Malware Forensics

Module 11

Malware Forensics

Malware Forensics of a system refers to a set of procedures for identifying the presence of malware using various techniques.

Lab Scenario

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

Malware is the most prevalent threat to the computing industry currently. The attacker uses different types of malware to perform various actions. The investigators should be aware of malware programs, methods used for their propagation, their impacts on the systems and devices.

Attackers can also use different methods to hide the presence of malware to avert forensics investigations. An investigator must be able to spot the malware, extract and analyze it to find the perpetrator.

The labs in this module will provide the real-time experience in malware detection and analysis.

Lab Objectives

The objective of this lab is to help students in scanning an executable file to verify the presence of malicious components by performing thorough file analysis

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv9 Module 11 Malware Forensics

Perform Static and Dynamic file Analysis:

- Static Analysis:
  - Scan the file across various online and offline antivirus databases.
  - Calculate hash value of the file and check for its presence in the database of already identified malware.
  - Find the Portable executables.
  - Identify the packed, obfuscated or encrypted files.
  - Find the file dependencies.
  - Disassemble the file to discover the internal properties of the file.
  - Identify the internally stored passwords used for encryption.

- Dynamic Analysis.

Lab Environment

In this lab, you need:

- A computer running Windows Server 2012 host machine.
- A computer running Windows 10 virtual machine.
- A Web browser with Internet access.
- Administrative privileges to run tools and perform scans.

Lab Duration

Time: 125 Minutes

Overview of Malware Analysis

Malware analysis is a procedure for identifying hidden malice in any file and for studying its impact on the system, network, and other networking devices. Malware analysis helps to understand how a malware works, corrupt the files, the path of transmitting copies, use ports, connect to URLs, changes it makes to the system, etc. The process helps investigators find the malware that helped the perpetrators perform an attack on a network or a system and trace them back.

Lab Tasks

TASK 1

Overview

Recommended labs to perform Static and Dynamic analysis of any file to identify the presence of any malware:

- Perform Static Analysis of the Suspicious File.
- Dynamic Malware Analysis.
- Analyze Malicious PDF File.
- Scan suspicious MS-office file for Malice.

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure through public and free information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab

1

Perform Static Analysis of the Suspicious File

Static analysis is the process of scanning a file for the presence of malicious code or strings, and also verify its impact on the system without running or executing the file. This method is more secure and requires minor security.

Lab Scenario

Attackers design malware in such a way that it looks as useful file and to trick the users into installing it on the system, leading to a security breach of the system and network it is present in. The investigators should be aware of such tricks and know the methods of finding if a file consists malicious coding without executing it. In malware analysis, the first procedure is to scan the target file across for presence of identified malware programs and code. This lab will help in dealing with such situations.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Objectives

This lab will help you understand how to check any file for the presence of malicious codes and snippets without executing the program.

Objectives of this lab include:

- Scan the file across antivirus databases
- Find any hidden or encrypted program or code
- Extract PE headers from the executables
- Find strings that signify malicious coding
- Find linked libraries and dependencies
- Disassemble the code

Lab Environment

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv9 Module 11 Malware Forensics

To carry out the lab, you need:

- A computer running **Windows 10** virtual machine.
- The A web browser with internet access.
- PEiD located at **Z:\CHFIv9 Module 11 Malware Forensics\Malicious Document Analysis Tools\PEiD**.
- PEview located at **Z:\CHFIv9 Module 11 Malware Forensics\PE Data Acquisition Tools\PEView**.
- Strings.exe located at **Z:\CHFIv9 Module 11 Malware Forensics\Strings Search Tools\Strings**.
- Dependency Walker located at **Z:\CHFIv9 Module 11 Malware Forensics\PE Data Acquisition Tools\Dependency Walker**.
- IDA Pro located at **Z:\CHFIv9 Module 11 Malware Forensics\Malware Analysis Tools\IDA Pro**.
- Administrative privileges to run the tools.

Lab Duration

Time: 40 Minutes

Lab Tasks

TASK 1

Scan the File across Antivirus Databases

1. Go to any browser and type in the following URL: **https://www.virustotal.com**.
2. Click on the **Choose File** button to upload the suspicious file.

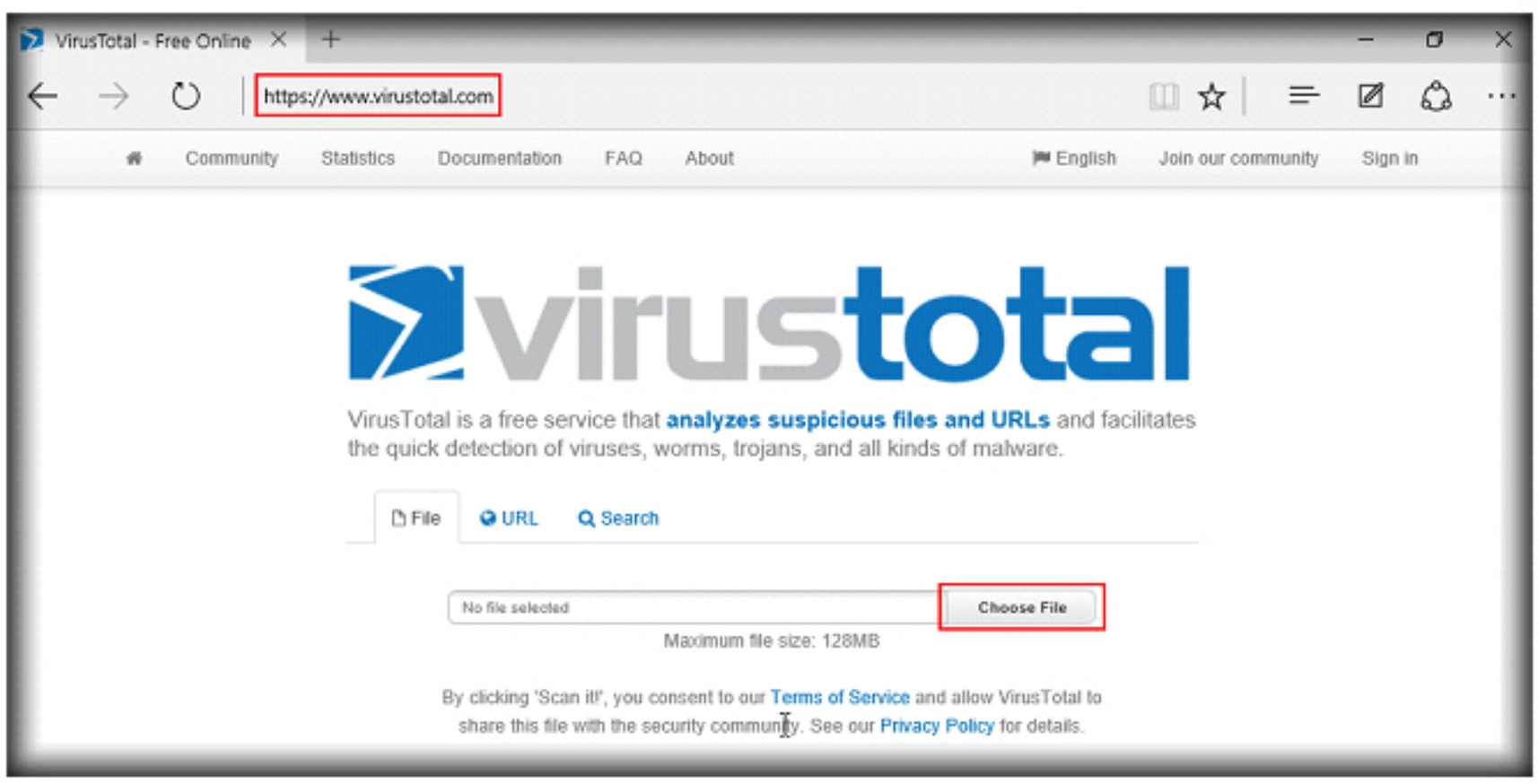


FIGURE 1.1: Using VirusTotal website to scan the suspicious file

3. Navigate to the following location: **Z:\Evidence Files\Unknown**, select the file Unknown.exe and click the **Open** button.

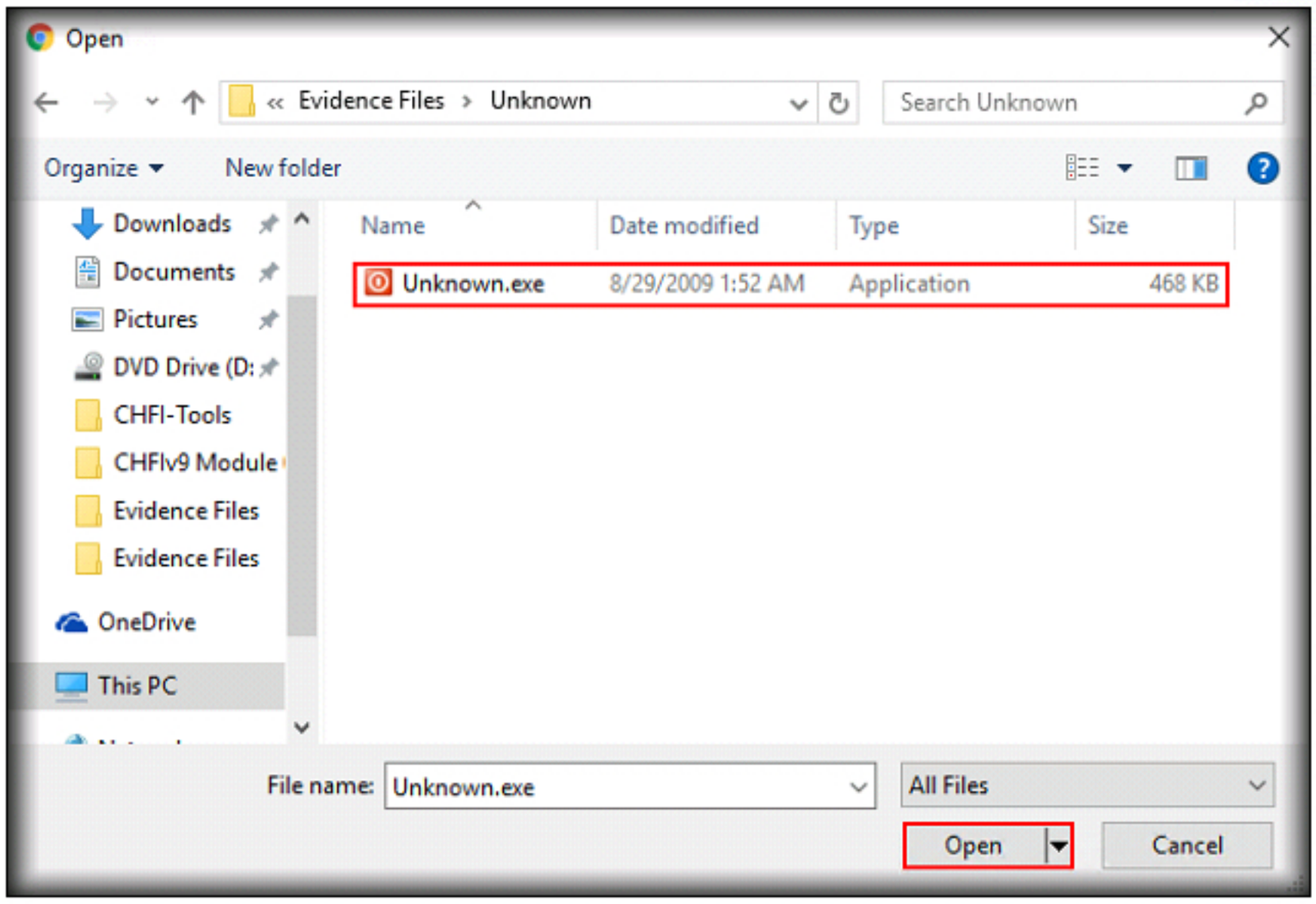


FIGURE 1.2: Navigating to the folder and selecting the sample suspicious file

4. Click the **Scan it!** button.

FIGURE 1.3: Scanning the suspicious file using VirusTotal

5. The website may generate a dialog box that displays a message, which provides information about the time when the website analyzed the file last time and first time.

   The dialog box will also mention the detection ration, which displays the ratio of anti-virus companies that have voted the file as malicious to the total number of companies collaborated with the website.

   Click the **Reanalyse** button to perform the analysis.

