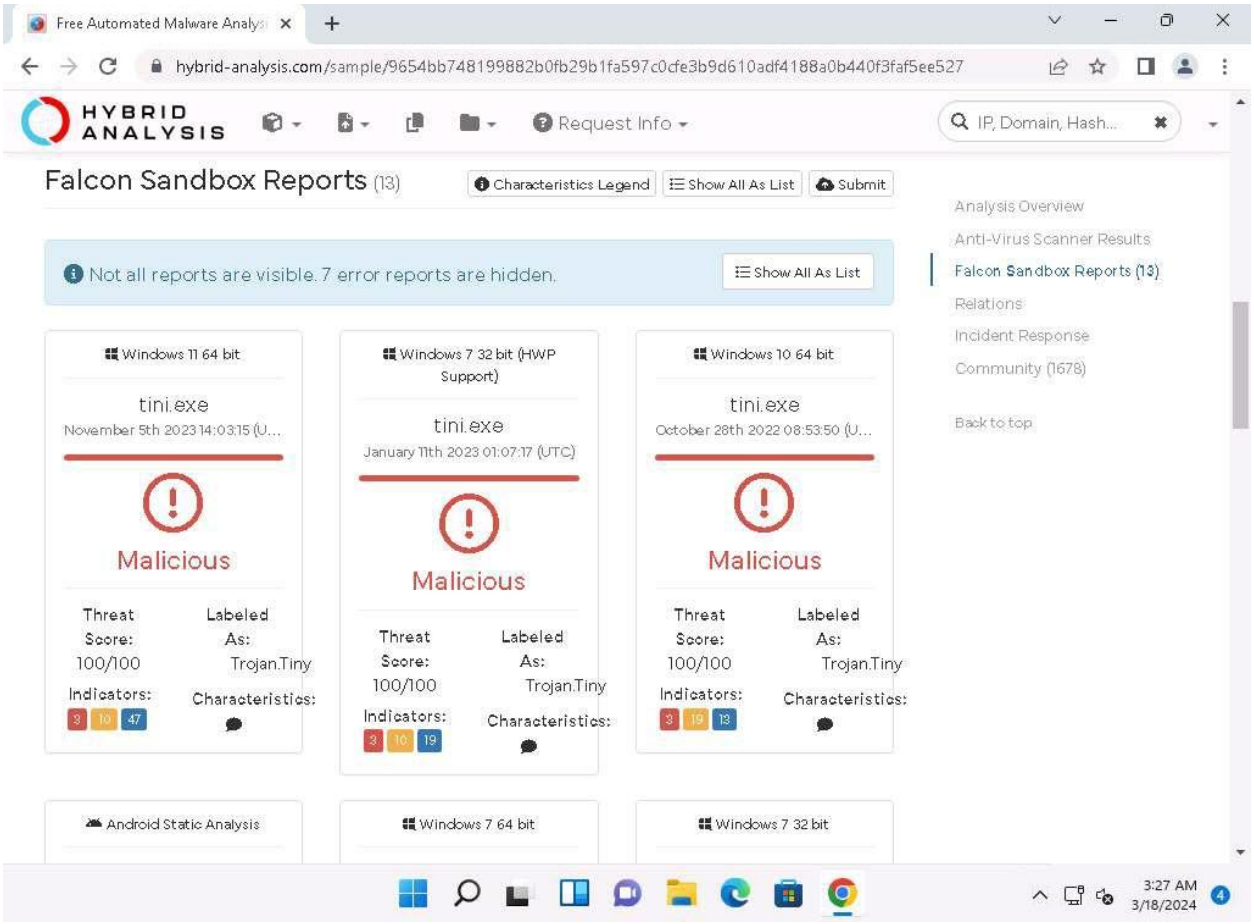
11. To further view the complete information obtained by the online resources you can click this icon ( ). Here, we will view the AV results obtained by the **MetaDefender**. Click **More Details** to open the result in the new tab.

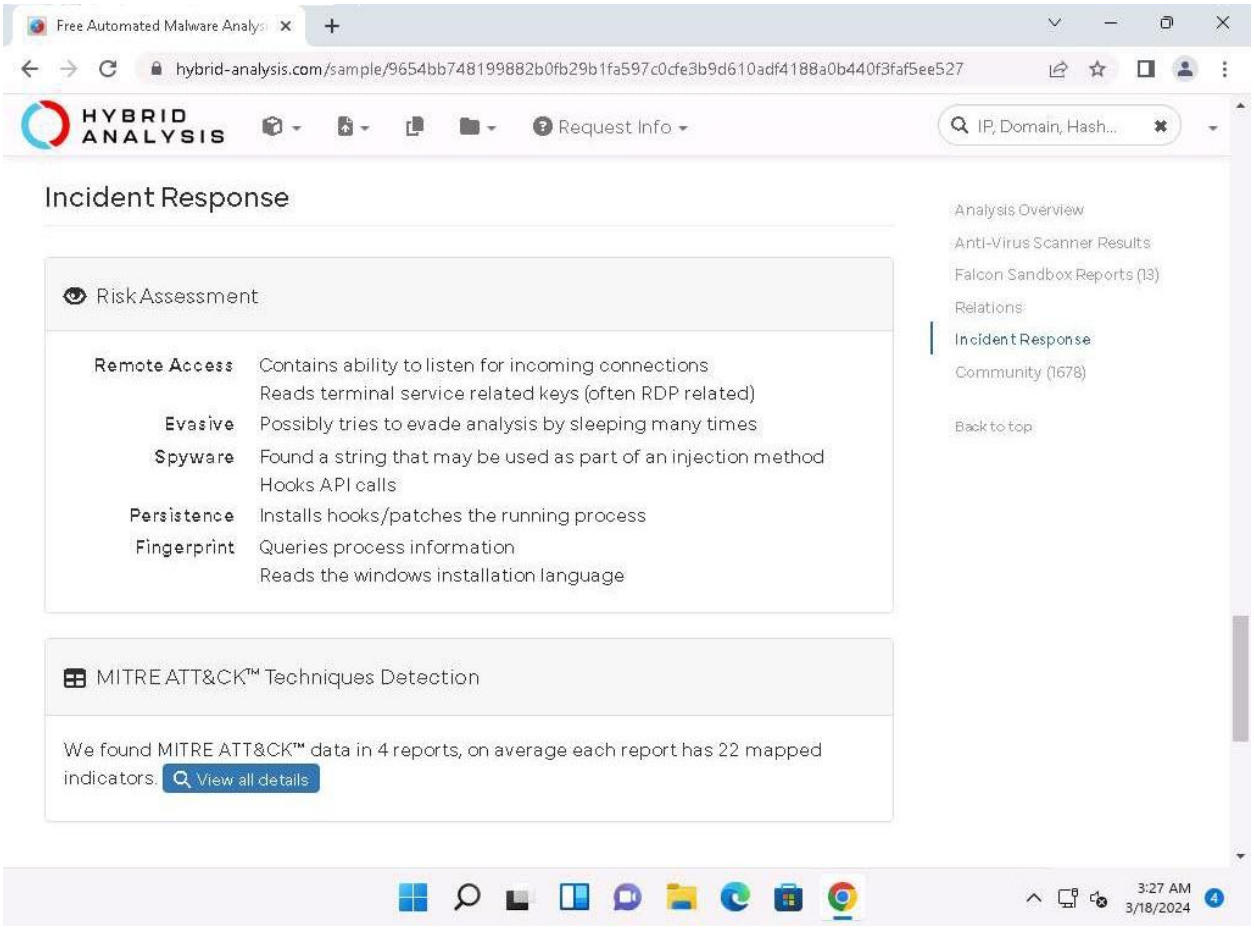


12. A pop-up appears showing the **Anti-Virus Scan Results for OPSWAT Metadefender**. Close the pop-up window.



13. You can further scroll-down in the results page to view information related to Falcon reports and Incident Response.


Free Automated Malware Analysis
hybrid-analysis.com/sample/9654bb748199882b0fb29b1fa597c0cfe3b9d610adf4188a0b440f3faf5ee527
HYBRID ANALYSIS
Request Info
IP, Domain, Hash...
Falcon Sandbox Reports (13)
Characteristics Legend
Show All As List
Submit
Not all reports are visible. 7 error reports are hidden.
Show All As List
Analysis Overview
Anti-Virus Scanner Results
Falcon Sandbox Reports (13)
Relations
Incident Response
Community (1678)
Back to top
Windows 11 64 bit
tini.exe
November 5th 2023 14:03:15 (U...
Malicious
Threat Score: 100/100
Labeled As: Trojan.Tiny
Indicators: 3 10 47
Characteristics:
Windows 7 32 bit (HWP Support)
tini.exe
January 11th 2023 01:07:17 (UTC)
Malicious
Threat Score: 100/100
Labeled As: Trojan.Tiny
Indicators: 3 10 19
Characteristics:
Windows 10 64 bit
tini.exe
October 28th 2022 08:53:50 (U...
Malicious
Threat Score: 100/100
Labeled As: Trojan.Tiny
Indicators: 3 19 13
Characteristics:
Android Static Analysis
Windows 7 64 bit
Windows 7 32 bit
3:27 AM
3/18/2024



14. This concludes the demonstration of malware scanning using Hybrid Analysis.
15. Close all open windows.
16. You can also use other local and online malware scanning tools such as **Any.Run** (https://app.any.run) **Valkyrie Sandbox** (https://valkyrie.comodo.com), **JOESandbox Cloud** (https://www.joesandbox.com), **Jotti** (https://virusscan.jotti.org) to perform online malware scanning.

Question 7.3.1.1

Analyze malware using online Hybrid Analysis services. What the name of the Analysis Environment that was selected in this task?

Correct

Question 7.3.1.2

Analyze malware using online Hybrid Analysis services. Enter the name of the malicious file that was uploaded for analysis in this lab.

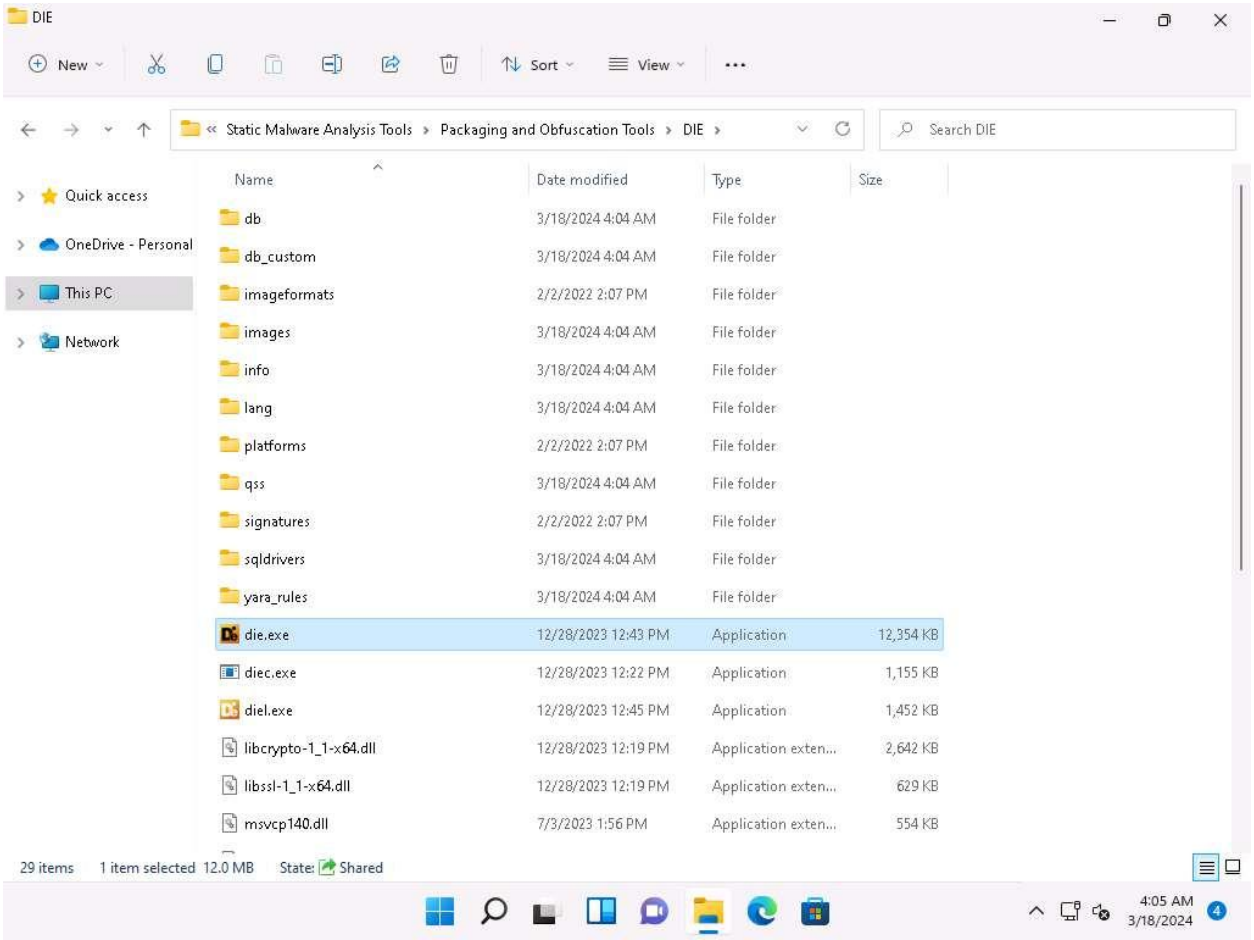
Correct

Task 2: Analyze ELF Executable File using Detect It Easy (DIE)