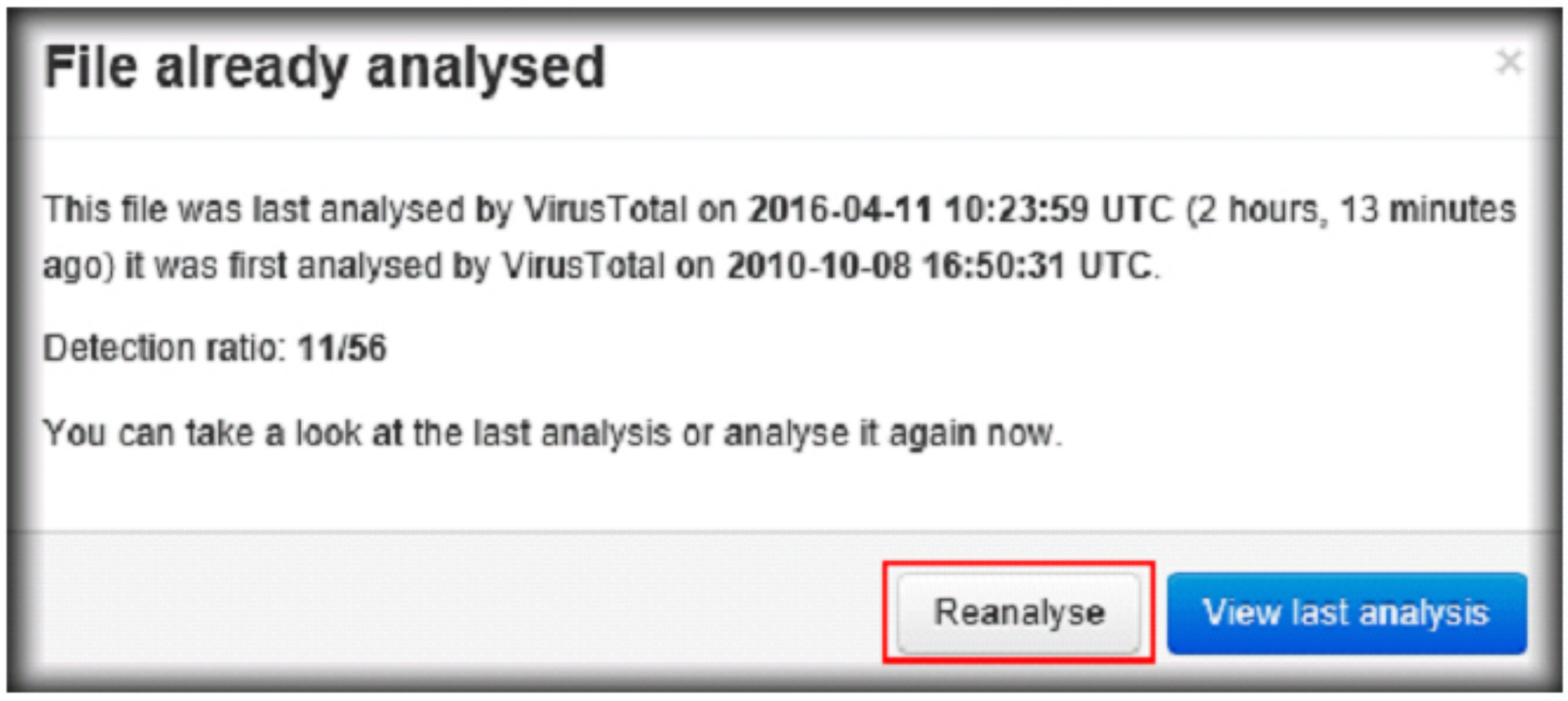


FIGURE 1.4: Scanning the suspicious file

6. The website will display various details of file such as, SHA256 hash value and the names of the companies that have voted the file as malicious.

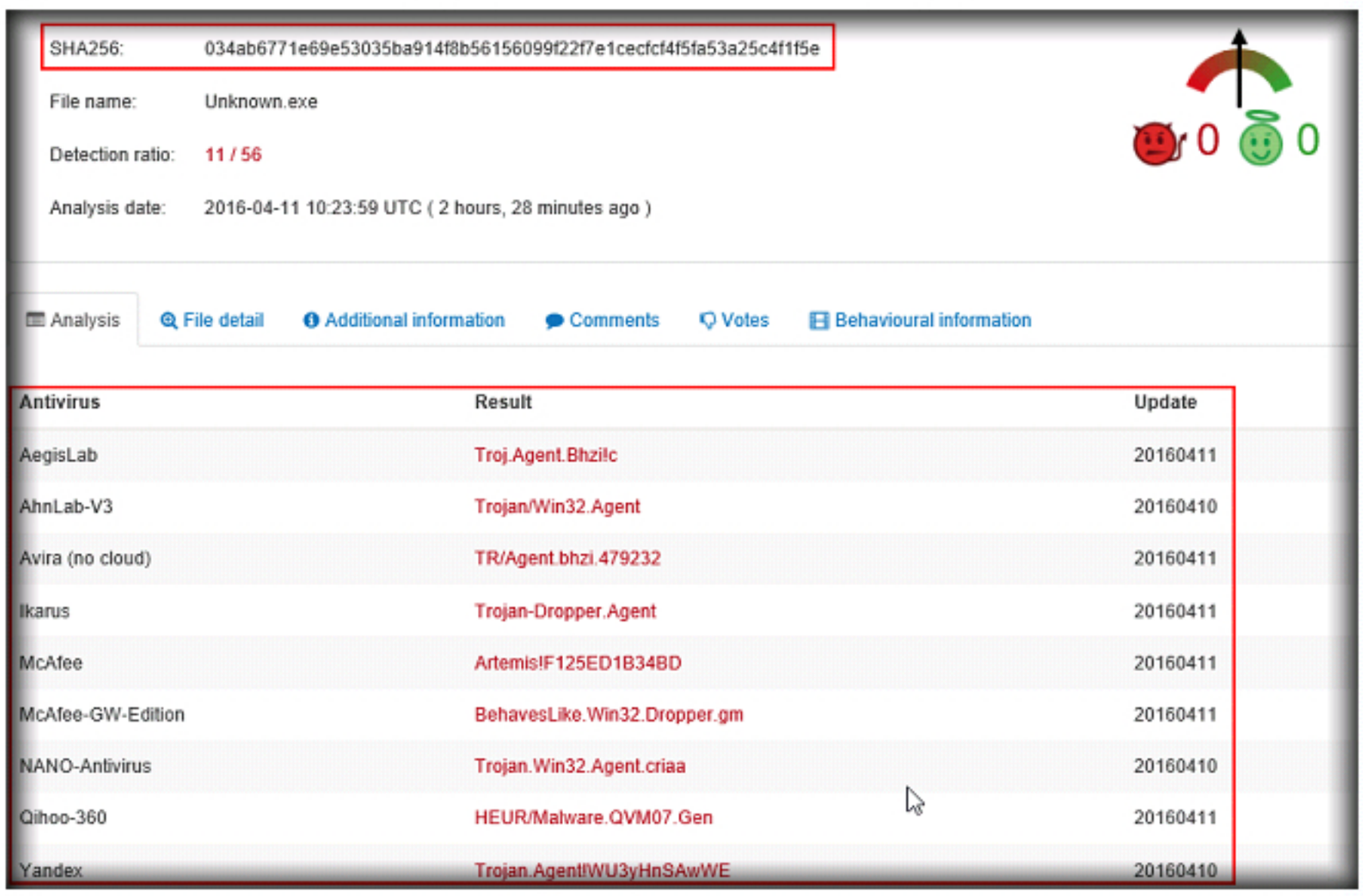


FIGURE 1.5: Screenshot displaying the scanned results

7. Click on the **File detail** tab for further relevant details of the online file analysis such as compilation time stamp, type of file, compatible processors, entry point and PE sections.

Analysis | File detail | Additional information | Comments | Votes | Behavioural information

The file being studied is a Portable Executable file! More specifically, it is a Win32 EXE file for the Windows GUI subsystem.

Packers identified

PEiD	InstallShield 2000

PE header basic information

Target machine	Intel 386 or later processors and compatible processors
Compilation timestamp	2005-01-30 07:50:29
Entry Point	0x000271B0
Number of sections	6

PE sections

Name	Virtual address	Virtual size	Raw size	Entropy	MD5
.text	4096	299517	299520	6.60	8844f67e881fa912ed3fdcd47cd92297
.rdata	307200	49032	49152	4.36	724ca494353ef626e6069b7c70ea42c3
.data	356352	36384	22528	3.20	83173ed6248130863c66e2cfcaf8833a

FIGURE 1.6: File detail tab showing further details of the scanned file

8. The result also provides information about the data link libraries (DLLs) that the programs may use and the types of PE resources it includes.

PE imports

[+] ADVAPI32.dll
[+] COMCTL32.dll
[+] GDI32.dll
[+] KERNEL32.dll
[+] OLEAUT32.dll
[+] SHELL32.dll
[+] USER32.dll
[+] WINMM.dll
[+] WINSPOOL.DRV
[+] comdlg32.dll
[+] ole32.dll
[+] oledlg.dll

Number of PE resources by type

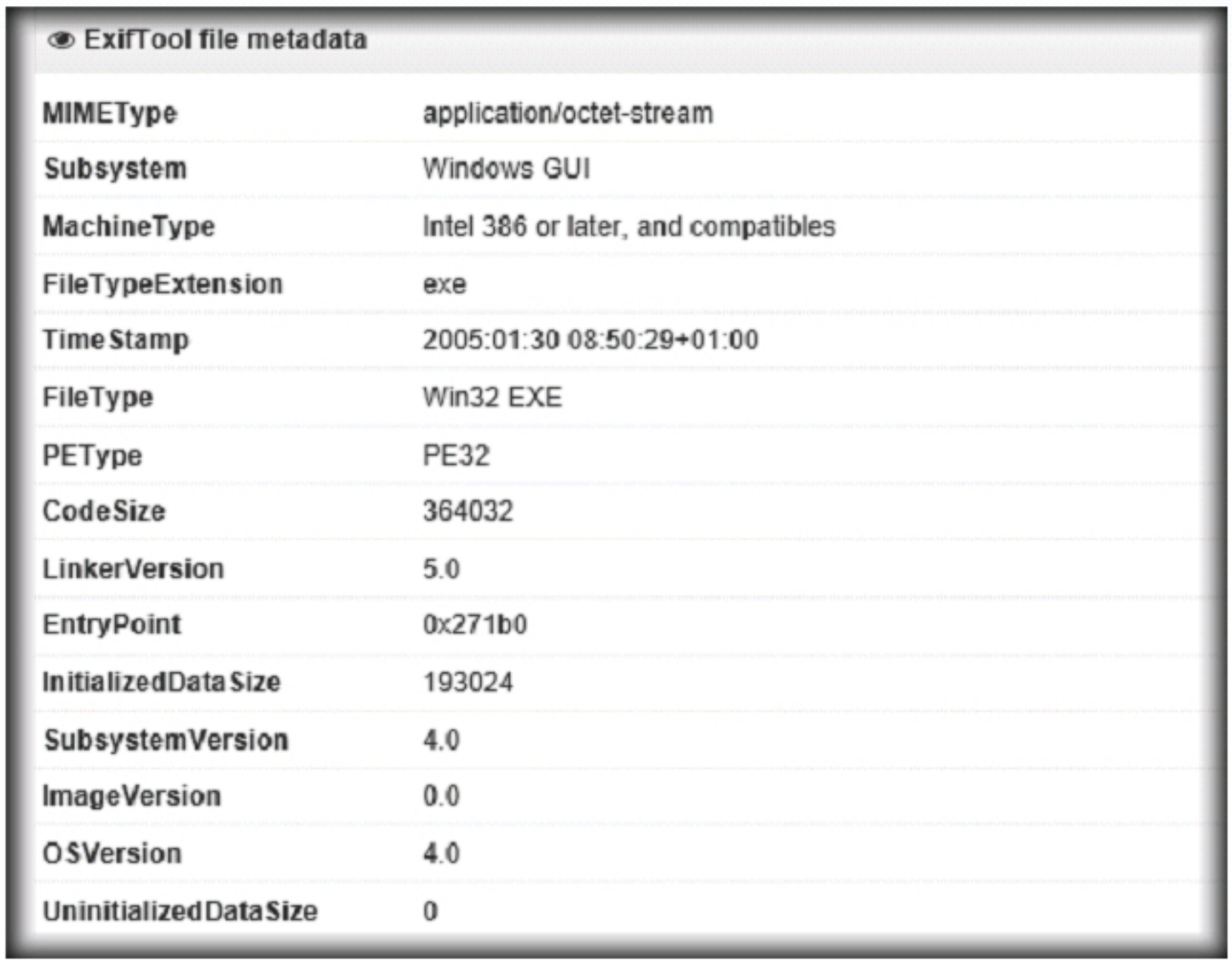
RT_ICON	12
RT_GROUP_ICON	1
RT_RCDATA	1

Number of PE resources by language

NEUTRAL DEFAULT	14

FIGURE 1.7: Screenshot showing the information about DLLs and PE resources

9. The website also displays the metadata present in the file.

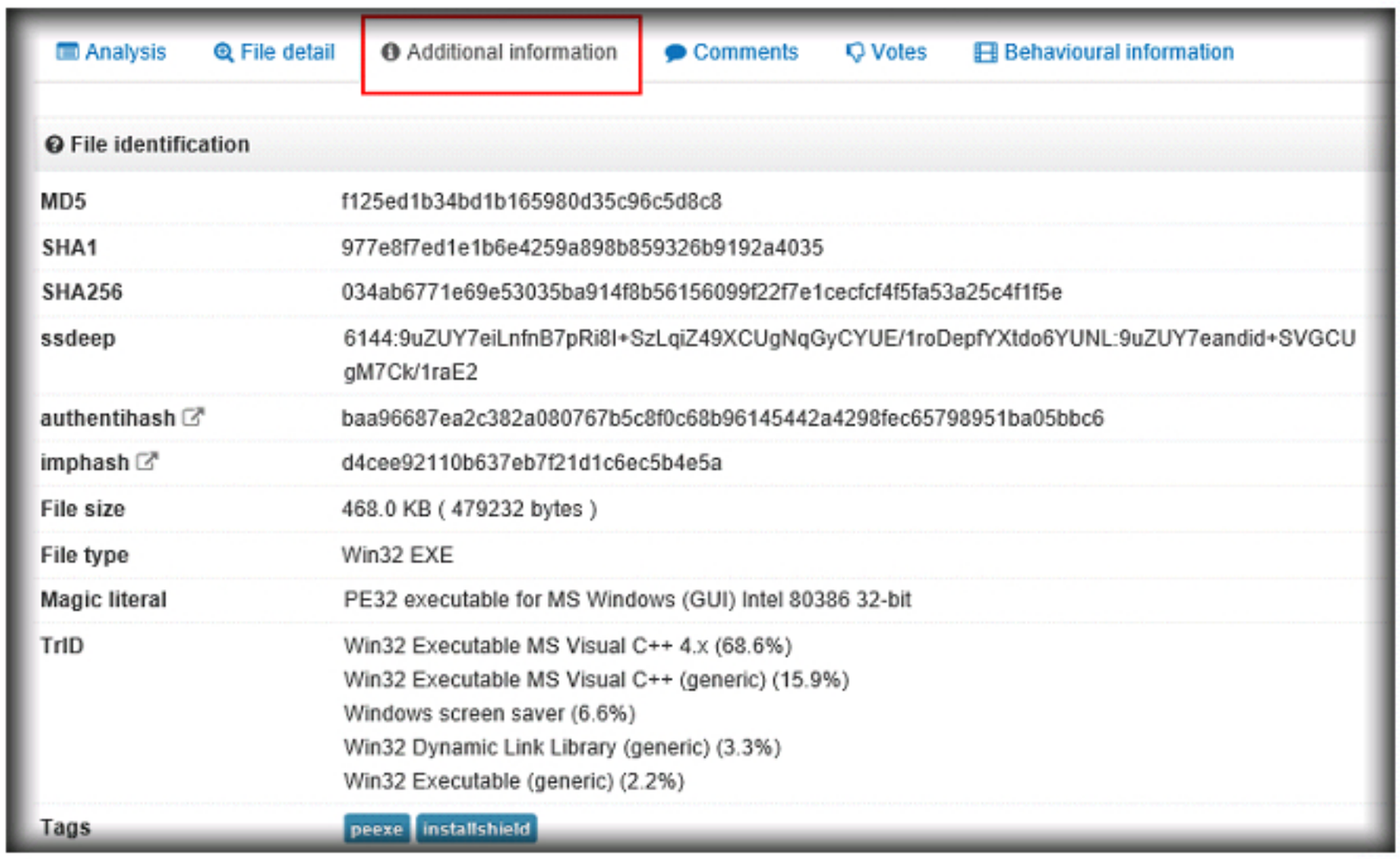


ExifTool file metadata

MIMEType	application/octet-stream
Subsystem	Windows GUI
MachineType	Intel 386 or later, and compatibles
FileTypeExtension	exe
TimeStamp	2005:01:30 08:50:29+01:00
FileType	Win32 EXE
PEType	PE32
CodeSize	364032
LinkerVersion	5.0
EntryPoint	0x271b0
InitializedDataSize	193024
SubsystemVersion	4.0
ImageVersion	0.0
OSVersion	4.0
UninitializedDataSize	0

FIGURE 1.8: Screenshot showing metadata present in the file

10. In the Additional information tab, the result displays the different hash values of the file including MD5, SHA1, SHA256, etc. Other details include TrID and tags.



Analysis | File detail | Additional information | Comments | Votes | Behavioural information

File identification

MD5	f125ed1b34bd1b165980d35c96c5d8c8
SHA1	977e8f7ed1e1b6e4259a898b859326b9192a4035
SHA256	034ab6771e69e53035ba914f8b56156099f22f7e1cecfcf4f5fa53a25c4f1f5e
ssdeep	6144:9uZUY7eiLnfnB7pRi8I+SzLqiZ49XCUgNqGyCYUE/1roDepfYXtdo6YUNL:9uZUY7eandid+SVGCUgM7Ck/1raE2
authentihash	baa96687ea2c382a080767b5c8f0c68b96145442a4298fec65798951ba05bbc6
imphash	d4cee92110b637eb7f21d1c6ec5b4e5a
File size	468.0 KB (479232 bytes)
File type	Win32 EXE
Magic literal	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
TrID	Win32 Executable MS Visual C++ 4.x (68.6%) Win32 Executable MS Visual C++ (generic) (15.9%) Windows screen saver (6.6%) Win32 Dynamic Link Library (generic) (3.3%) Win32 Executable (generic) (2.2%)
Tags	peexe installshield

FIGURE 1.9: Viewing Additional information section of the file

11. The Behavioural information tab holds information of runtime DLLs the suspicious program may call during execution, device drivers it uses, as well as the IP addresses the program tries to connect with and the protocol it deploys.

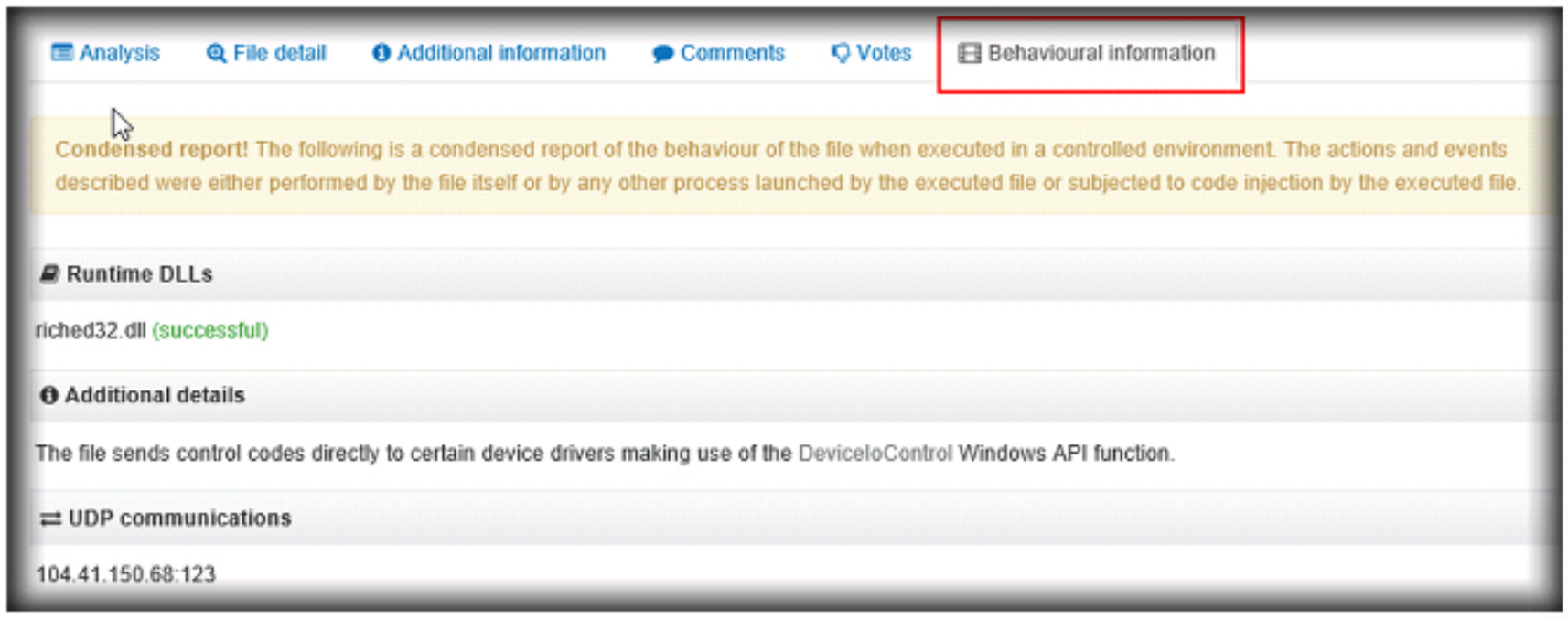


FIGURE 1.10: Viewing the Behavioural information section of the file

TASK 2

Find Obfuscated Program or Code

12. Investigators need to find if the file has any hidden or encrypted program or code within as well as find its type and packer used to encrypt it. In this lab, we will use the PEiD tool, which is an open source detection tool that detects packers, cryptors and compilers stored inside the PE files.
13. To run the PEiD tool, navigate to the **Z:\CHFIv9 Module 11 Malware Forensics\Malicious Document Analysis Tools\PEiD** location and double-click the **PEiD.exe** file.
14. PEiD GUI appears; to add a target file to the tool, click the ellipsis (**...**) associated with the **File** option, to upload the respective evidence file.

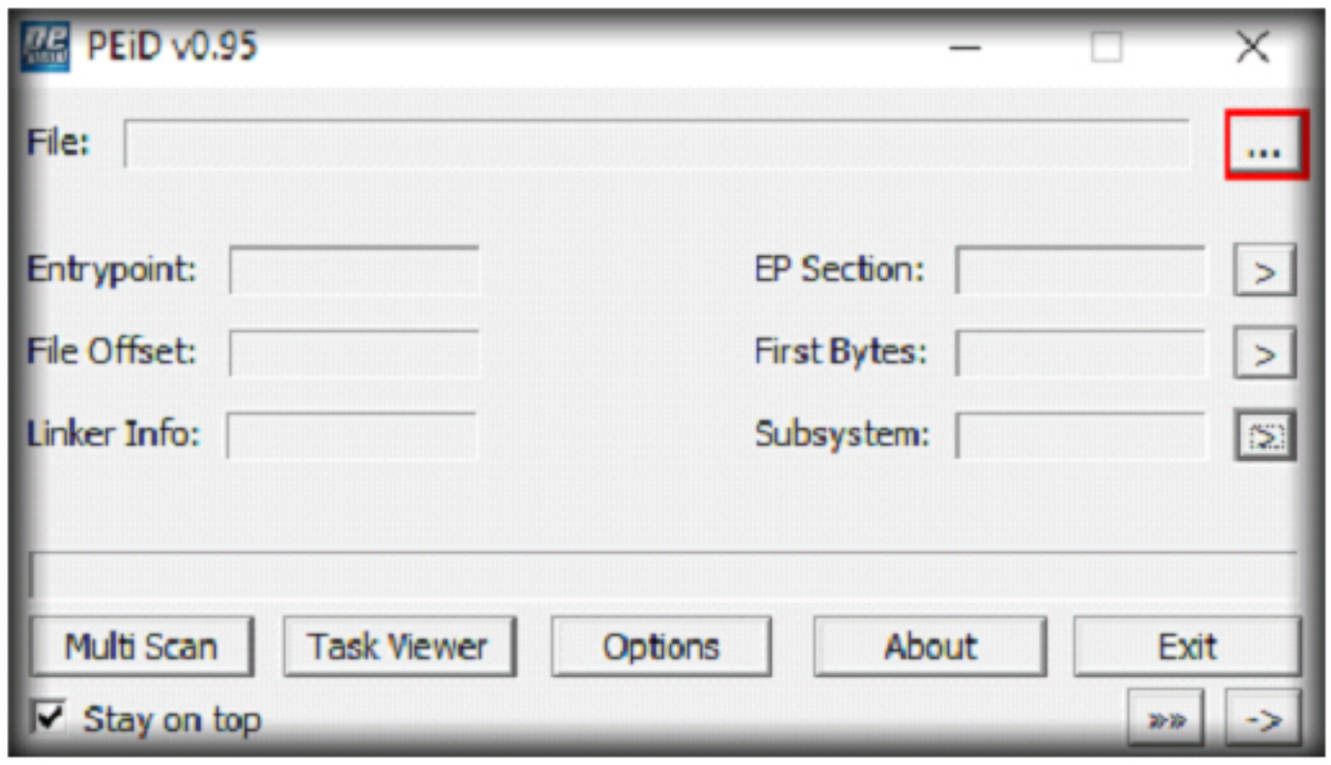


FIGURE 1.11: Adding target file in PEiD tool

15. Navigate to the following location: **Z:\Evidence Files\Unknown**, select the file **Unknown.exe** and click the **Open** button.