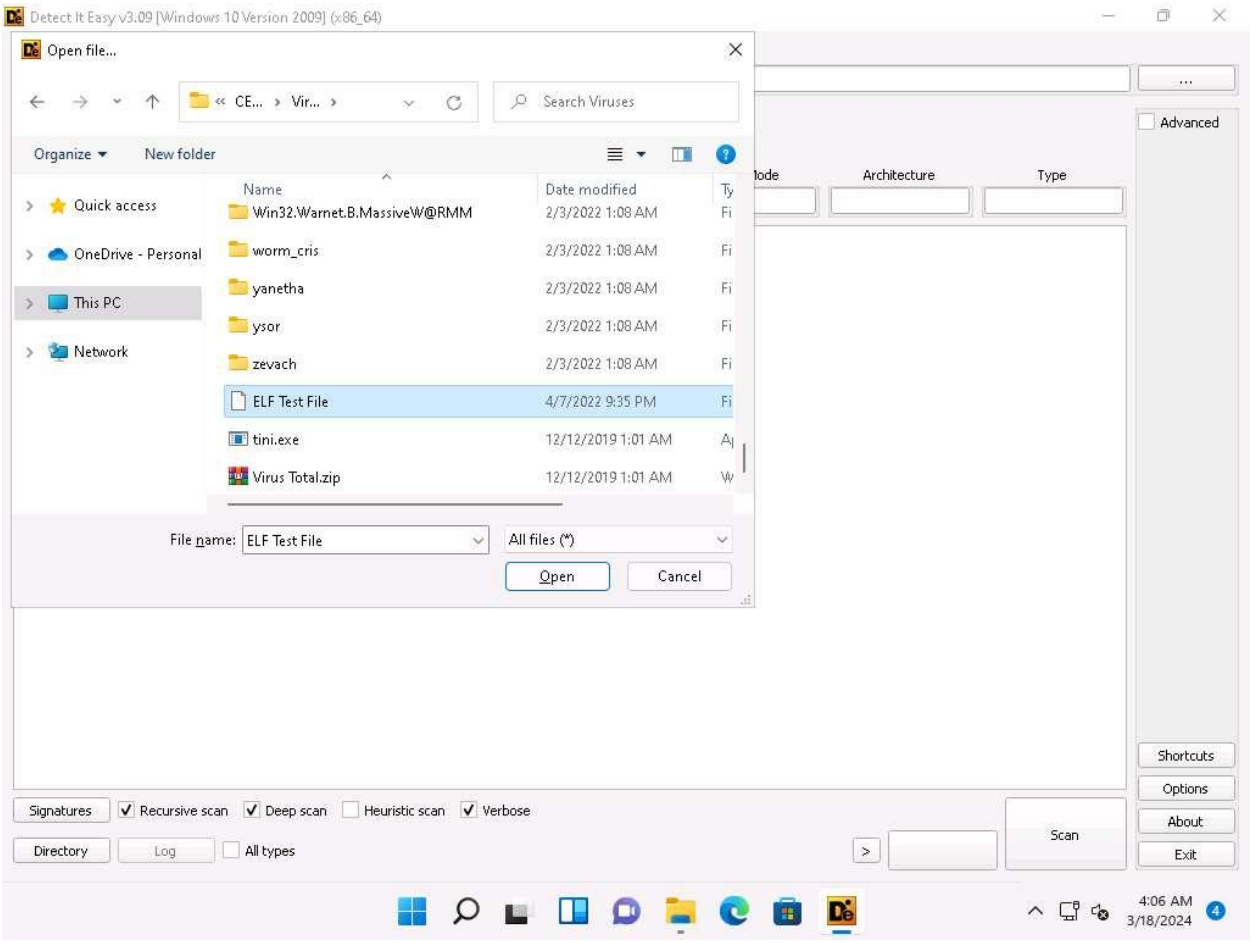
The Executable and Linkable Format (ELF) is a generic executable file format in Linux environment. It contains three main components including ELF header, sections, and segments. Each component plays an independent role in the loading and execution of ELF executables. The static analysis of an ELF file involves investigating an ELF executable file without running or installing it. It also involves accessing the binary code and extracting valuable artifacts from the program. Numerous tools can be used to perform static analysis on ELF files. In this task, we will be using Detect It Easy (DIE) tool to analyze ELF file.

Detect It Easy (DIE) is an application used for determining the types of files. Apart from the Windows, DIE is also available for Linux and Mac OS. It has a completely open architecture of signatures and can easily add its own algorithms for detecting or modifying the existing signatures. It detects a file's compiler, linker, packer, etc. using a signature-based detection method.

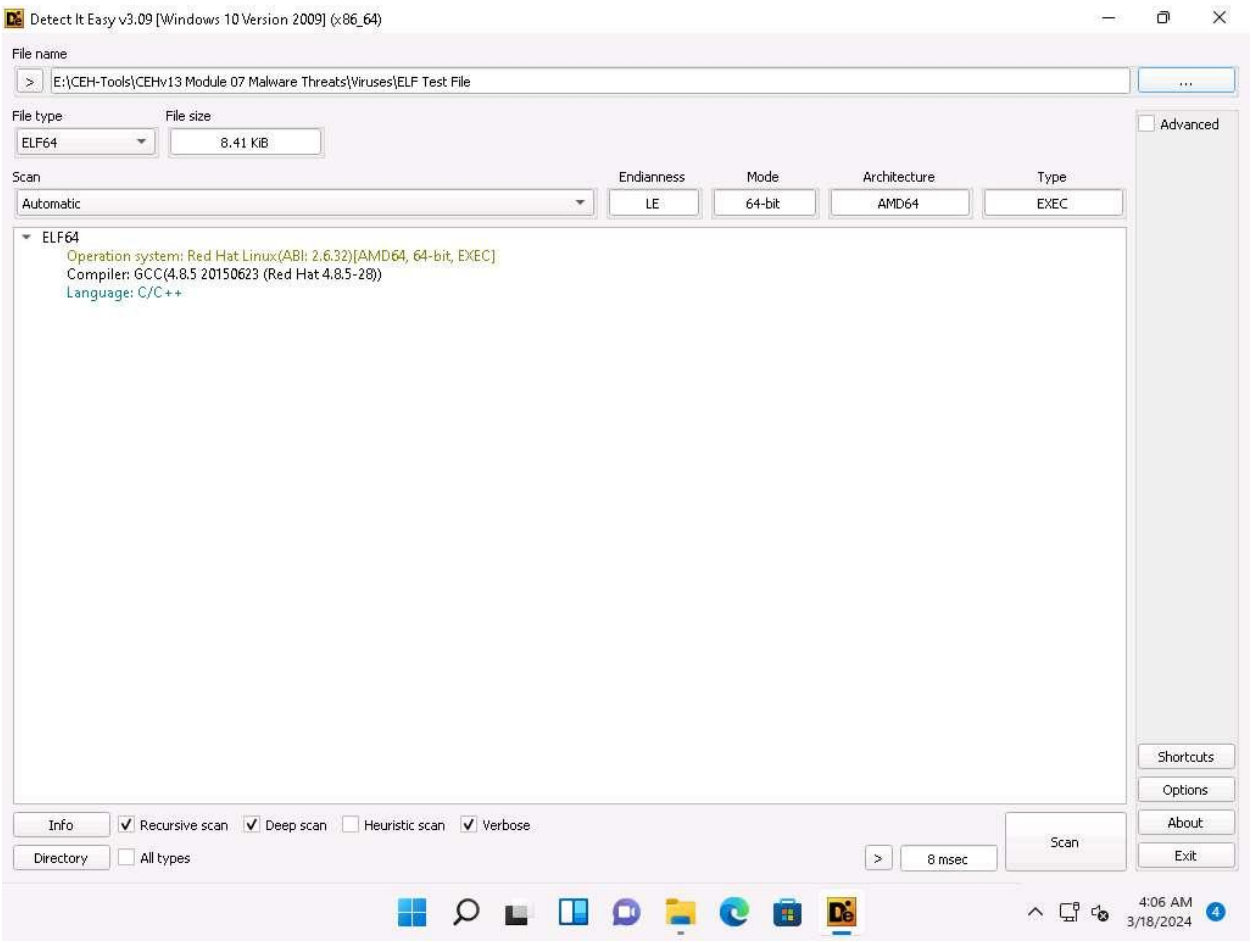
1. In the **Windows 11** machine, navigate to **E:\CEH-Tools\CEHv13 Module 07 Malware Threats\Malware Analysis Tools\Static Malware Analysis Tools\Packaging and Obfuscation Tools\DIE** and double-click **die.exe**.



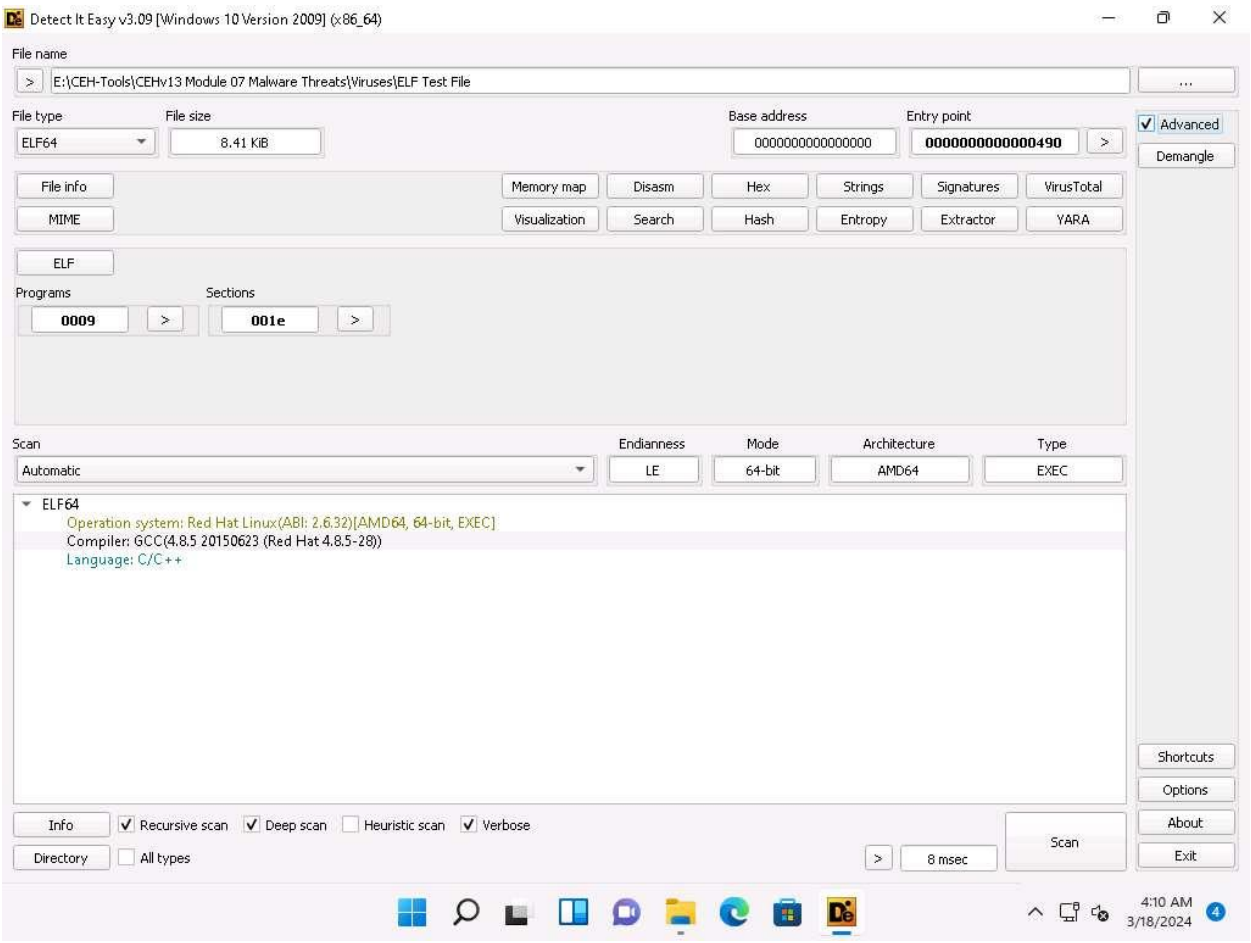
2. **Open File - Security Warning** appears, click **Run**.
3. **Detect It Easy** window appears. Click ellipses icon next to the **File name** text field.
4. The **Open file…** window appears; navigate to **E:\CEH-Tools\CEHv13 Module 07 Malware Threats\Viruses**, select **ELF Test File**, and click **Open**.



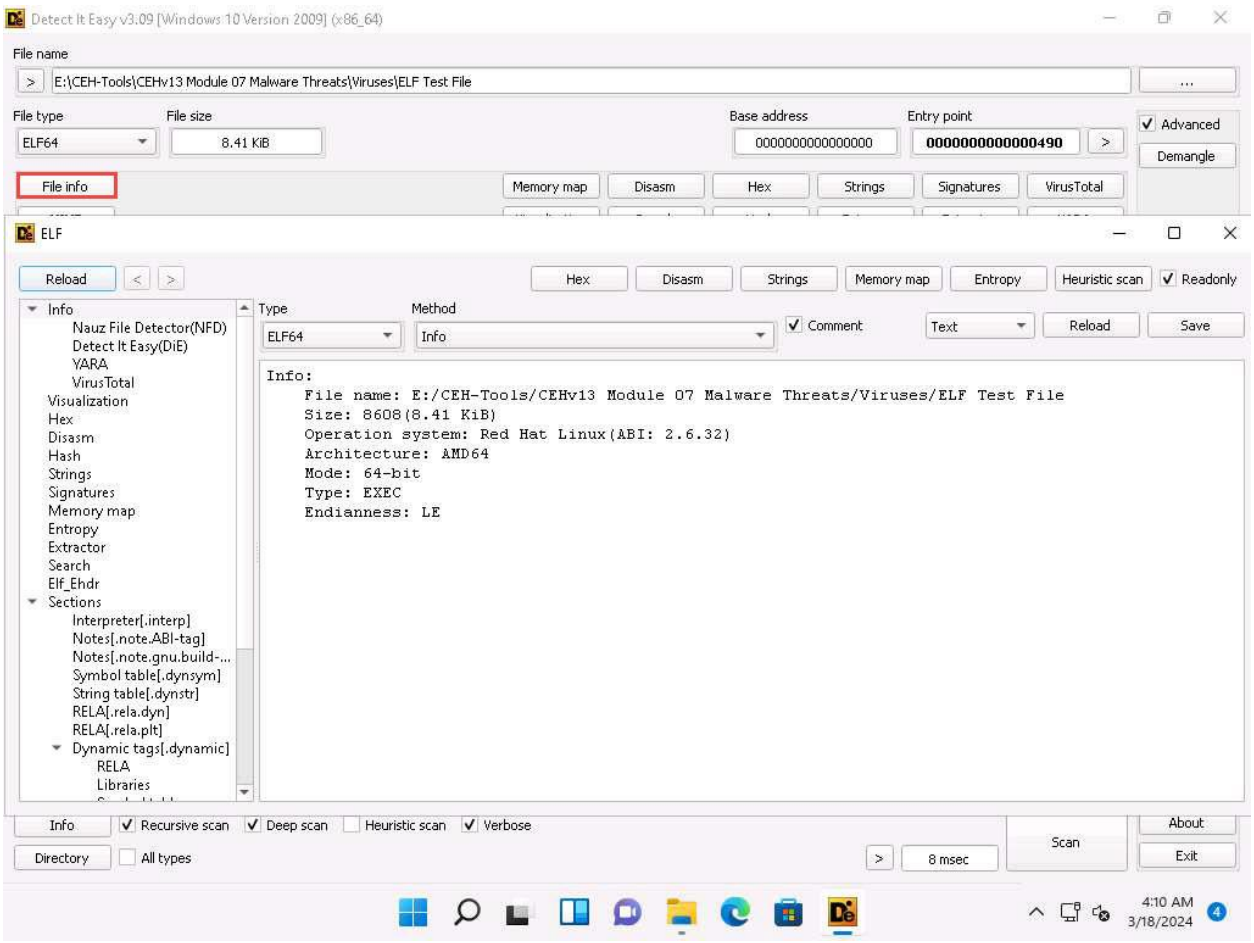
5. **Detect It Easy** automatically scans the file and result appears showing the Operating system, compiler and language details in the middle pane, as shown in the screenshot.