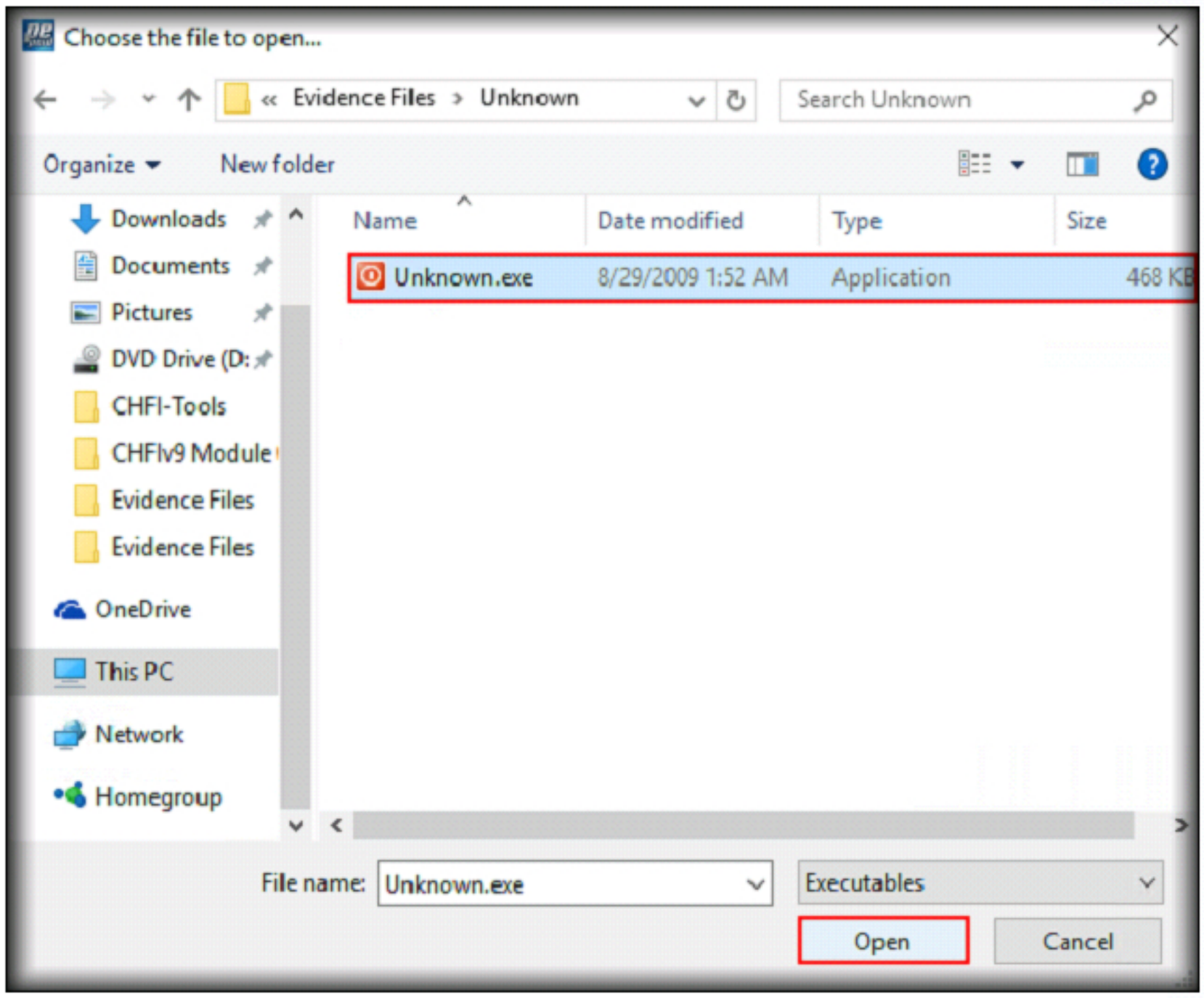


FIGURE 1.12: Navigating to the target file

16. The PEid tool will display the extension of obfuscated file if any in the EP section and display the name of packer used to encrypt the file at the bottom.

17. In this case, the tool was able to detect only **.text** files which show that there is no obfuscated file present in the suspicious file.

    The file uses **Microsoft Visual C++5.0** tool to pack the contents.

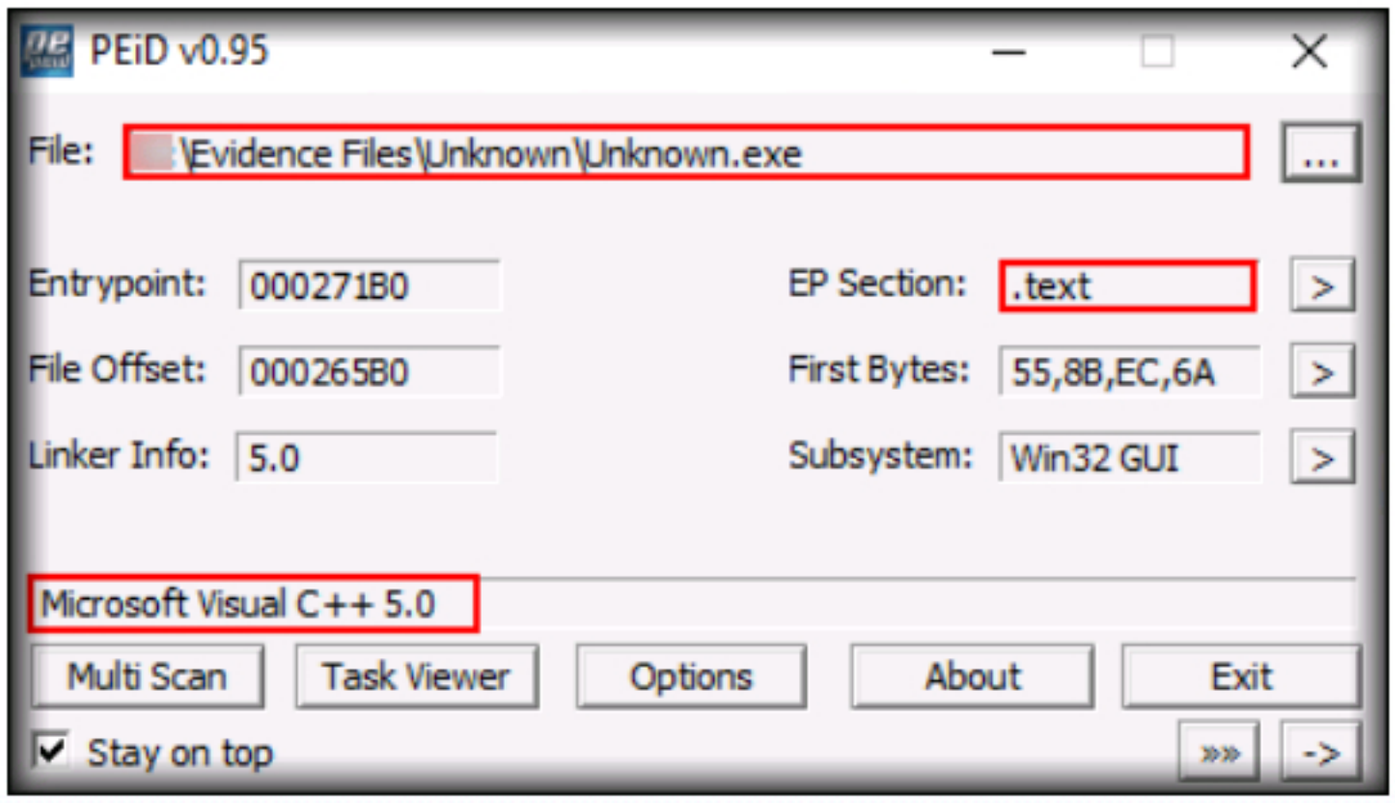


FIGURE 1.13: Viewing results in PEiD tool

TASK 3

Extract PE Headers of the File

18. Since the file is not packed and encrypted, the next step would be to extract its **PE headers** using the **PEview** tool. So, exit the PEiD tool.
19. To launch the **PEview** tool on the system, navigate to the **Z:\CHFIv9 Module 11 Malware Forensics\PE Data Acquisition Tools\PEView** and double click the **PEview.exe** file.
20. A **PEview License** pop-up appears, click **Accept**.
21. The PEview tool will start running and prompt an **Open** dialog box to add the target file. Navigate to the evidence file location **Z:\Evidence Files\Unknown**, select the **Unknown.exe** file and click the **Open** button.

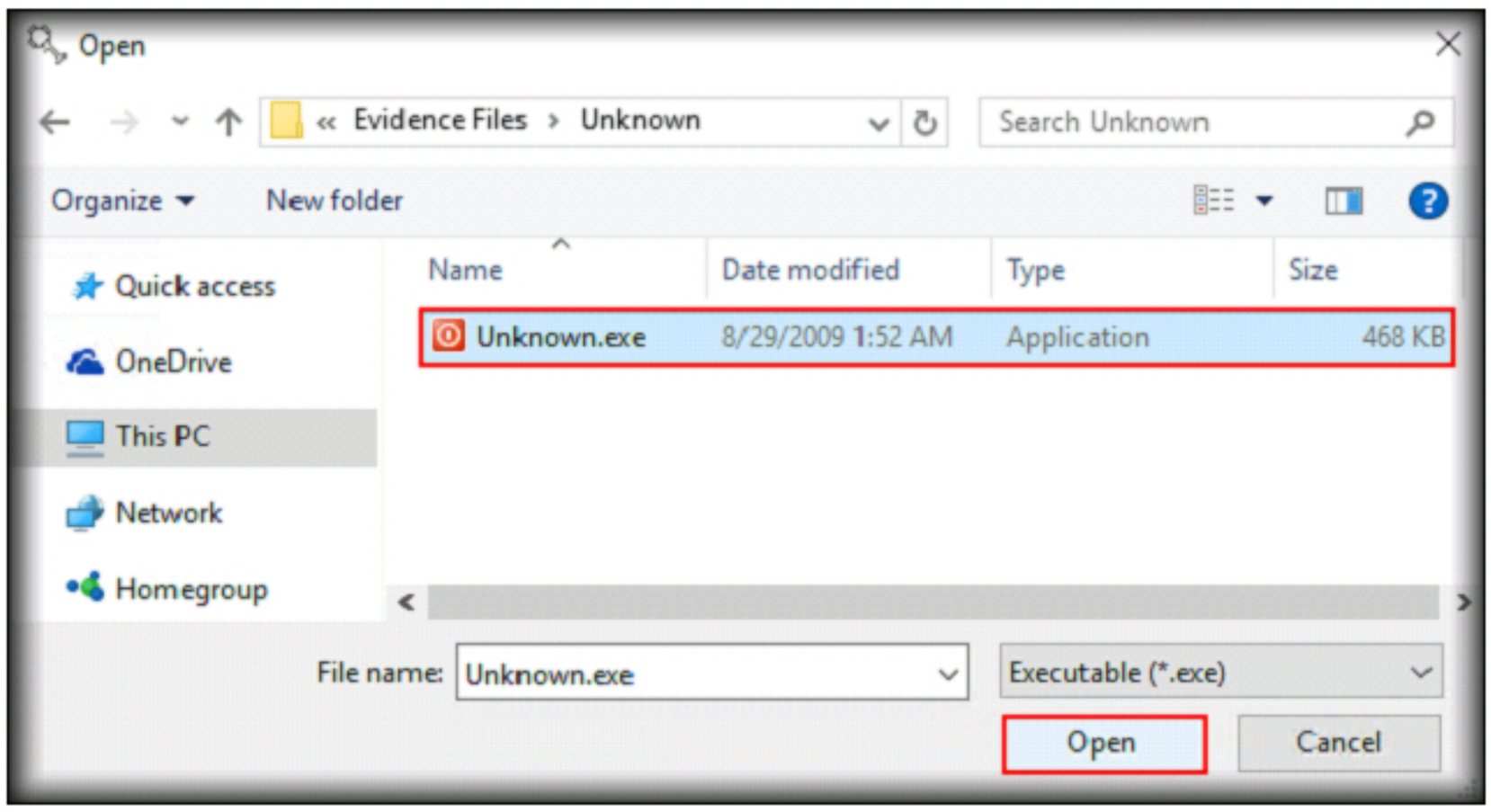


FIGURE 1.14: Navigating to the target file in PEview tool

22. The tool will display the hex data of the file. Click the **IMAGE_NT_HEADER** from the left column and from the drop-down list, select **IMAGE_FILE_HEADER** to view the header data of the file.