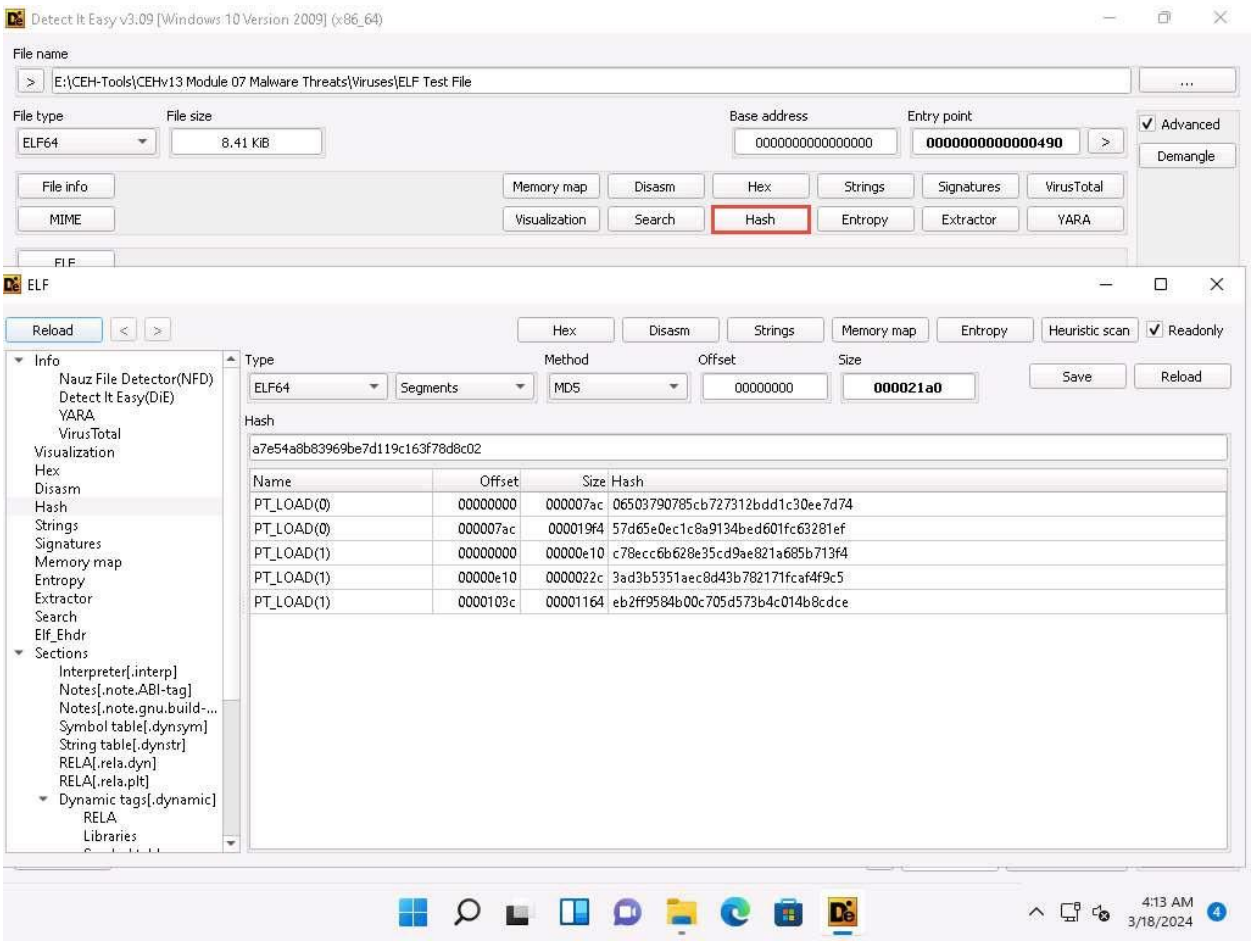
6. Now, check the **Advanced** checkbox present at the right pane.



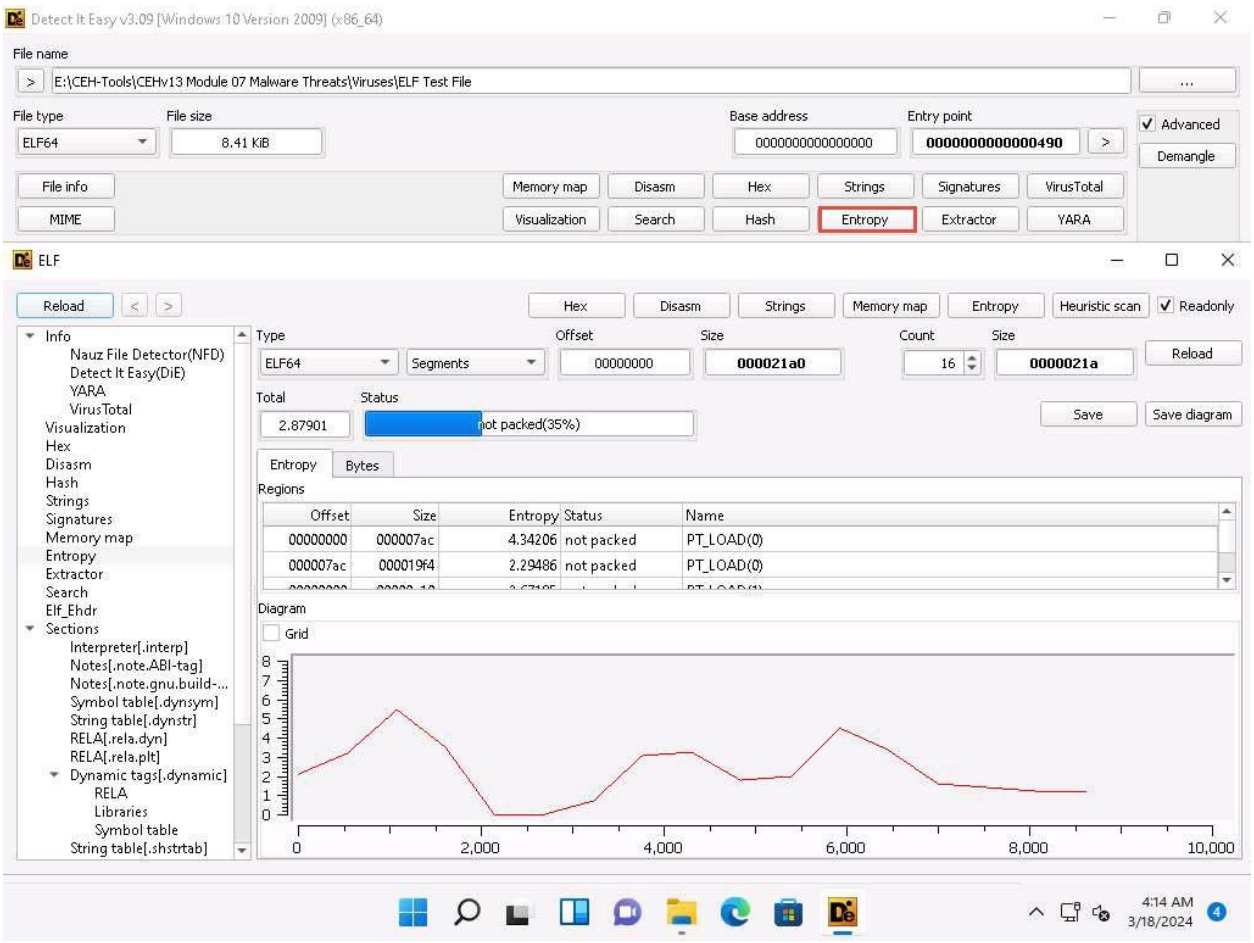
7. Click **File info** button from the top left corner of the window. Info window appears, you can observe information such as File name, size, MD5, SHA1, Entropy, entry points, etc.



8. After viewing the information, close the window.
9. Similarly, click **Hash** button from the top right corner of the window to view the information related to hash. Close the window after viewing the information.



10. Click **Entropy** button from the top right corner of the window. Here, you can observe the status, size and graph of entropy. Close the window after viewing the Entropy information.



11. Similarly, you can further explore other functions such as MIME, Hex, Signatures and Demangle.
12. This concludes the demonstration of ELF file analysing using Detect It Easy (DIE).
13. Close all the open windows.
14. You can also use other packaging/obfuscation tools such as **Macro_Pack** (https://github.com), **UPX** (https://upx.github.io), **ASPack** (http://www.aspack.com), or **VMprotect** (https://vmpsoft.com) to identify packing/obfuscation methods.

Question 7.3.2.1

Detect a file's compiler, linker, packer, etc. using Detect It Easy (DIE). Enter the name of the operating system that was detected from the ELF file in this task.

Correct

Task 3: Perform Malware Disassembly using IDA and OllyDbg

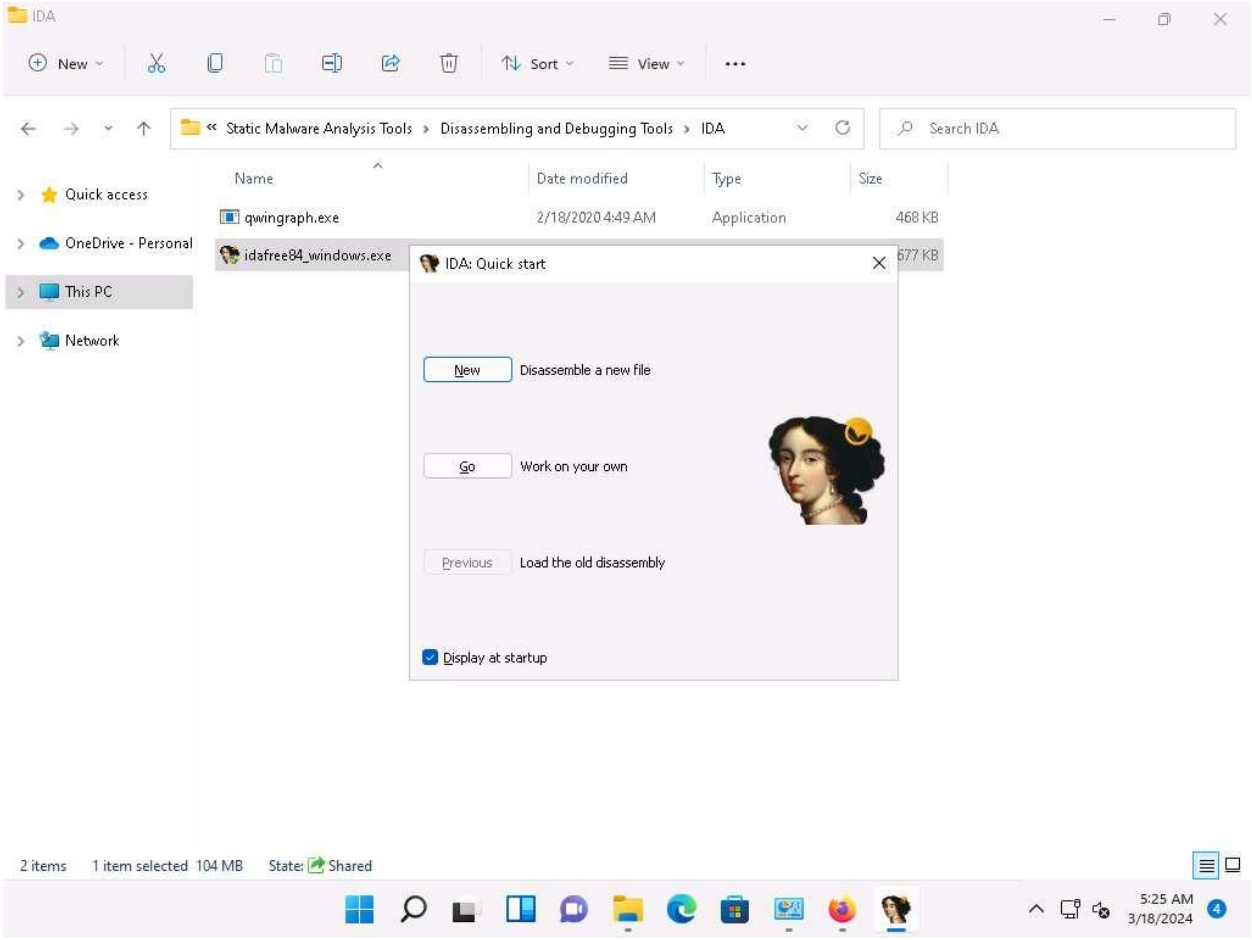
Static analysis also includes the dismantling of a given executable into binary format to study its functionalities and features. This process helps identify the language used for programming the malware, look for APIs that reveal its function, and retrieve other information. Based on the reconstructed assembly code, you can inspect the program logic and recognize its threat potential. This process uses debugging tools such as IDA Pro and OllyDbg.

IDA As a disassembler, IDA explores binary programs, for which the source code might not be available, to create maps of their execution. The primary purpose of a disassembler is to display the instructions actually executed by the processor in a symbolic representation called “assembly language.” However, in real life, things are not always simple. Hostile code usually does not cooperate with the analyst. Viruses, worms, and Trojans are often armored and obfuscated; as such, more powerful tools are required. The debugger in IDA complements the static analysis capabilities of the disassembler. By allowing an analyst to single-step through the code being investigated, the debugger often bypasses the obfuscation. It helps obtain data that the more powerful static disassembler will be able to process in depth.

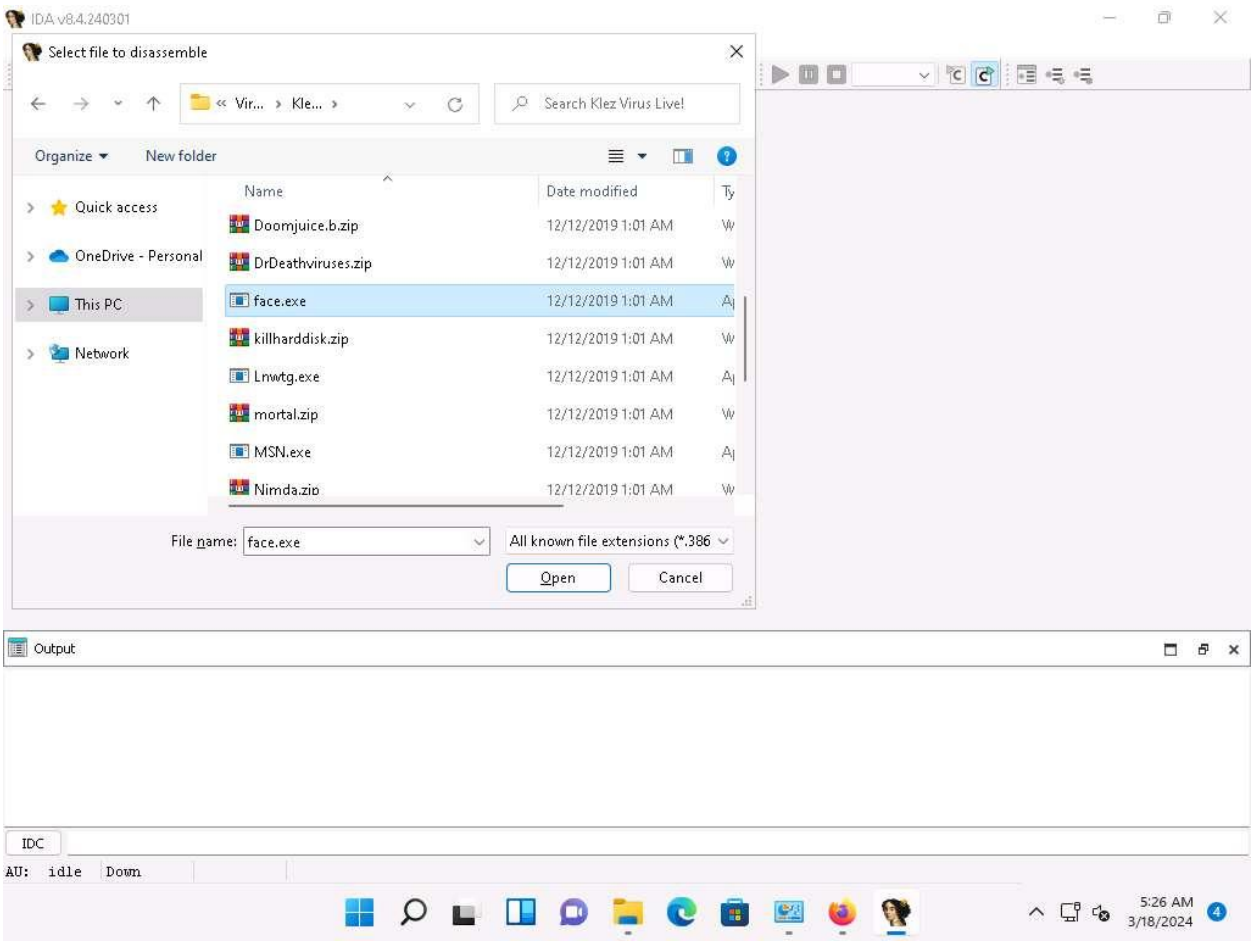
OllyDbg OllyDbg is a debugger that emphasizes binary code analysis, which is useful when source code is unavailable. It traces registers, recognizes procedures, API calls switches, tables, constants, and strings, and locates routines from object files and libraries.

There is a new debugging option, “Set permanent breakpoints on system calls.” When active, it requests OllyDbg to set breakpoints on KERNEL32.UnhandledExceptionFilter(), NTDLL.KiUserExceptionDispatcher(), NTDLL.ZwContinue(), and NTDLL.NtQueryInformationProcess().

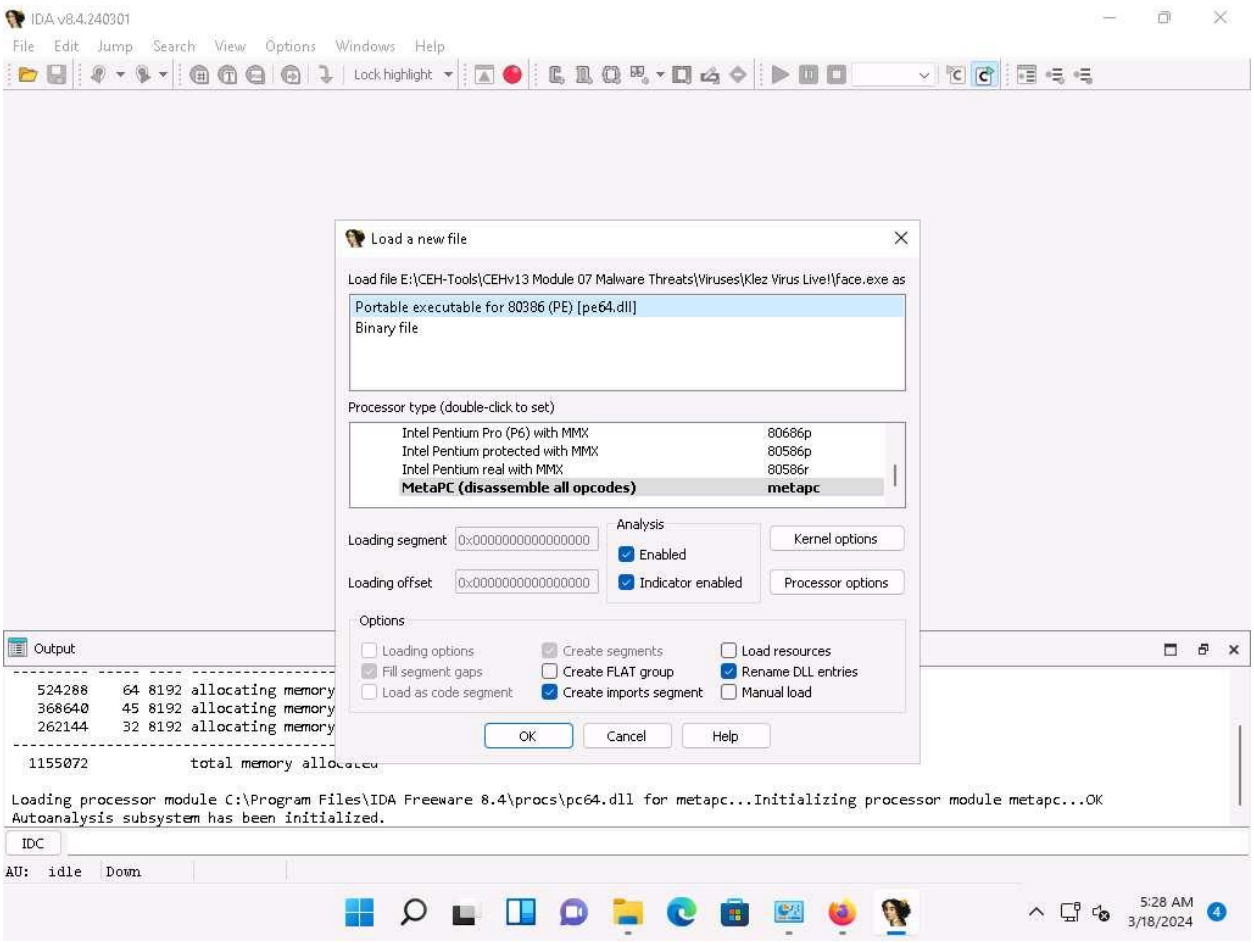
1. In the **Windows 11** machine, search for **ida** in the Windows search, the **IDA Freeware 8.4** appears in the result, click **Open** to launch it.
2. If the **IDA License** window appears, click on **I Agree**.
3. **User interface telemetry** window appears, uncheck **Yes, I want to help improve IDA** checkbox and click **OK**.
4. The **IDA: Quick start** pop-up appears; click on **New** to select a malicious file for disassembly.



5. The **IDA** main window appears, along with the **Select file to disassemble** window.
6. In the **Select file to disassemble** window, navigate to **E:\CEH-Tools\CEHv13 Module 07 Malware Threats\Viruses\Klez Virus Live!**, select **face.exe**, and click **Open**.



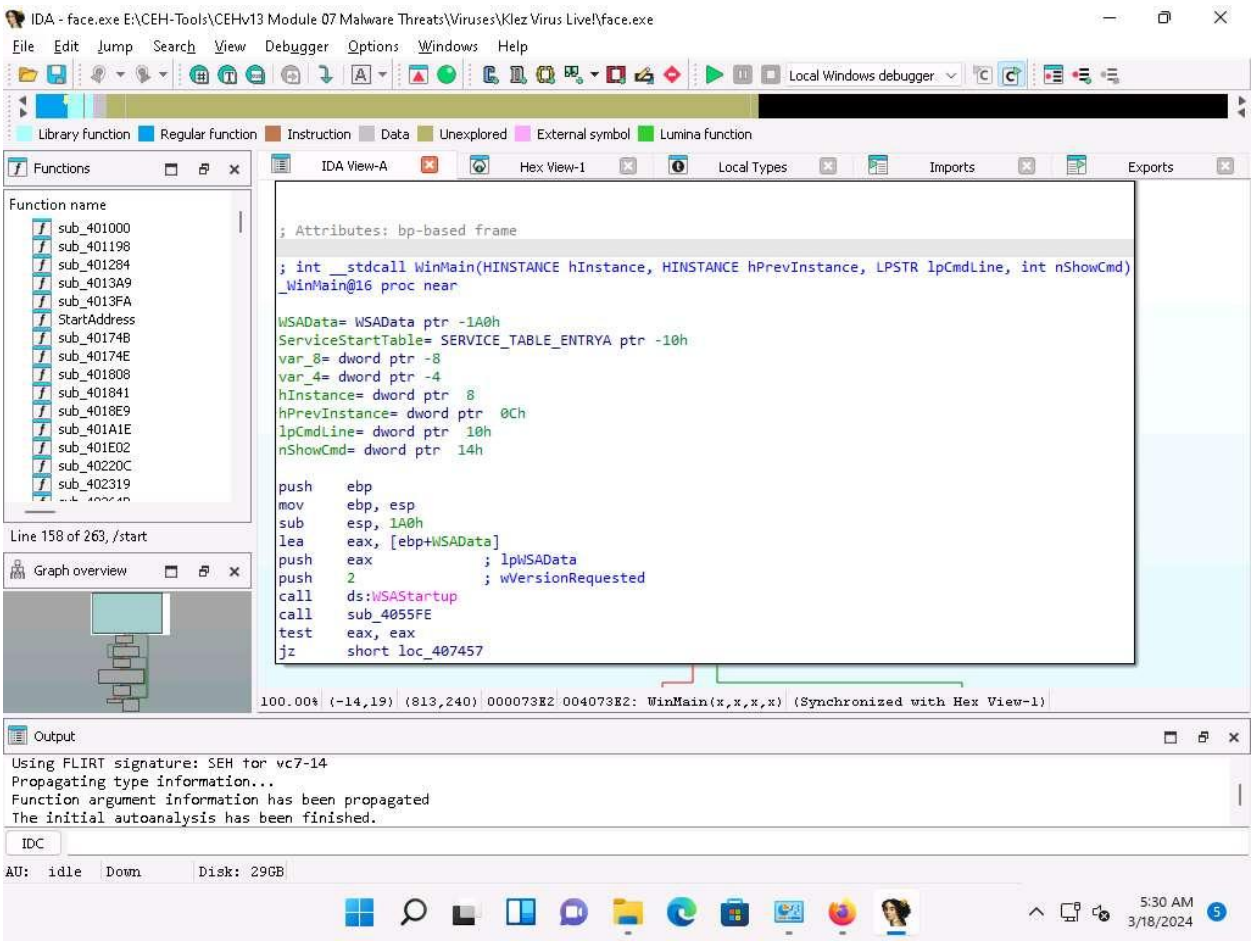
7. The **Load a new file** window appears; by default, the **Portable executable for 80386 (PE) [pe64.dll]** option selected; click **OK**.



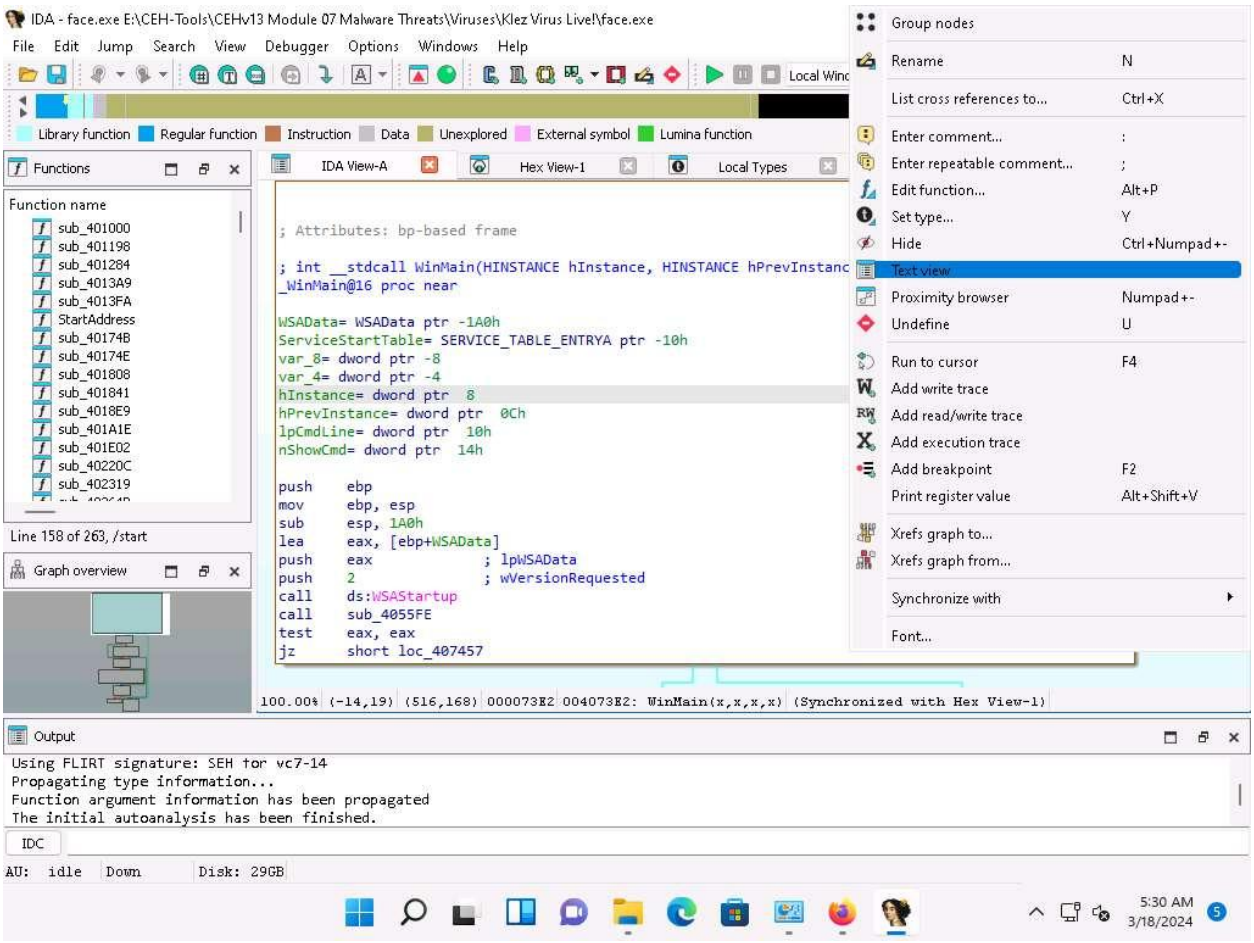
If a **Warning** pop-up appears, click **OK**.

If a **Please confirm** dialog-box appears, read the instructions carefully, and then click **Yes**.