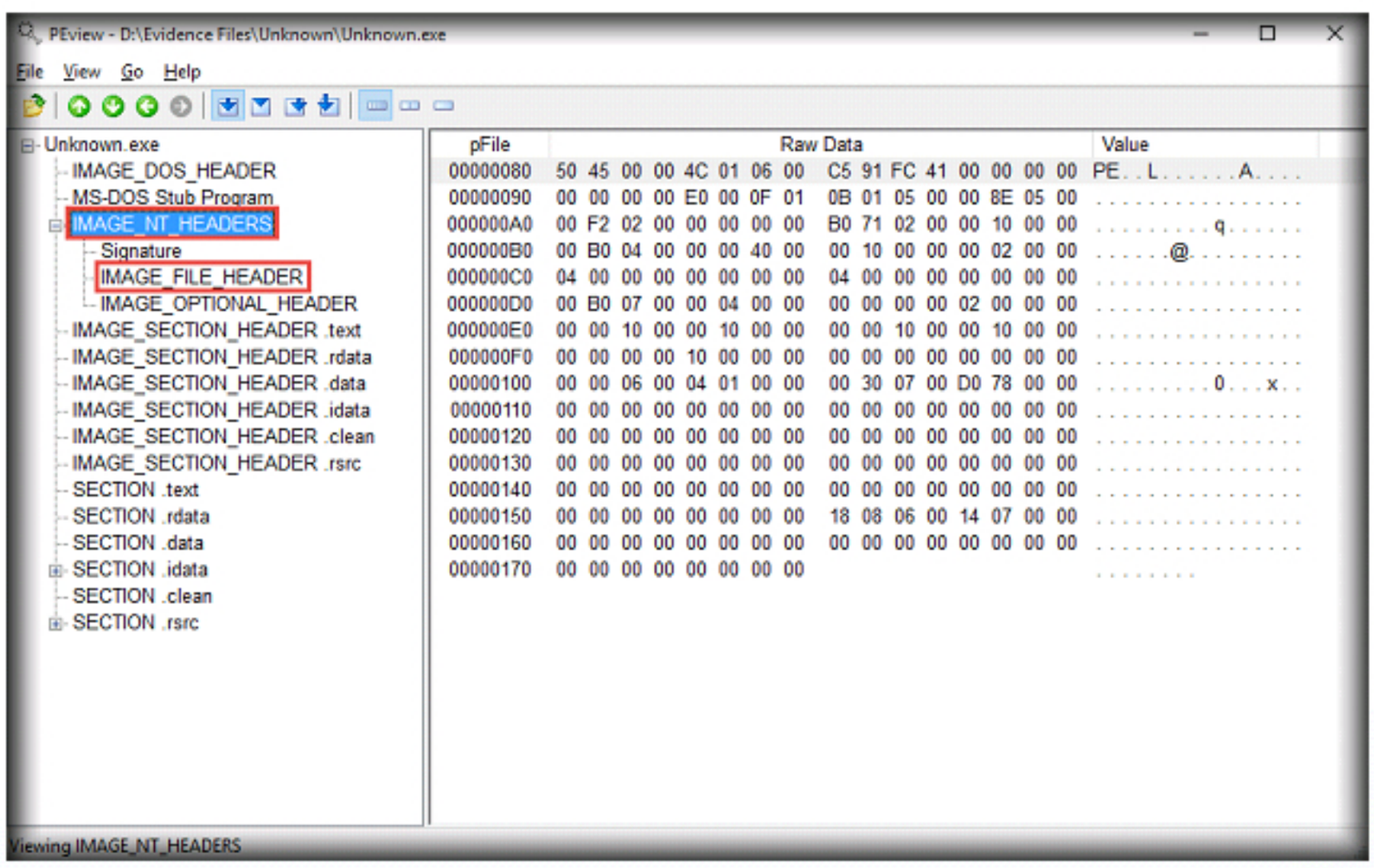


FIGURE 1.15: Viewing header data of the selected file in PEview tool

23. The header contains basic information about the file such as date timestamp description, compilation time, etc.
24. The information also helps the investigator to know if the file contains a GUI or console program.

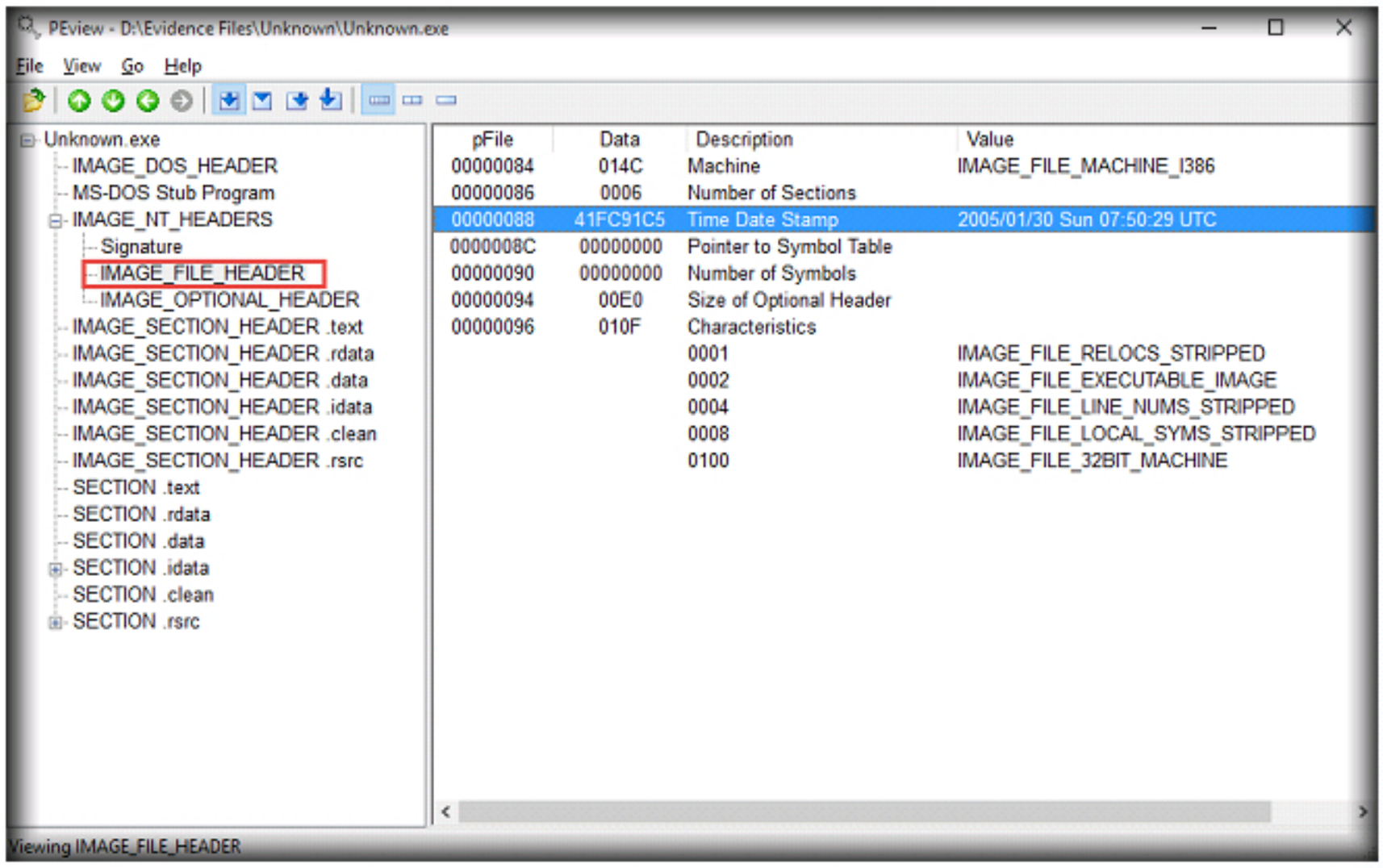


FIGURE 1.16: Viewing header data of the selected file in PEview tool

25. Click the **IMAGE_SECTION_HEADER.text** section. This section displays the virtual and raw size of the file. Virtual Size is the space allocated for a file while loading, whereas Raw Data, is the space it occupies on a disk.
26. The virtual and raw data sizes of the file are same if it does not contain any obfuscated code. By comparing these sizes, the investigator can detect the presence of hidden encrypted code.

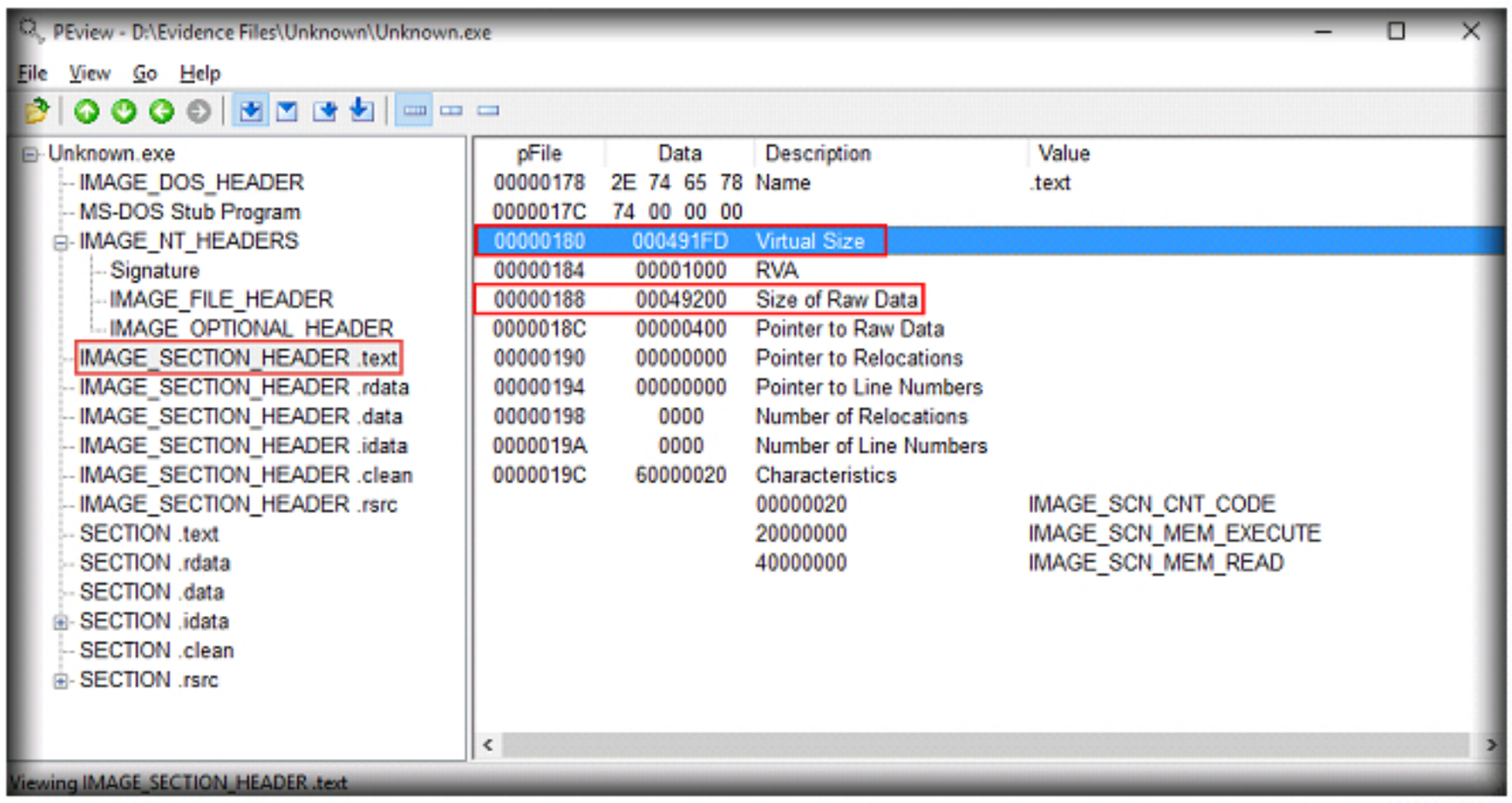


FIGURE 1.17: Comparing results in IMAGE_SECTION_HEADER.text section

27. Click the **IMAGE_OPTIONAL_HEADER** section from the left pane; it will display the state of the file, the size of initialized data section, the size of uninitialized data, the address of entry point, the subsystem required to run the file, etc.