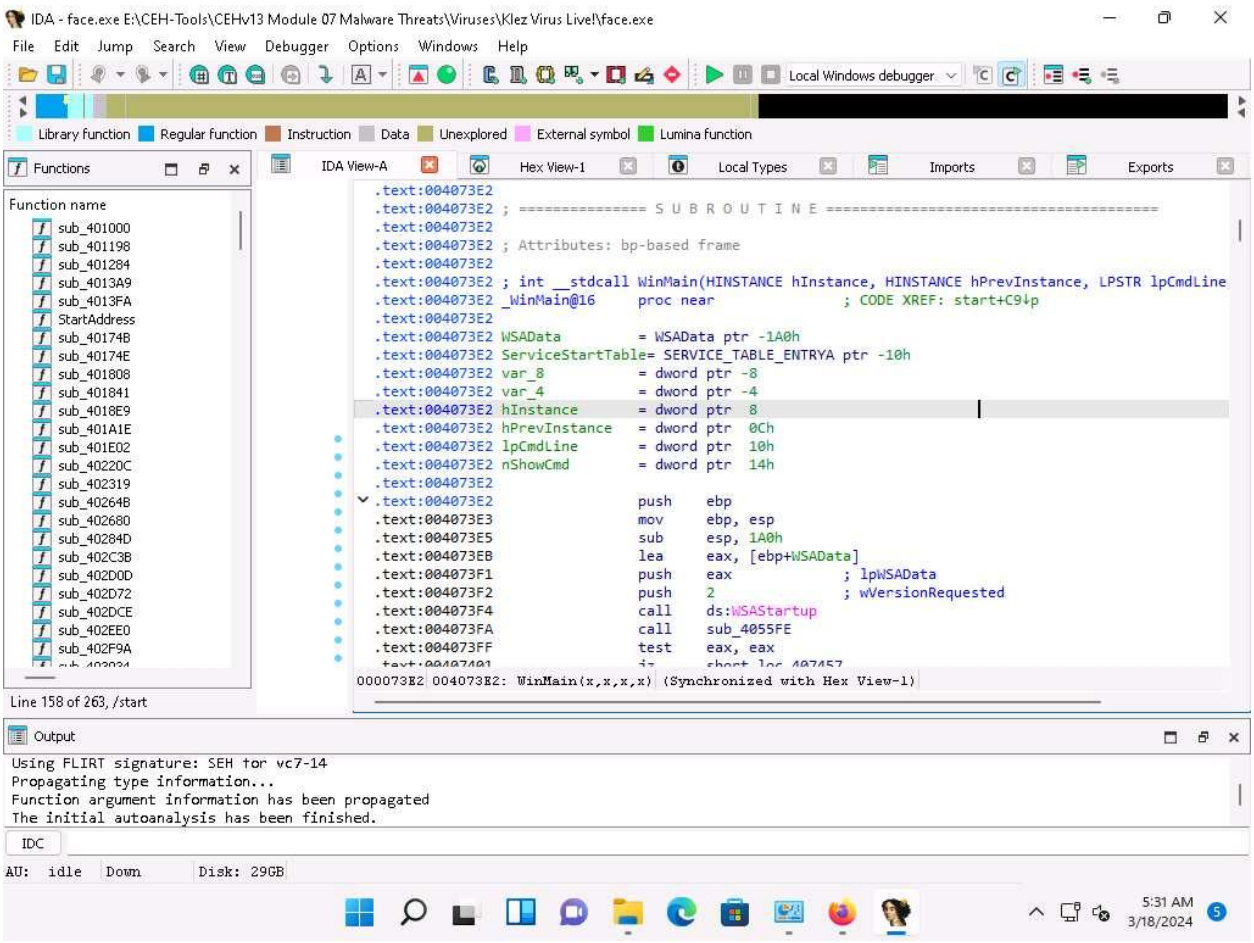
8. IDA completes the analysis of the imported malicious file and displays the results in the **IDA View-A** tab, as shown in the screenshot.



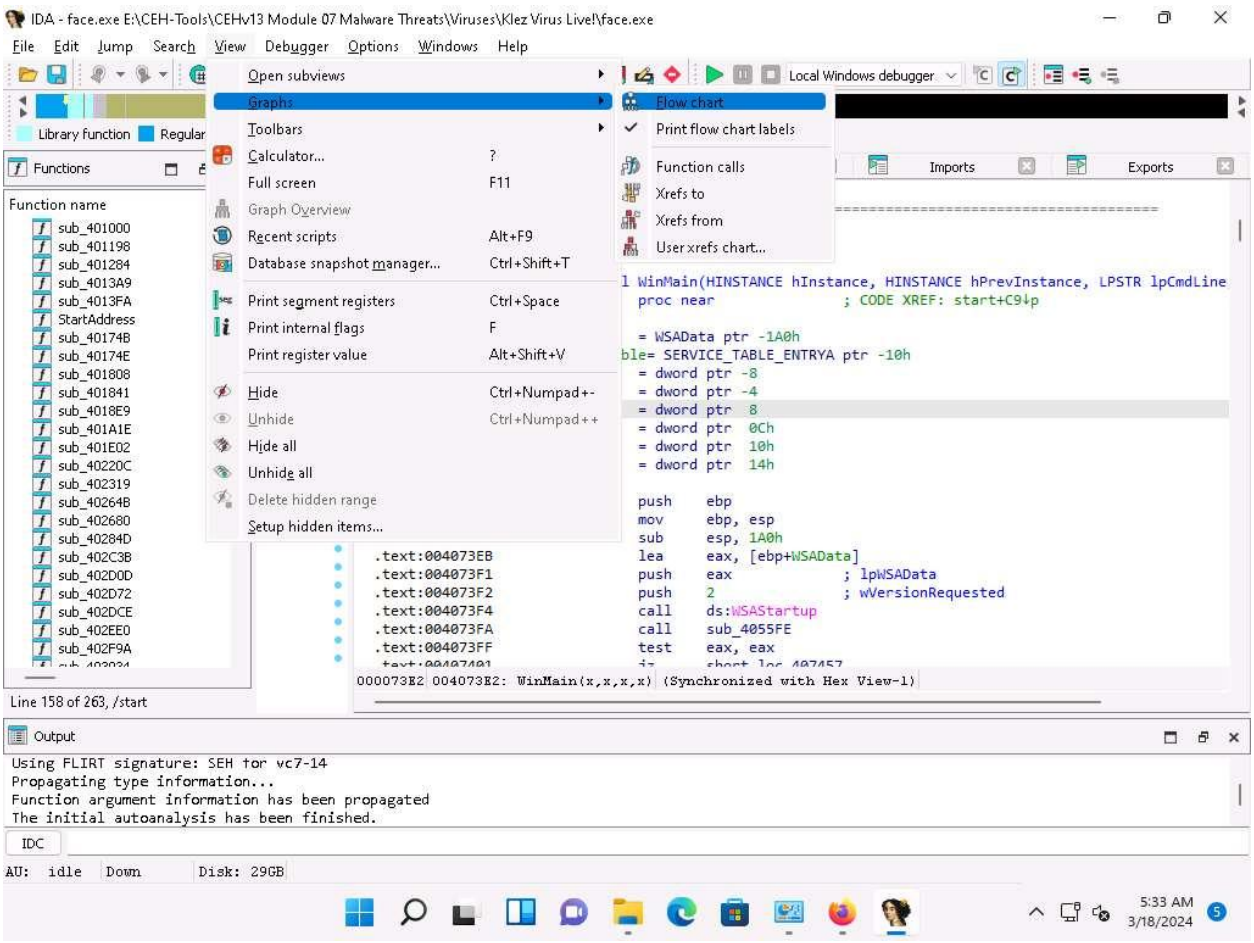
9. In the **IDA View-A** section, right-click anywhere and choose **Text view** from the context menu to view the text information of the malicious file uploaded to IDA for analysis.



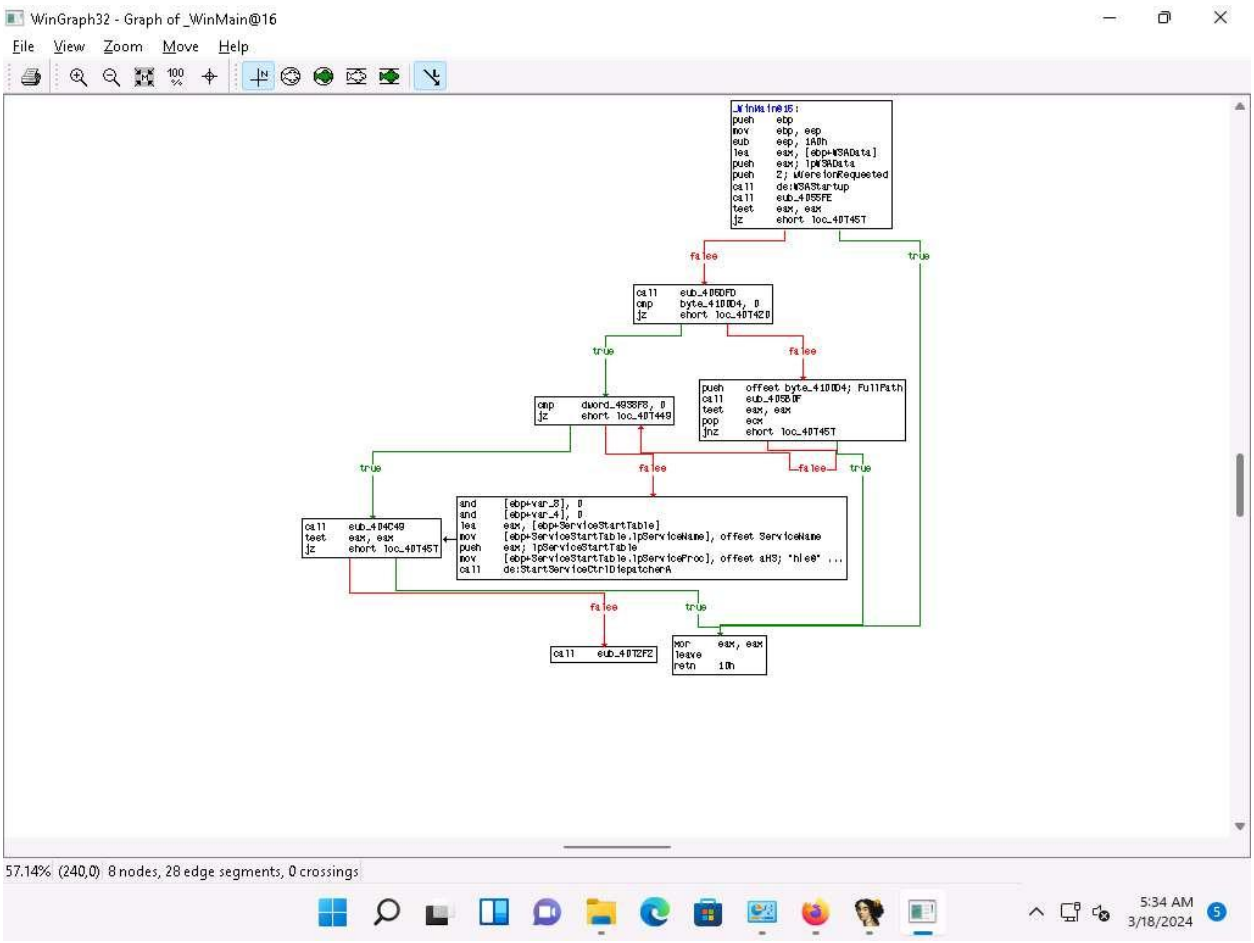
10. This reveals the text view of the malicious file, allowing analysis of its information.



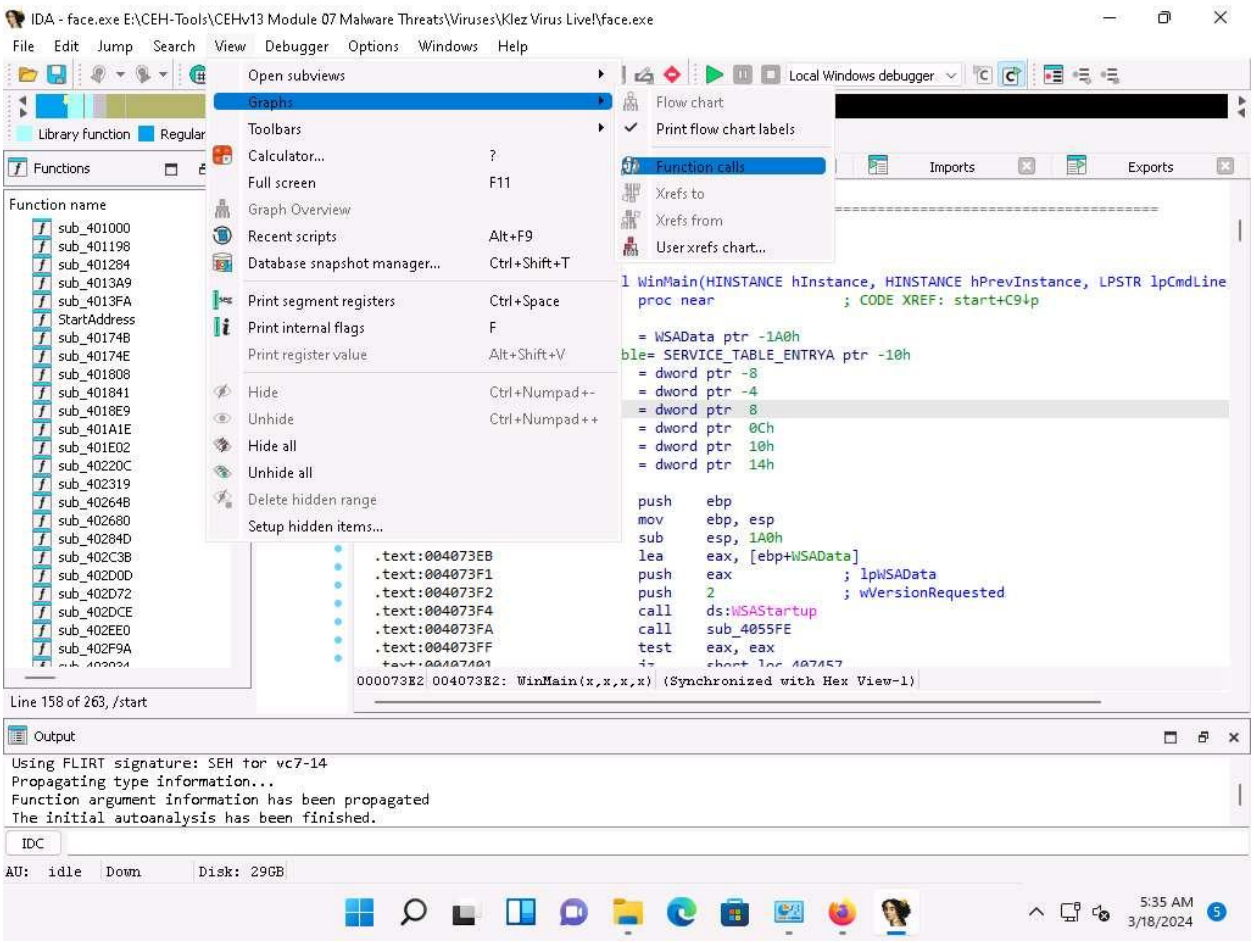
11. Maximize the IDA window. To view the flow of the uploaded malicious file, navigate to **View** --> **Graphs** and click **Flow chart**.



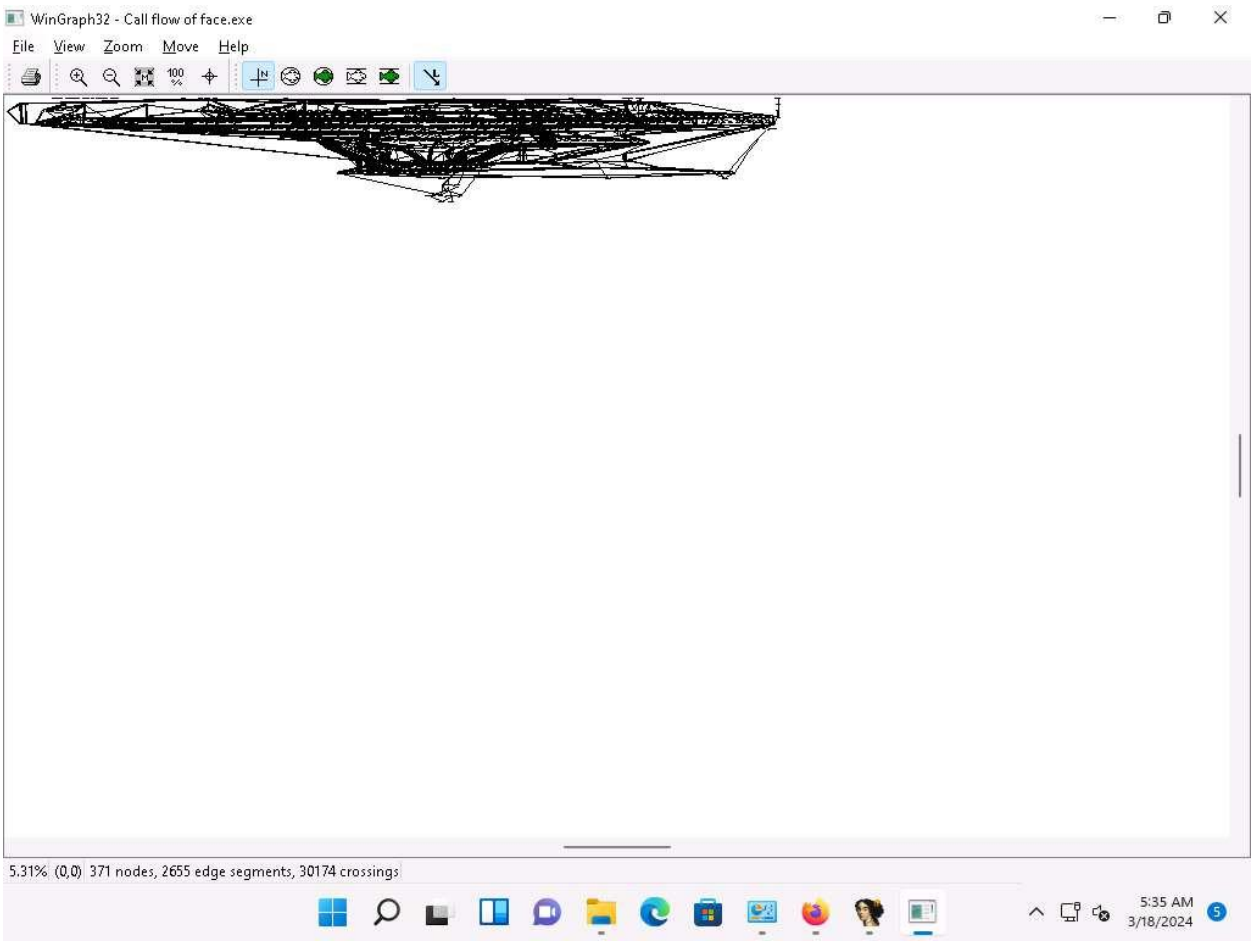
12. A **Graph** window appears with the flow. You may zoom in and adjust the screen to view this more clearly.

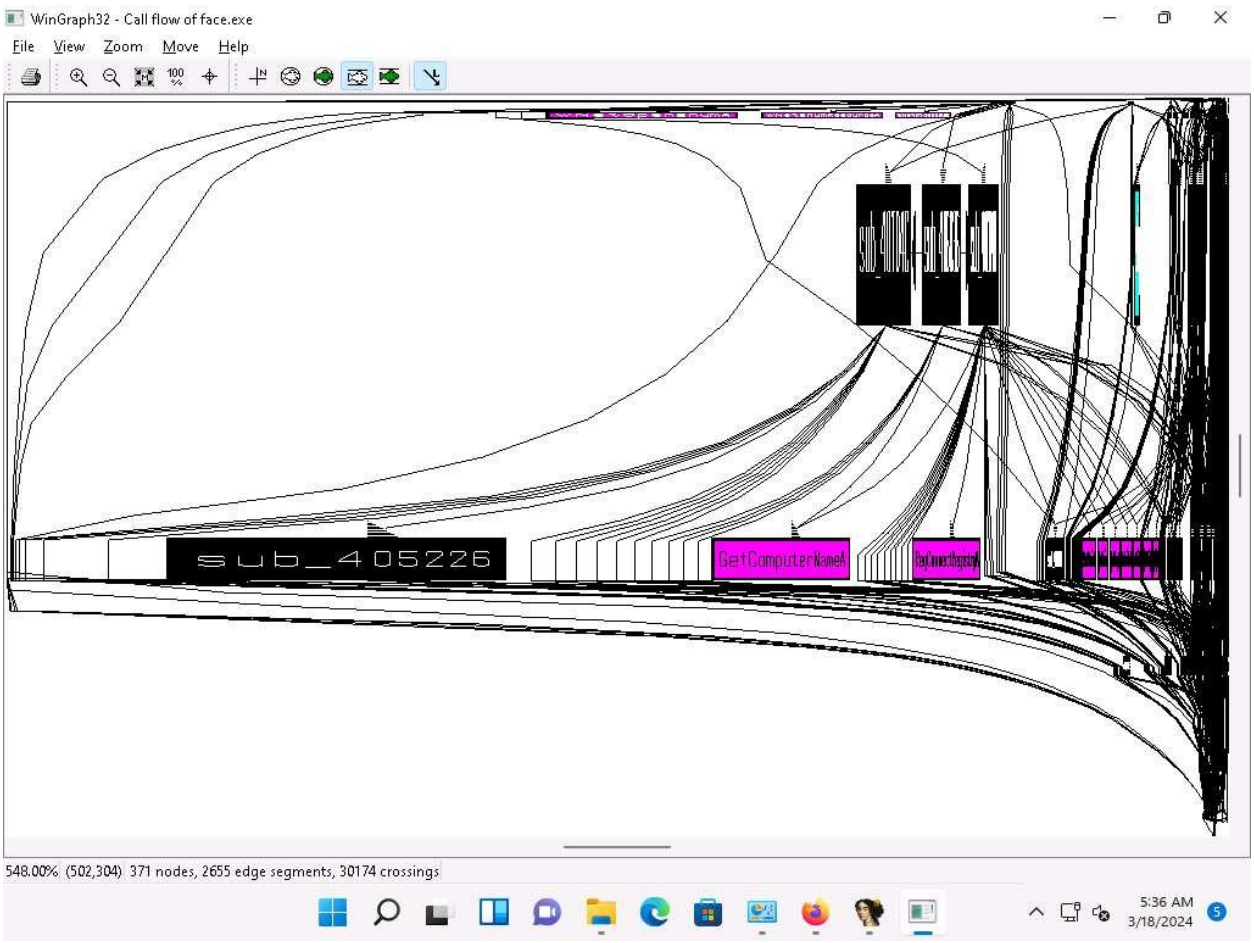


13. Close the **Graph** window, go to **View** --> **Graphs**, and click **Function calls** from the menu bar.

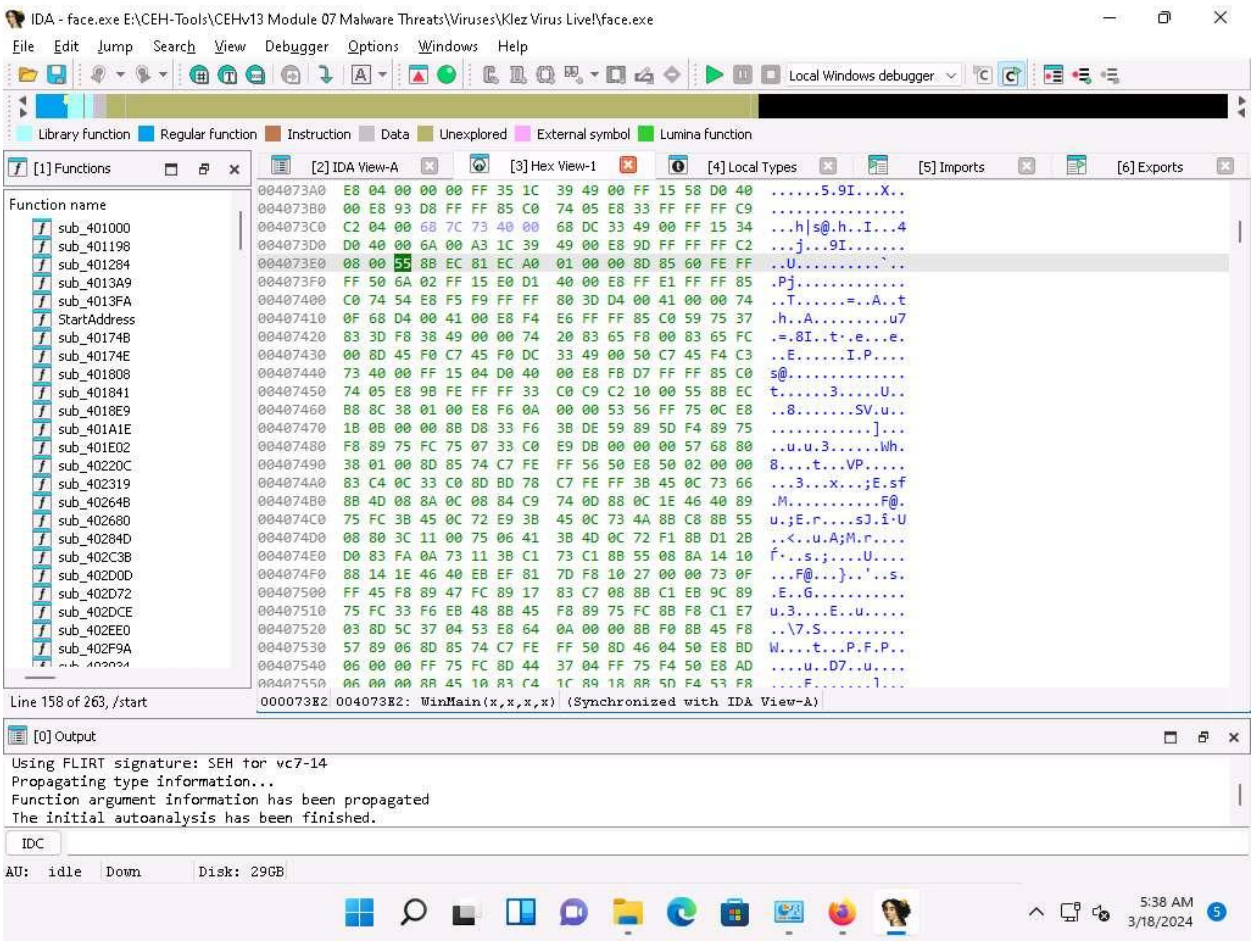


14. A window showing **call flow** appears; zoom in for a better view. Close the **WinGraph32 Call flow** window after completing the analysis.

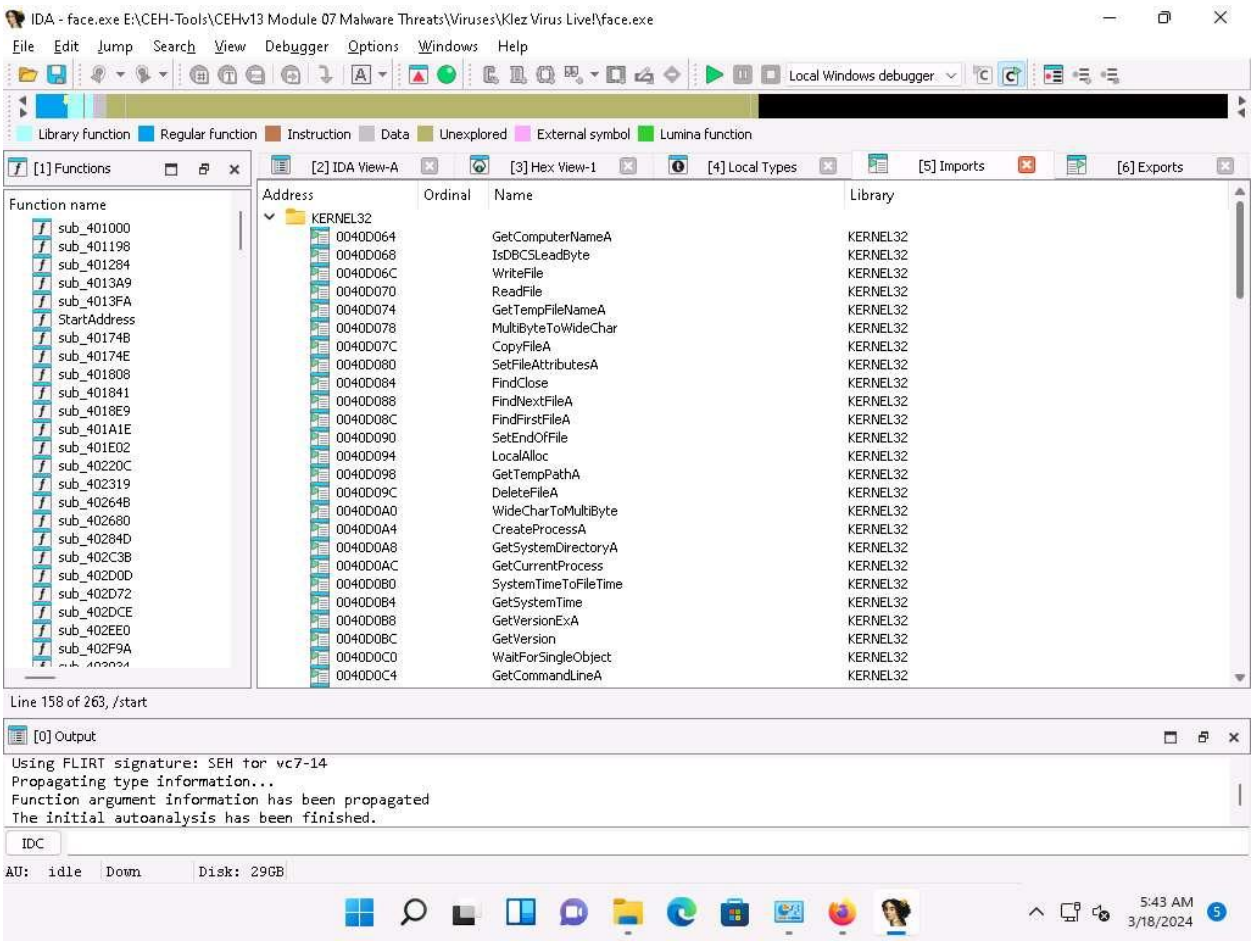

WinGraph32 - Call flow of face.exe
File View Zoom Move Help
5.31% (0,0) 371 nodes, 2655 edge segments, 30174 crossings
5:35 AM
3/18/2024