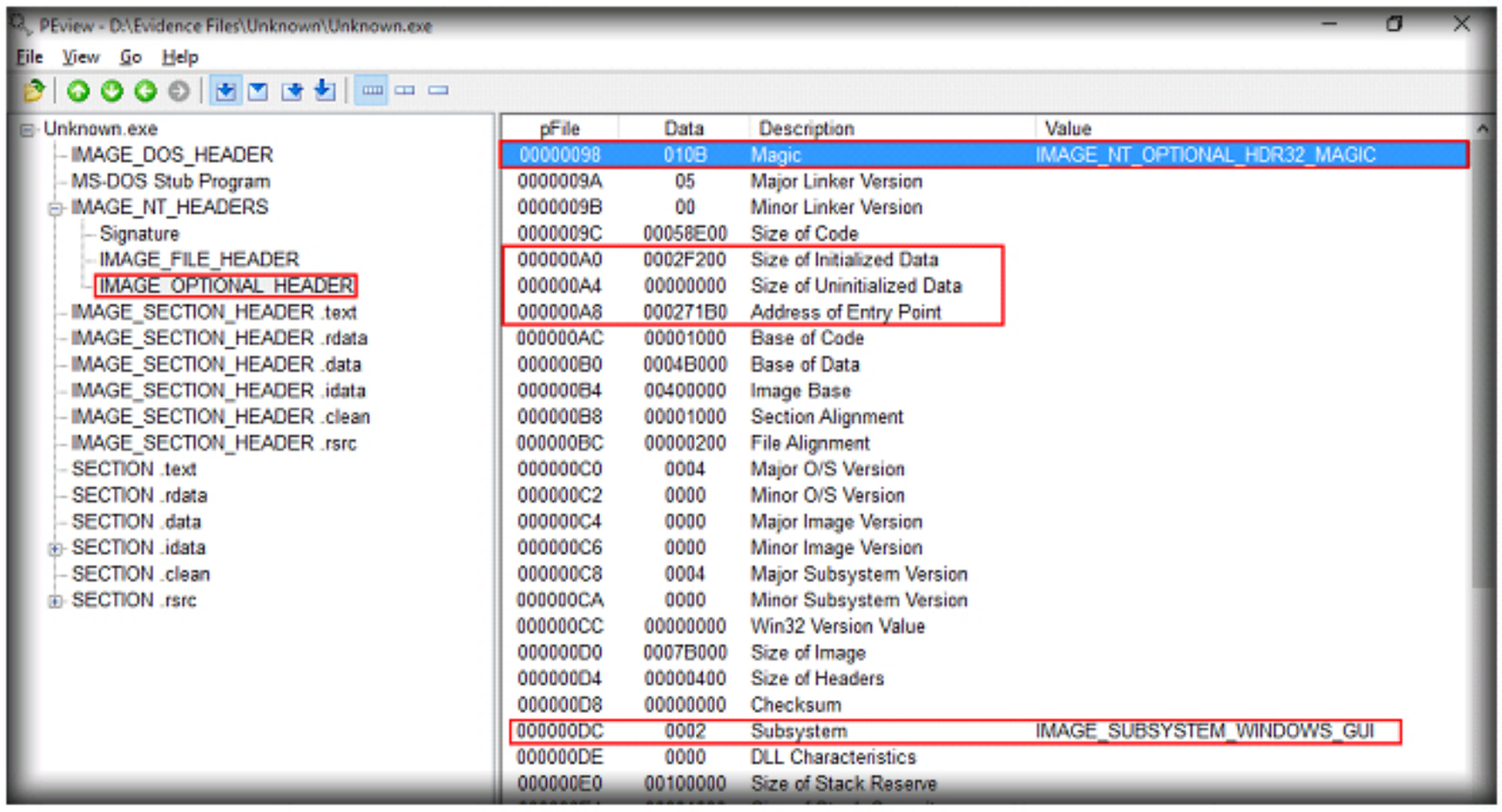


FIGURE 1.18: Viewing results in IMAGE_OPTIONAL_HEADER section

28. Other information it displays includes, the RVA and size of import table, export table, resource table, exception table, certificate table, base relocation table, debug directory, TLS table, etc. Information about these tables is important for understanding the impact of malware.

    For e.g., Malware may use the TLS callback function to evade debugging. If the malware has TLS callback function, then the TLS table would store some value.

pFile	Data	Description	Value
000000F4	00000010	Number of Data Directories	
000000F8	00000000	RVA	EXPORT Table
000000FC	00000000	Size	
00000100	00060000	RVA	IMPORT Table
00000104	00000104	Size	
00000108	00073000	RVA	RESOURCE Table
0000010C	000078D0	Size	
00000110	00000000	RVA	EXCEPTION Table
00000114	00000000	Size	
00000118	00000000	Offset	CERTIFICATE Table
0000011C	00000000	Size	
00000120	00000000	RVA	BASE RELOCATION Table
00000124	00000000	Size	
00000128	00000000	RVA	DEBUG Directory
0000012C	00000000	Size	
00000130	00000000	RVA	Architecture Specific Data
00000134	00000000	Size	
00000138	00000000	RVA	GLOBAL POINTER Register
0000013C	00000000	Size	
00000140	00000000	RVA	TLS Table
00000144	00000000	Size	
00000148	00000000	RVA	LOAD CONFIGURATION Table
0000014C	00000000	Size	

FIGURE 1.19: Viewing results in IMAGE_OPTIONAL_HEADER section

29. Click on the **SECTION .idata** section to see a list of its contents, and select **IMPORT Directory Table** from the list.
30. This table stores information about the DLL files the file calls during execution. Scan these files to check for any suspicious content.

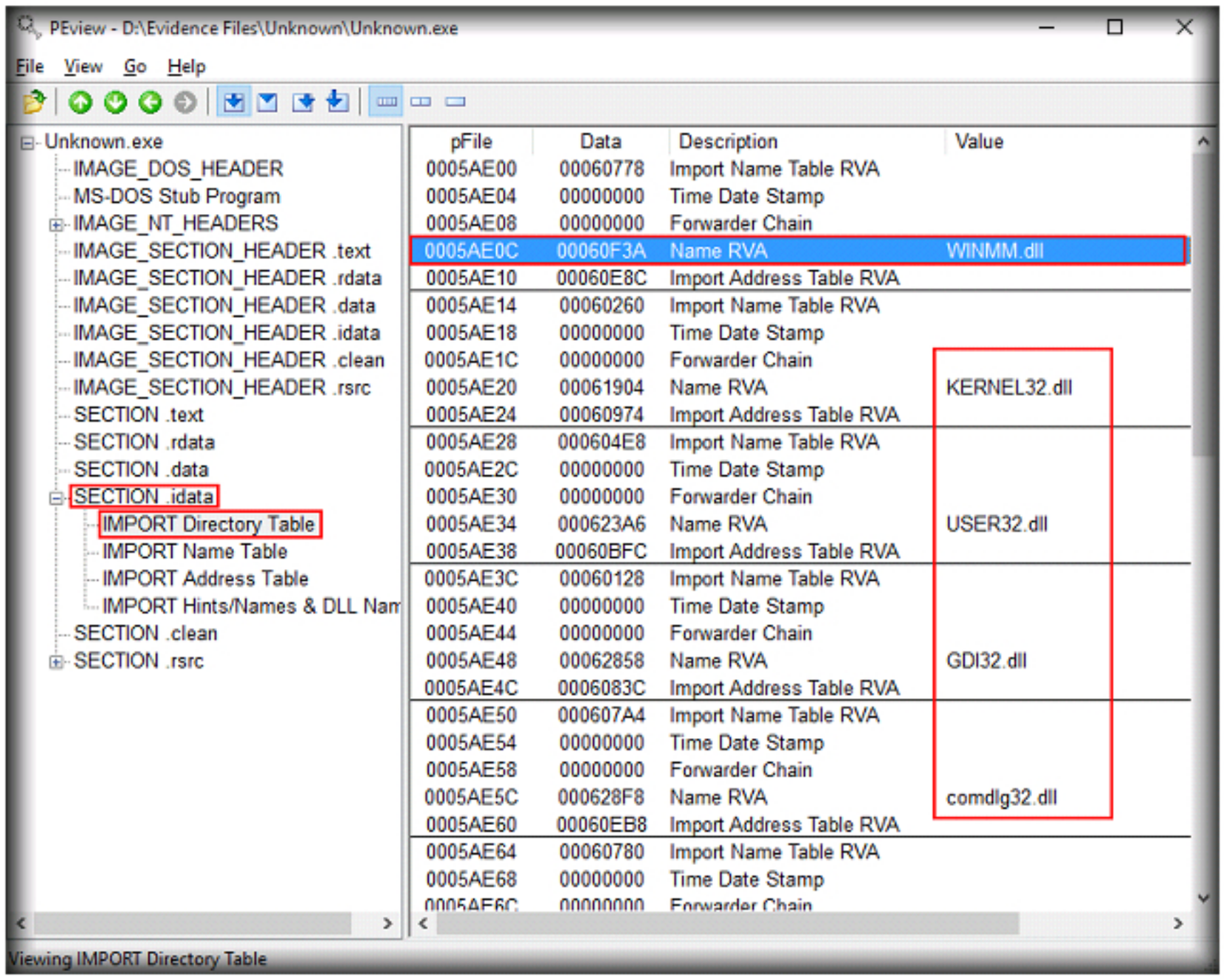


FIGURE 1.20: Viewing results in SECTION .idata section

31. Analyze each section of the file thoroughly and mark the suspicious content.

TASK 4

Finding Suspicious Strings

32. The next step is to find the suspicious strings from the target file using the **Strings** tool.
33. Right click on the **Start** or **Windows menu icon** and select **Command Prompt**.
34. Command Prompt appears as shown in the screenshot:

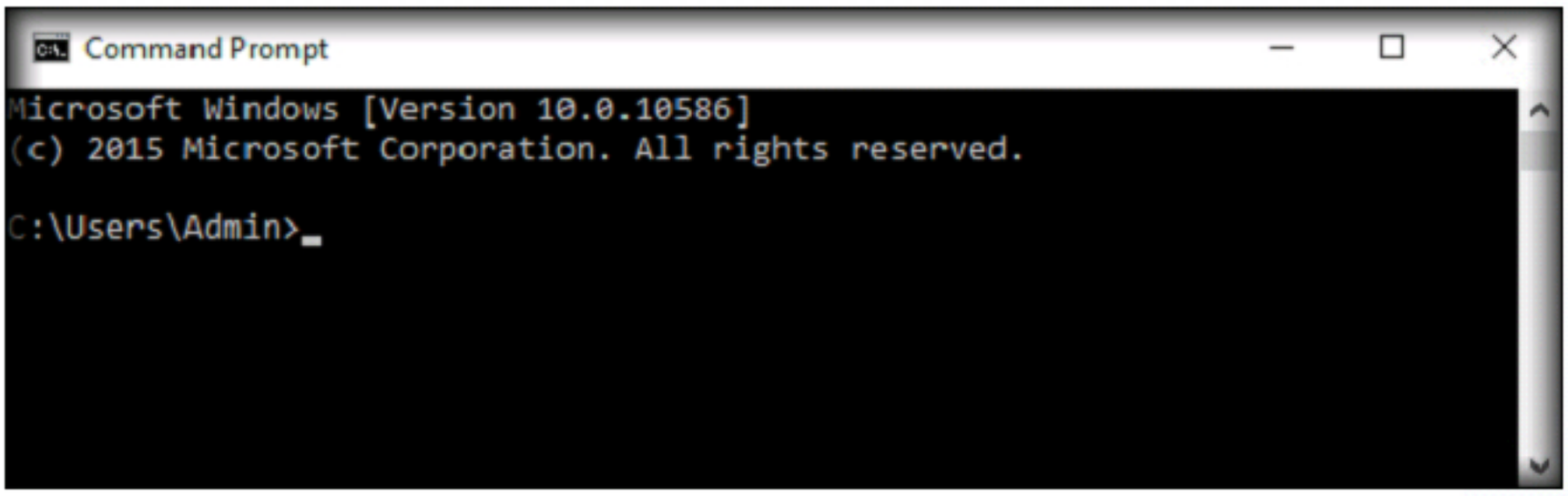


FIGURE 1.21: Command Prompt

35. Navigate to **Z:\CHFIv9 Module 11 Malware Forensics\Strings Search Tools\Strings**, click the **strings.exe** file, drag and drop it onto the command prompt console.
36. The syntax for extracting all the strings of a file named **Program.exe** is **Location:\>Filepath\strings.exe Filepath\Program.exe**.
37. Navigate to the location, **Z:\Evidence Files\Unknown**, select the **Unknown.exe** file, drag and drop it onto the command prompt console.