15. Click the **HexView-1** tab to view the hex value of the malicious file.

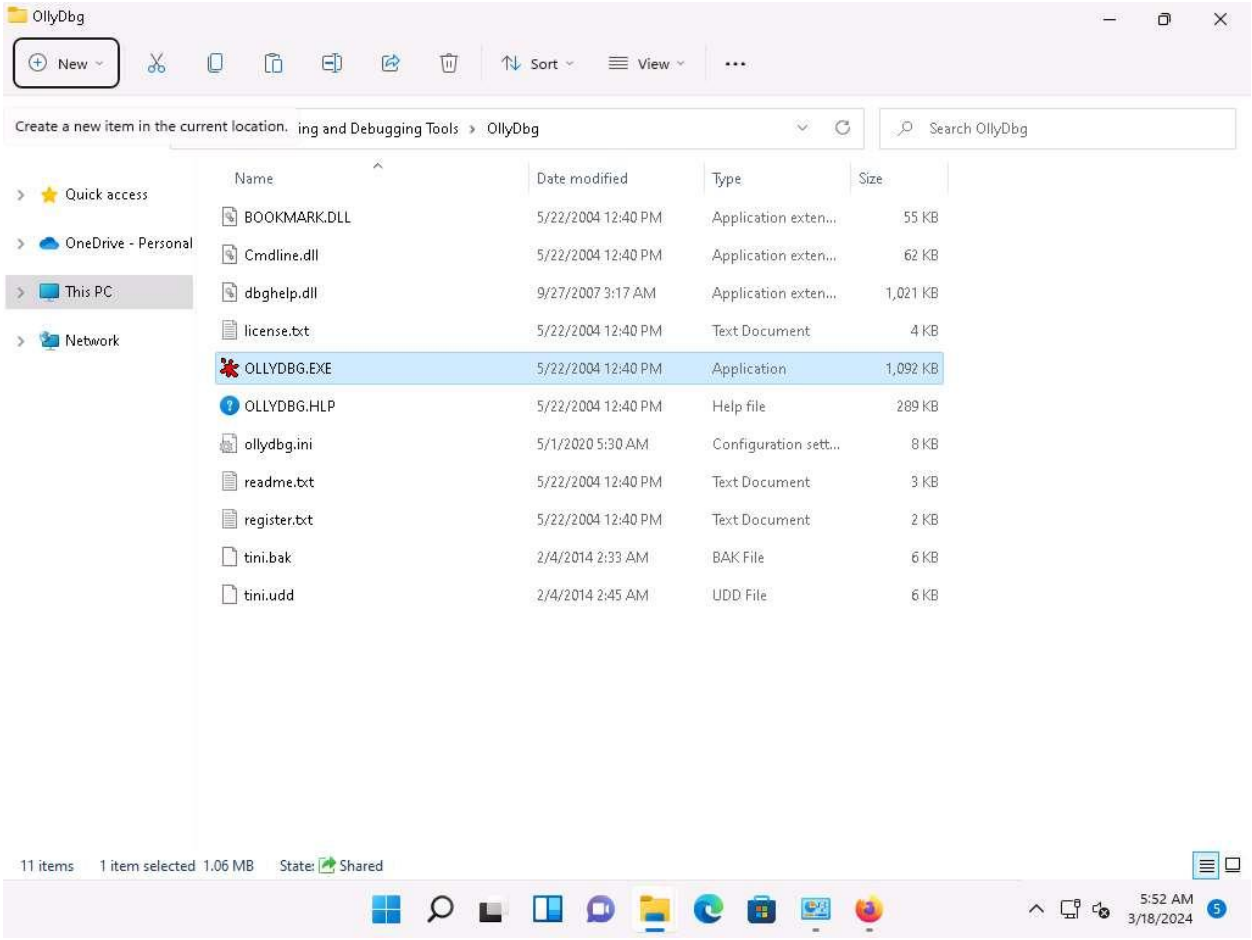


16. Click the **Imports** tab to view list of all functions that the executable calls.



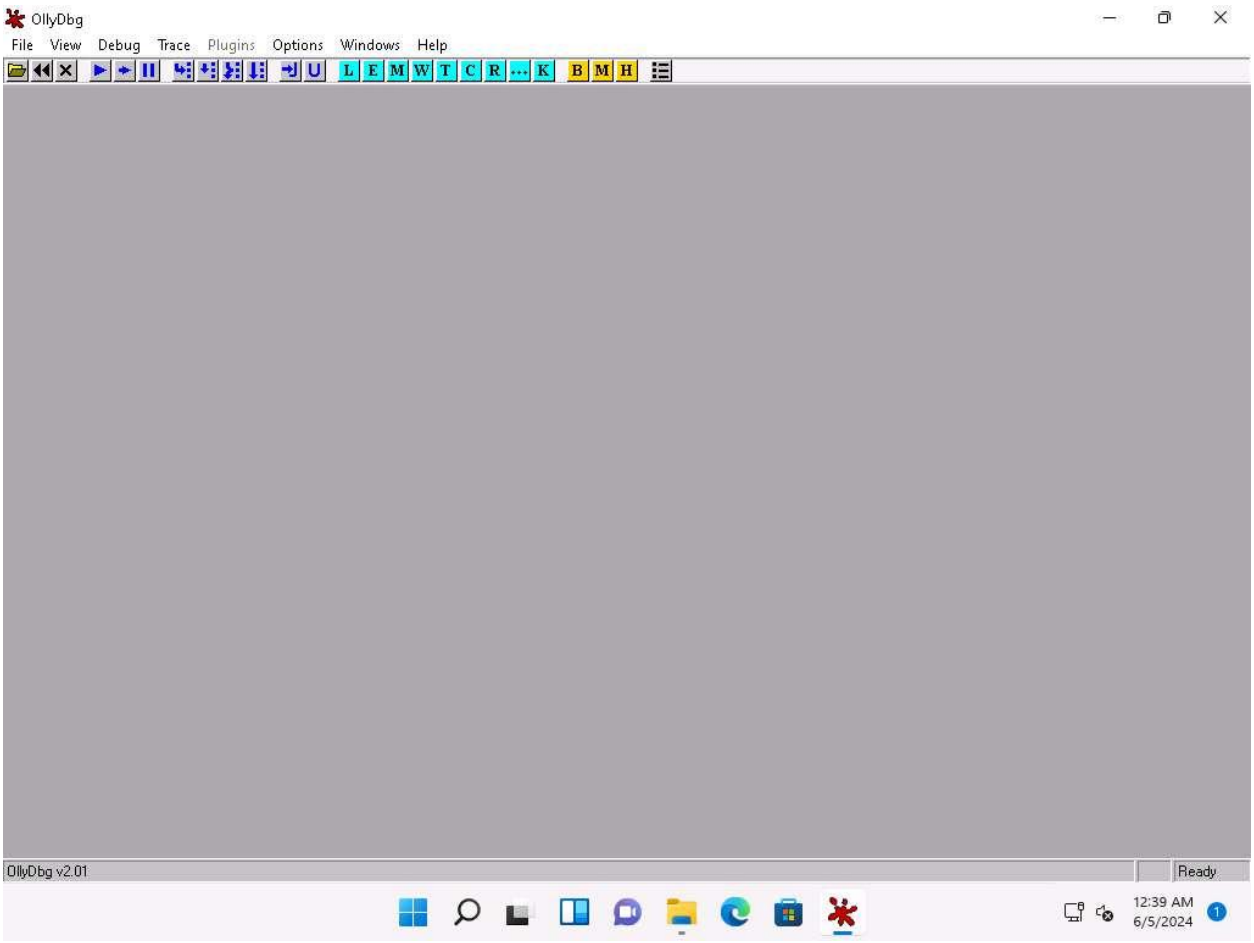
17. Close all open windows. In the **Save database** pop-up, click **OK**.
18. Navigate to **E:\CEH-Tools\CEHv13 Module 07 Malware Threats\Malware Analysis Tools\Static Malware Analysis Tools\Disassembling and Debugging Tools\OllyDbg** and double-click **Ollydbg.exe**.

If an **Open File - Security Warning** pop-up appears, click **Run**.

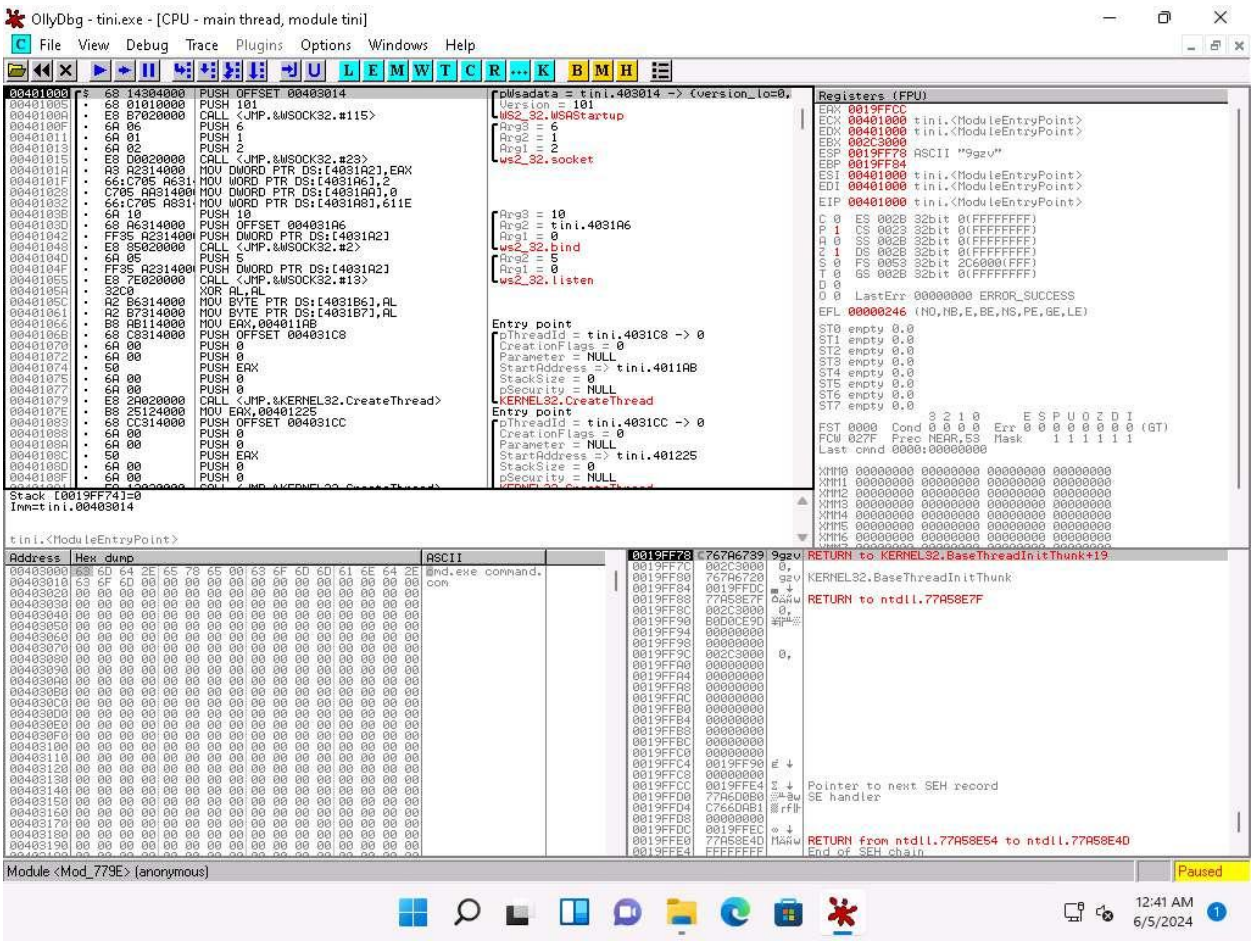


19. If a **Old DLL** dialog box appears, click **Yes**.
20. If an OllyDbg warning message appears, for administrative rights, click **OK**.
21. The **OllyDbg** main window appears, as shown in the screenshot.

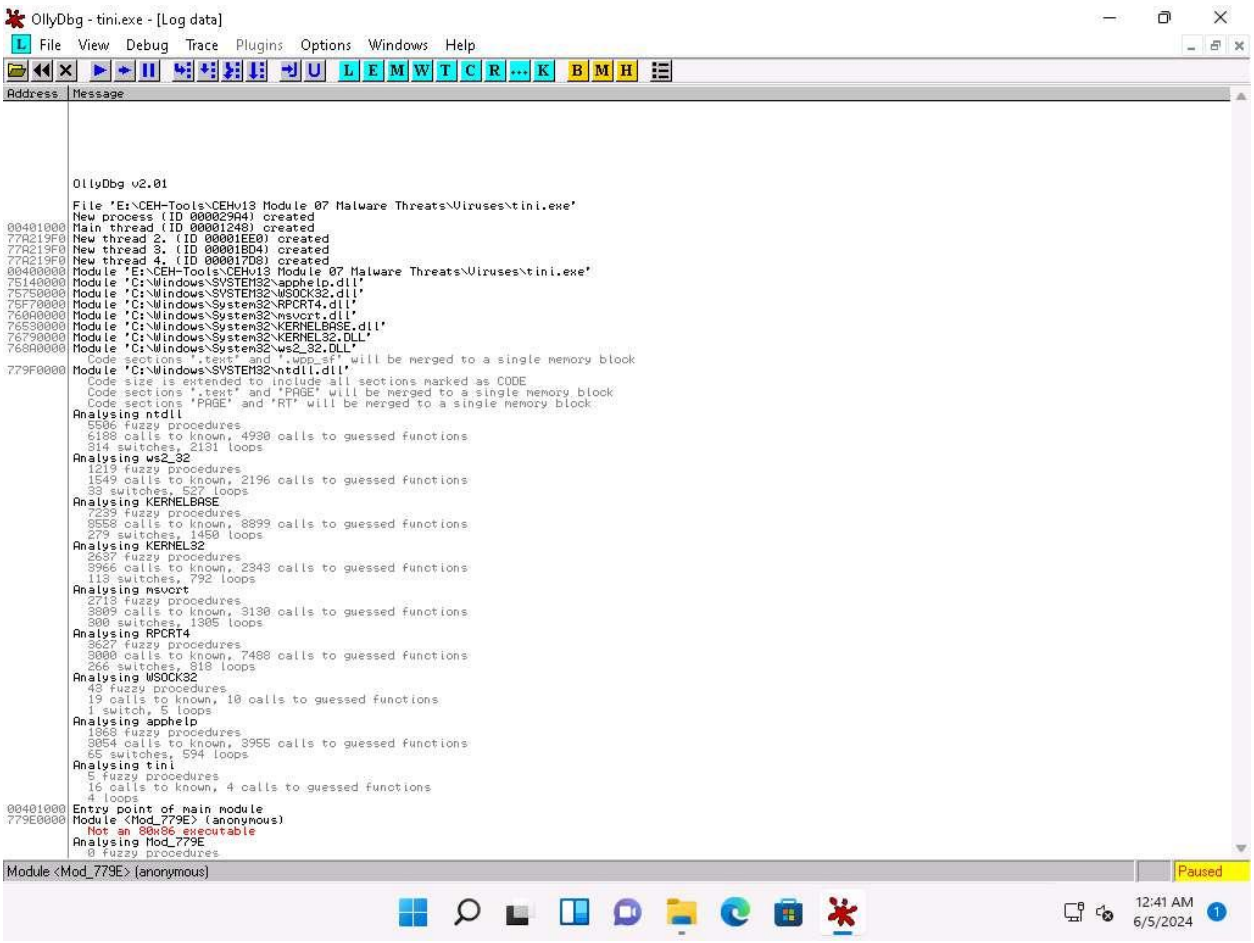
When you launch OllyDbg for the first time, several sub-windows might appear in the main window of OllyDbg; close all of them.