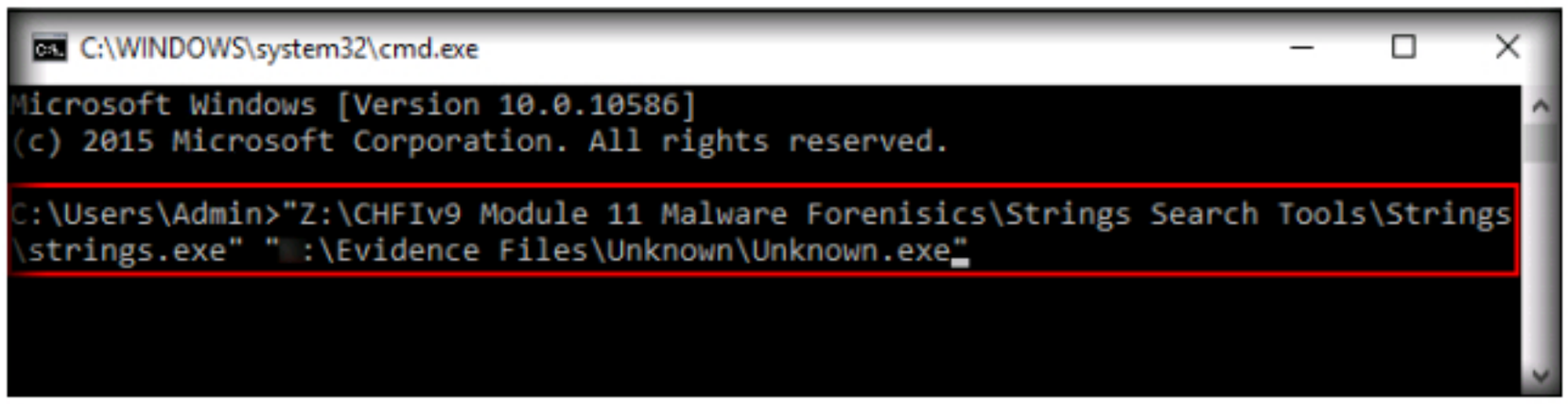


FIGURE 1.22: Dragging and dropping Unknown.exe file in command prompt

38. Press enter to run the tool. The tool will search the target file and display all the possible strings from it.

Note: The tool will display the **Strings License Agreement** dialog box, click the **Agree** button.

39. The command prompt will display a list of strings embedded into the target file.

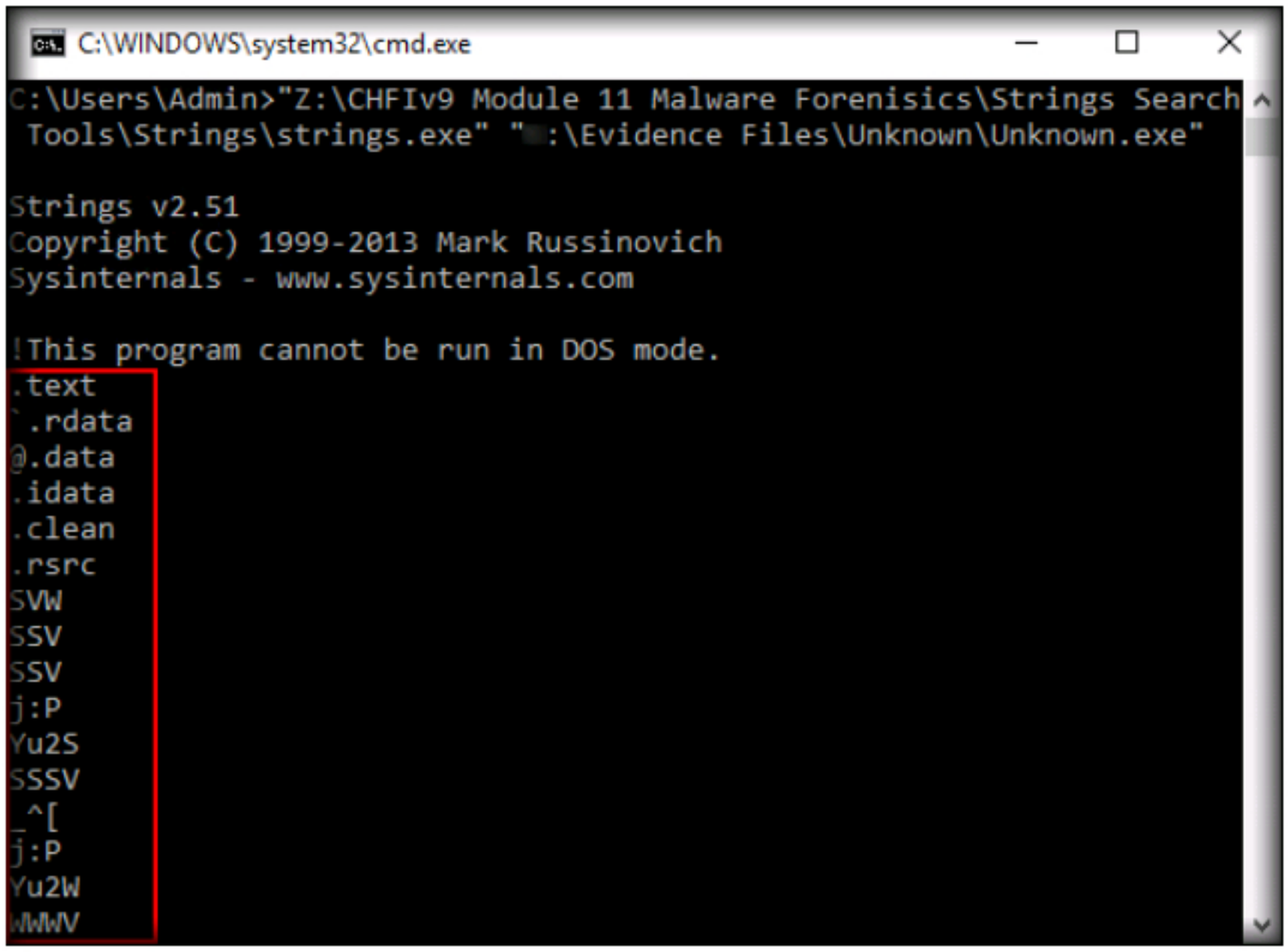


FIGURE 1.23: Command prompt displaying strings embedded into target file

40. Investigators need to go through these strings to find the harmful or malicious functions.
41. Analysis of strings in the suspicious file reveals that the file contains shell programming that tries to gather system information, an identifier of the device control, etc.
42. For better understanding and readability investigators can also use the tool to copy the strings list to a notepad file.
43. Syntax for copying the strings list to the strings1.txt notepad file: **C:\>Filepath\strings.exe Filepath\Program.exe >Filepath\strings1.txt**.

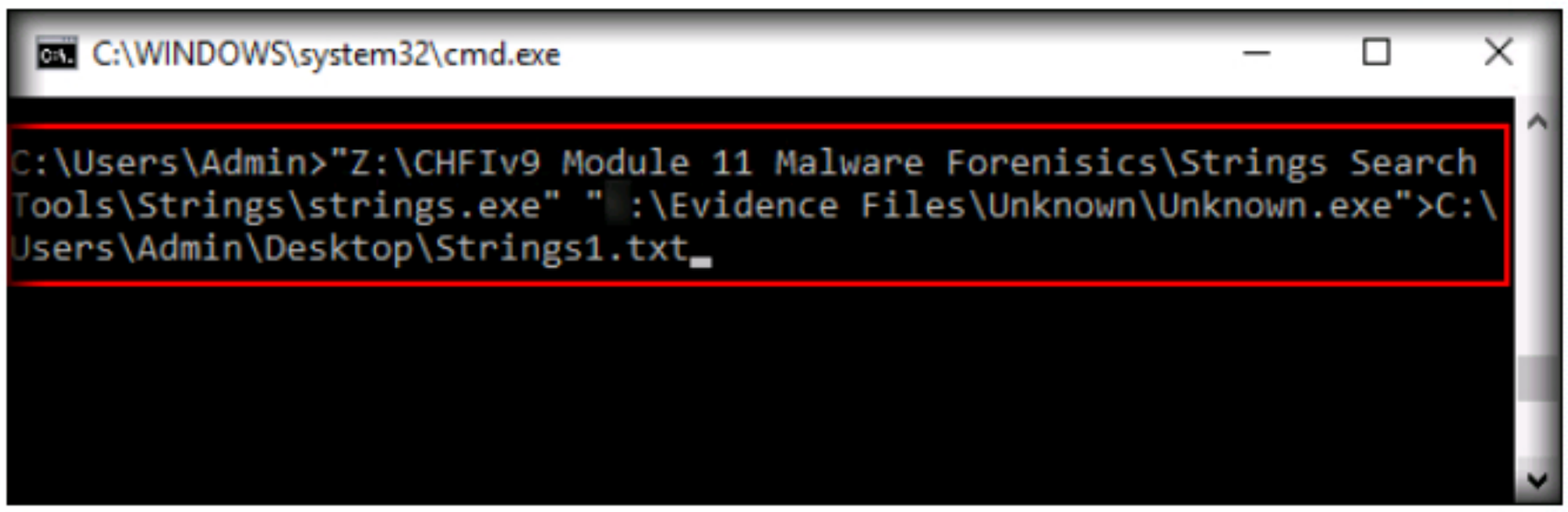


FIGURE 1.24: Syntax for copying strings list

44. The command line interface will copy the result to the mentioned text file, which can be used for future analysis.

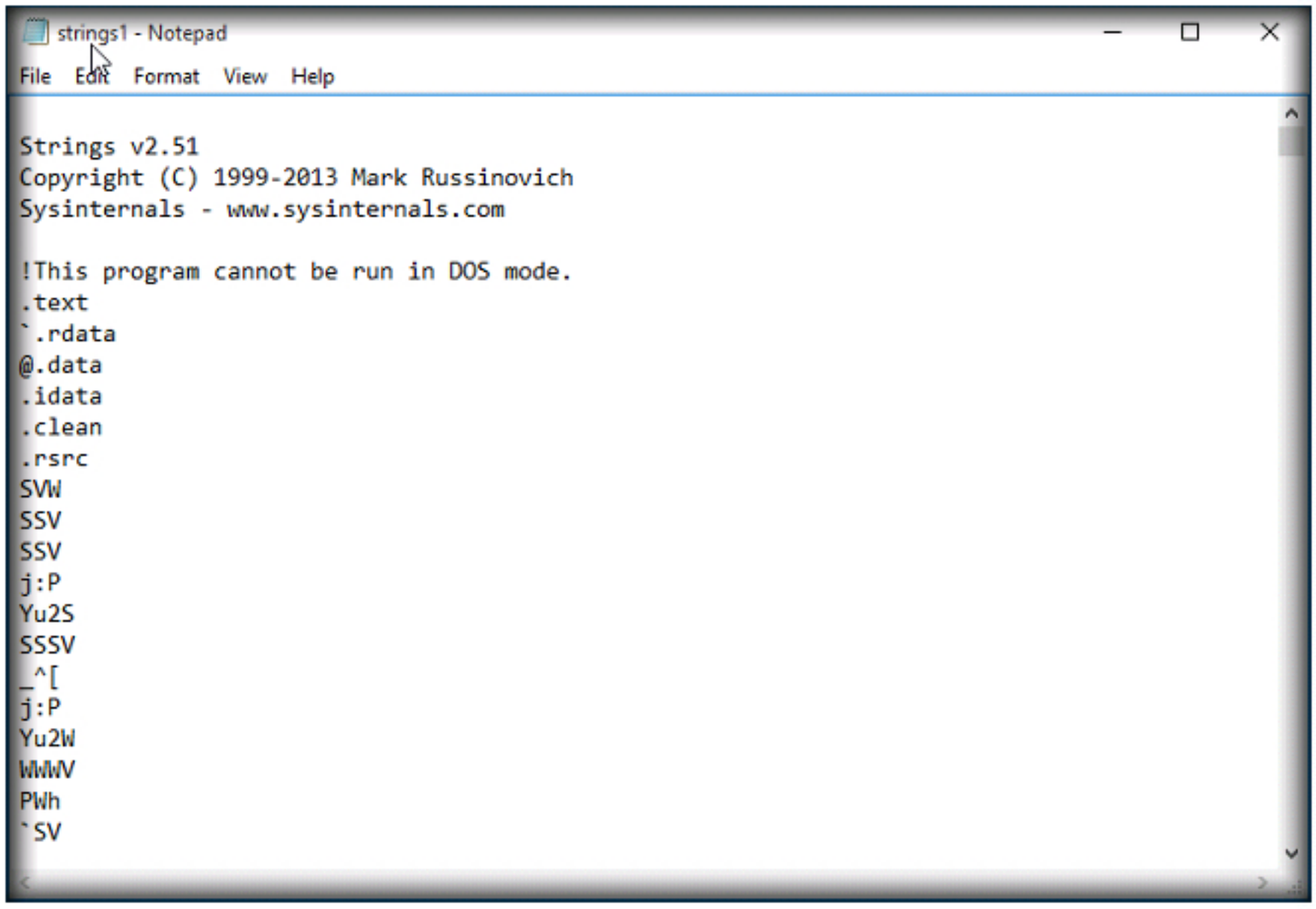


FIGURE 1.25: Examining the strings list in a notepad file

45. The strings define the function of a program such as copy a file, enter a password, etc. The investigator needs to find the specific strings that could result in malicious activities. Examples of malicious strings include **GetLastActivePopup** and **GetActiveWindow**, which provide the attacker information about the last active popup and window that were running on the system.