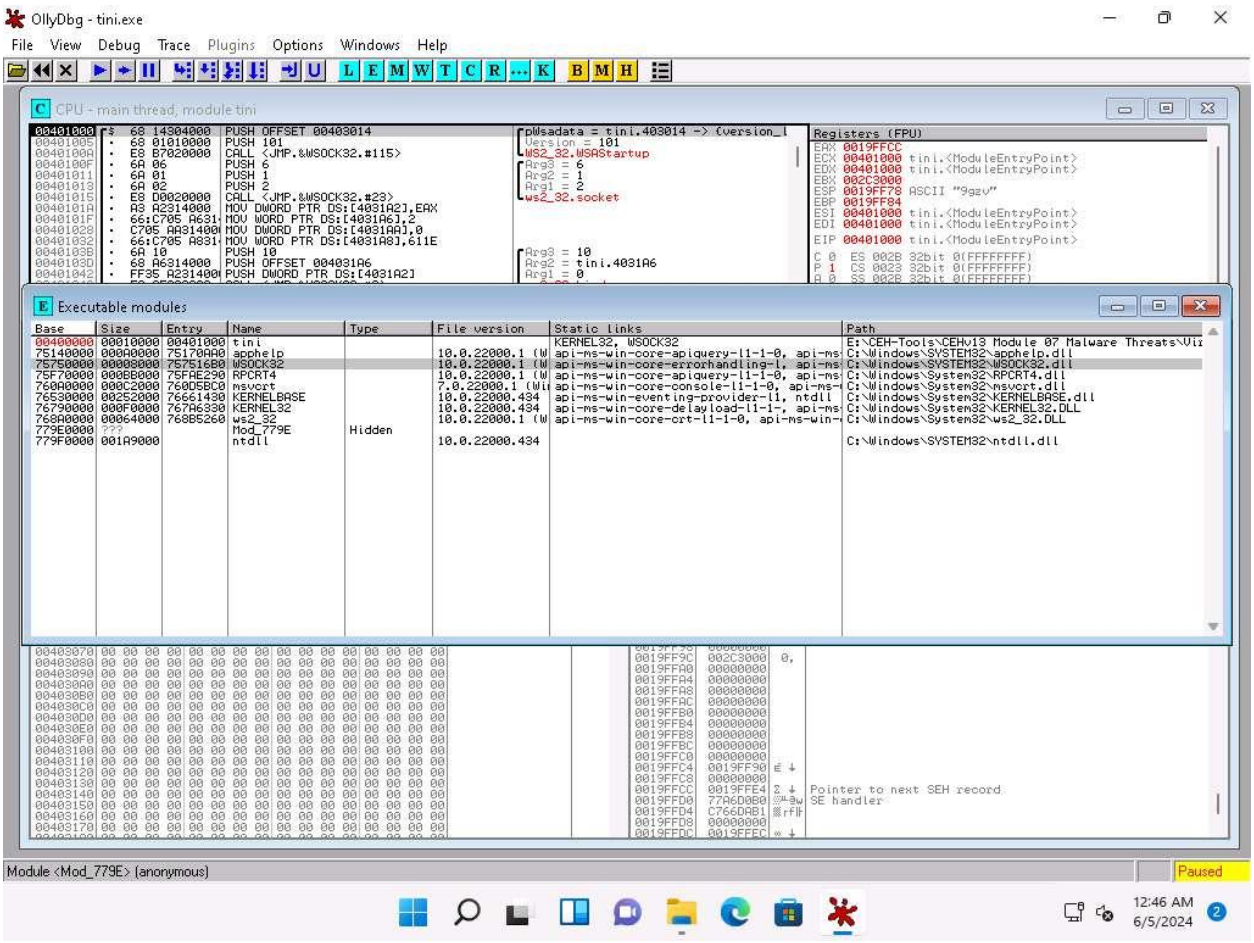
22. Choose **File** from the menu bar, and then choose **Open**.
23. The **Select 32-bit executable** window appears; navigate to **E:\CEH-Tools\CEHv13 Module 07 Malware Threats\Viruses**, select **tini.exe**, and click **Open**.
24. The output appears in a window named **CPU - main thread, module tini**, maximize the window.



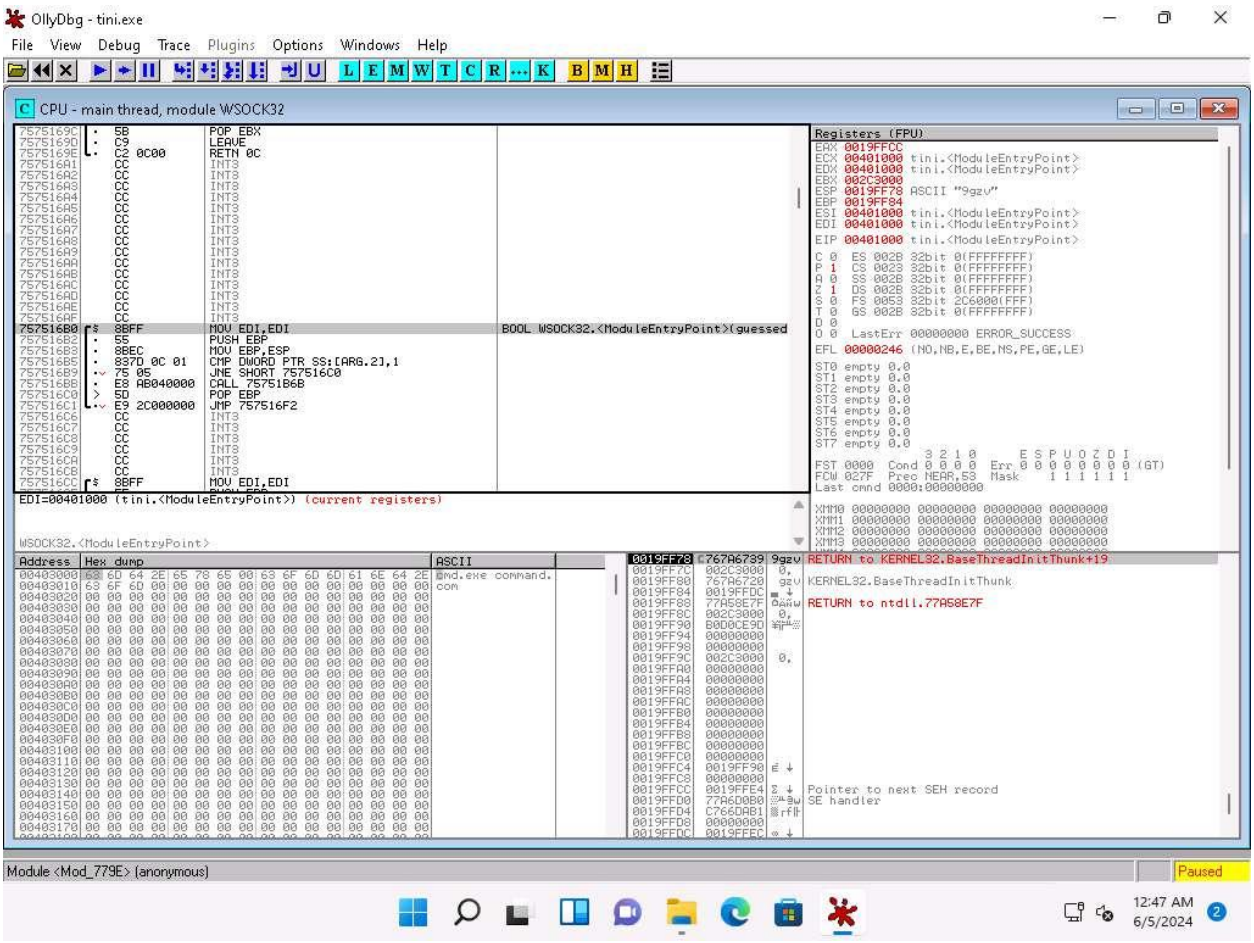
25. Choose **View** in the menu bar, and then choose **Log**.
26. A window named **Log data** appears in OllyDbg, displaying the log details.
27. The **Log data** also displays the program entry point and its calls to known functions. Close the **Log data** window after completing the analysis.



28. Choose **View** in the menu bar, and then choose **Executable modules**.
29. A window named **Executable modules** appears in OllyDbg, displaying all executable modules.
30. Double-click any module to view the complete information of the selected module.
31. In this task, we are choosing the **75750000** module. The results might differ when you perform this task.

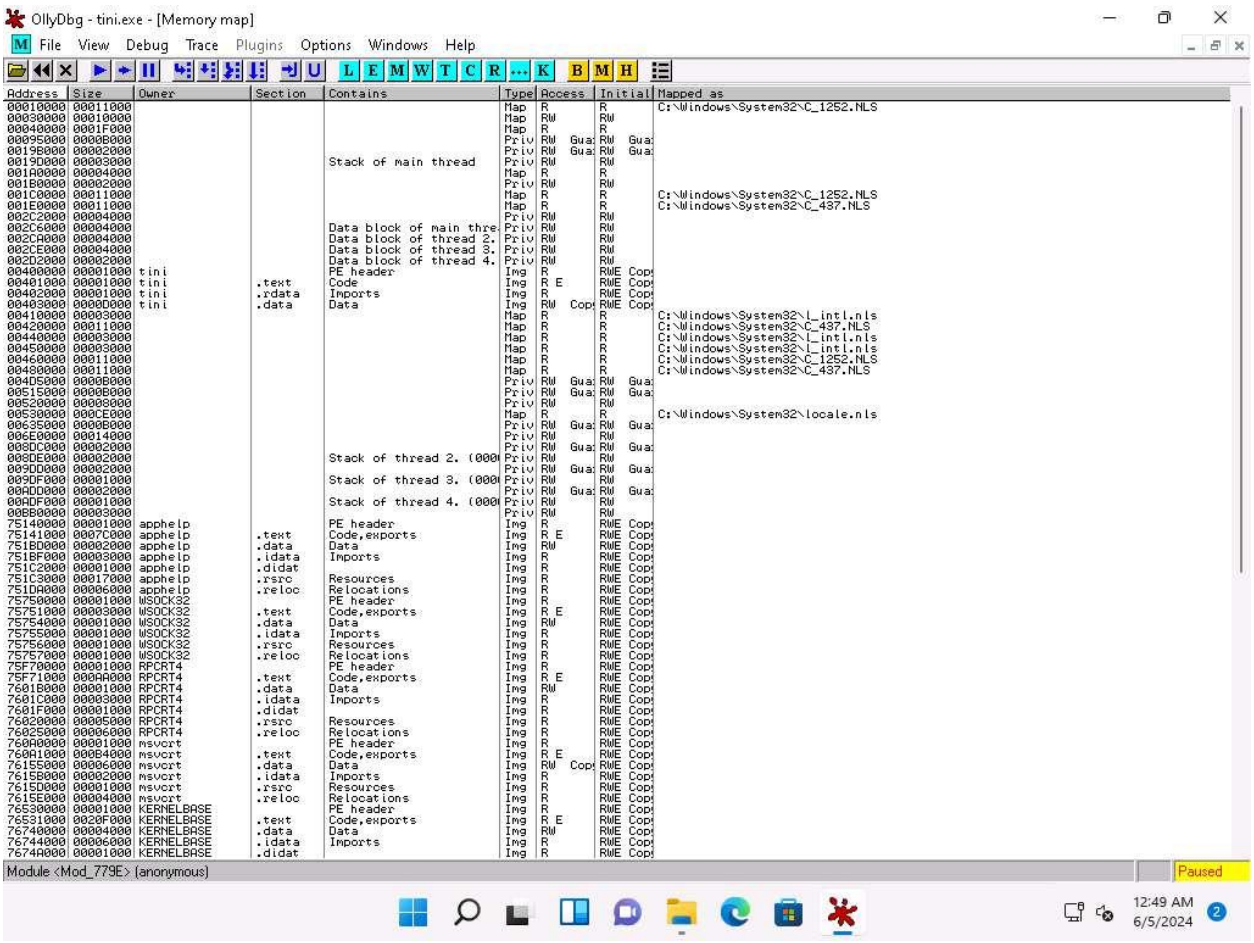


32. This will redirect you to the **CPU - main thread** window, as shown in the screenshot.



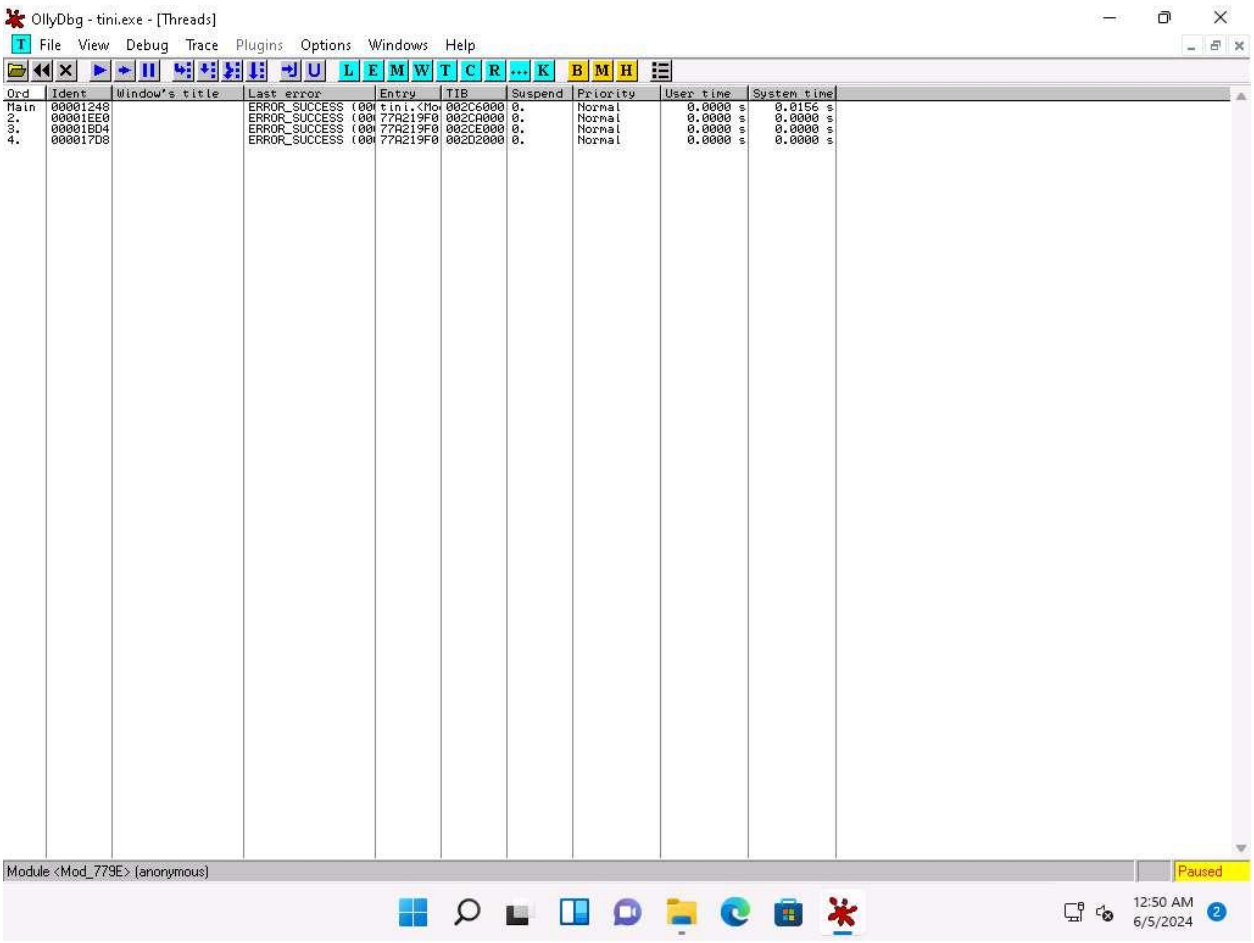
33. Choose **View** in the menu bar, and then choose **Memory map**.

34. A window named **Memory map** appears in OllyDbg, displaying all memory mappings, as shown in the screenshot. Close the **Memory map** window.



35. Choose **View** in the menu bar, and then choose **Threads**.

36. A window named **Threads** appears in OllyDbg, displaying all threads, as shown in the screenshot.



37. This way, you can scan files and analyze the output using OllyDbg.

38. Close all open windows.

Question 7.3.3.1

On the Windows 11 machine, use the IDA tool to analyze the file face.exe located in the directory E:\CEH-Tools\CEHv12 Module 07 Malware Threats\Viruses\Klez Virus Live!. What is the first subroutine function identified by IDA?