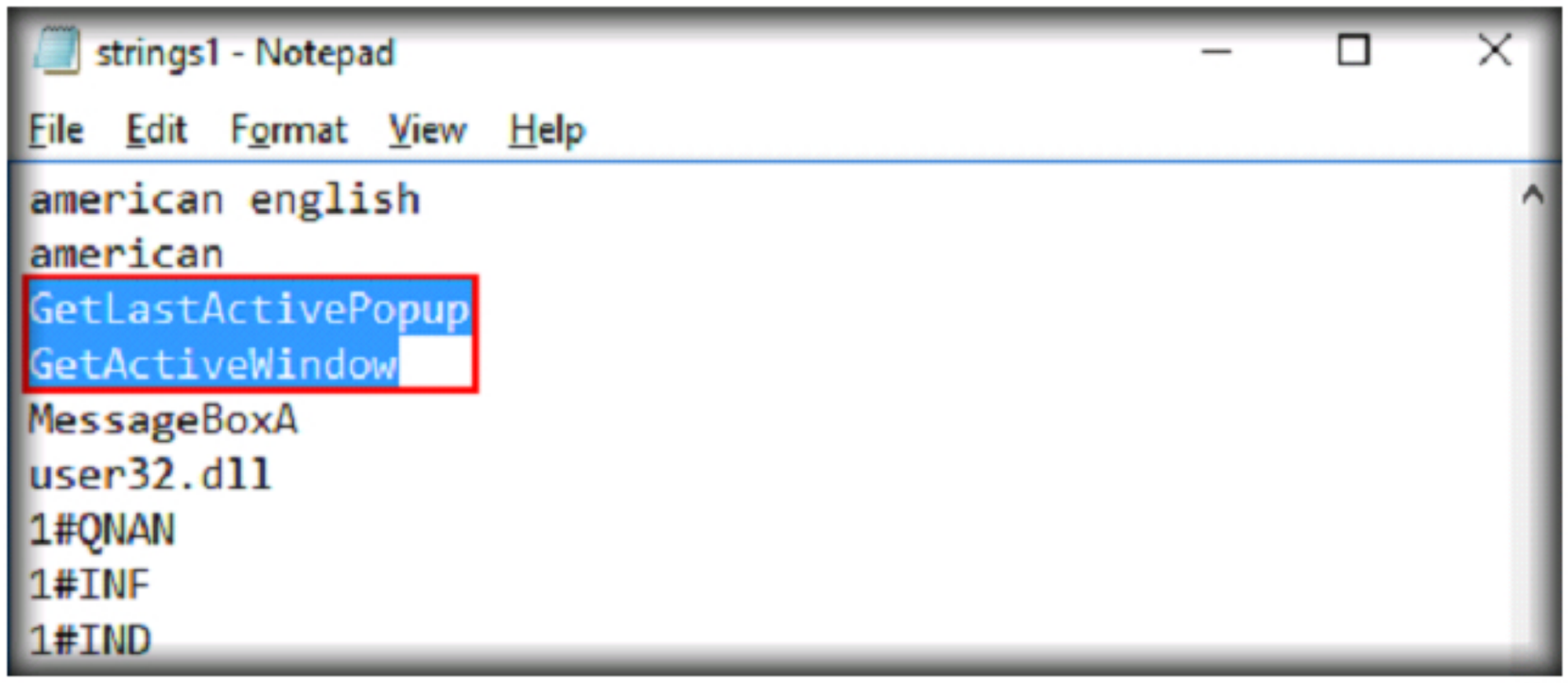


FIGURE 1.26: Detecting the malicious strings

TASK 5

Finding File Dependencies

46. After finding the strings, the investigator needs to check the processes and dependencies of the file using the Dependency Walker tool.

47. To run the dependency walker tool, navigate to the location **Z:\CHFIv9 Module 11 Malware Forensics\PE Data Acquisition Tools\Dependency Walker**, and double click the **depends.exe** file.

48. Click the **Open** file icon from the taskbar, navigate to the location, **Z:\Evidence Files\Unknown**, select the **Unknown.exe** file and click the **Open** button.

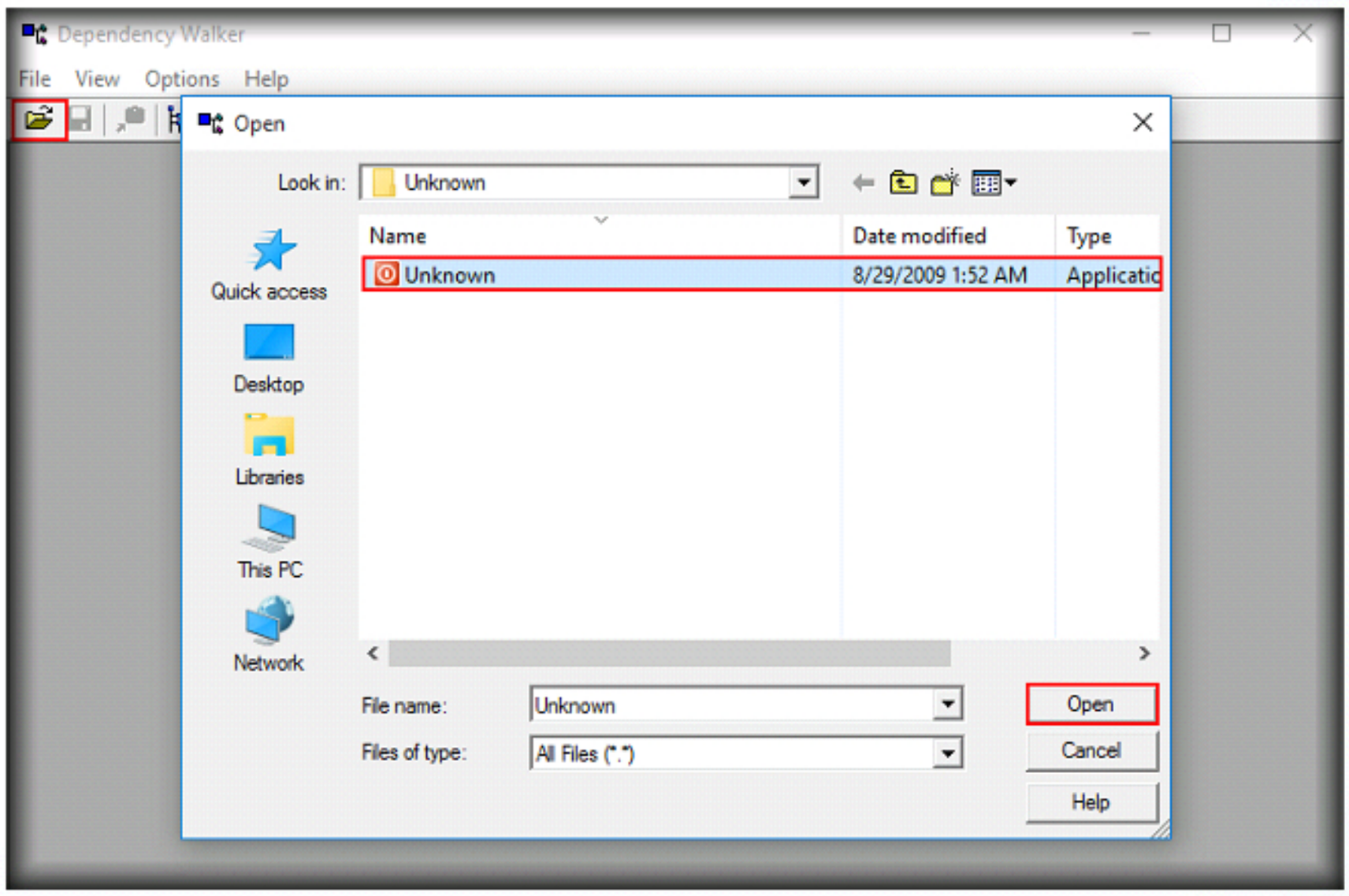


FIGURE 1.27: Navigating to the evidence file in Dependency Walker tool

49. The tool takes some time to examine the file and then, displays the list of DLLs linked to the file.

Note: If a **Dependency Walker** pop-up appears displaying an error, click **OK**.

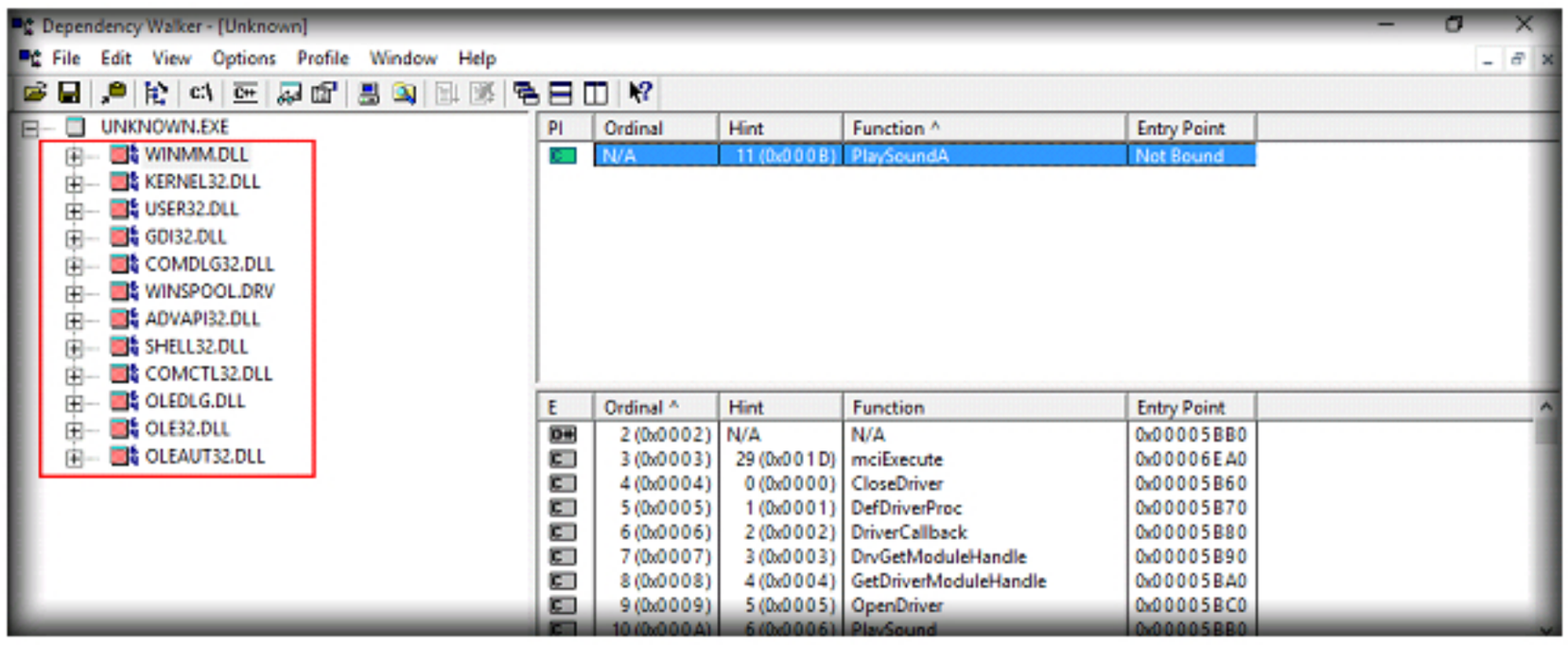


FIGURE 1.28: Examining the list of DLLs of the target file

50. Click on each dll name to find its functions and other linked resource libraries. Look for the malicious strings that prompt functions to copy system data, events or other private information.

51. In the image given below, the selected (memcpy) is a function that prompts the system to copy certain bytes to a destination location.

    From this, an investigator can derive that the malware is a Trojan and would tend to copy user information.

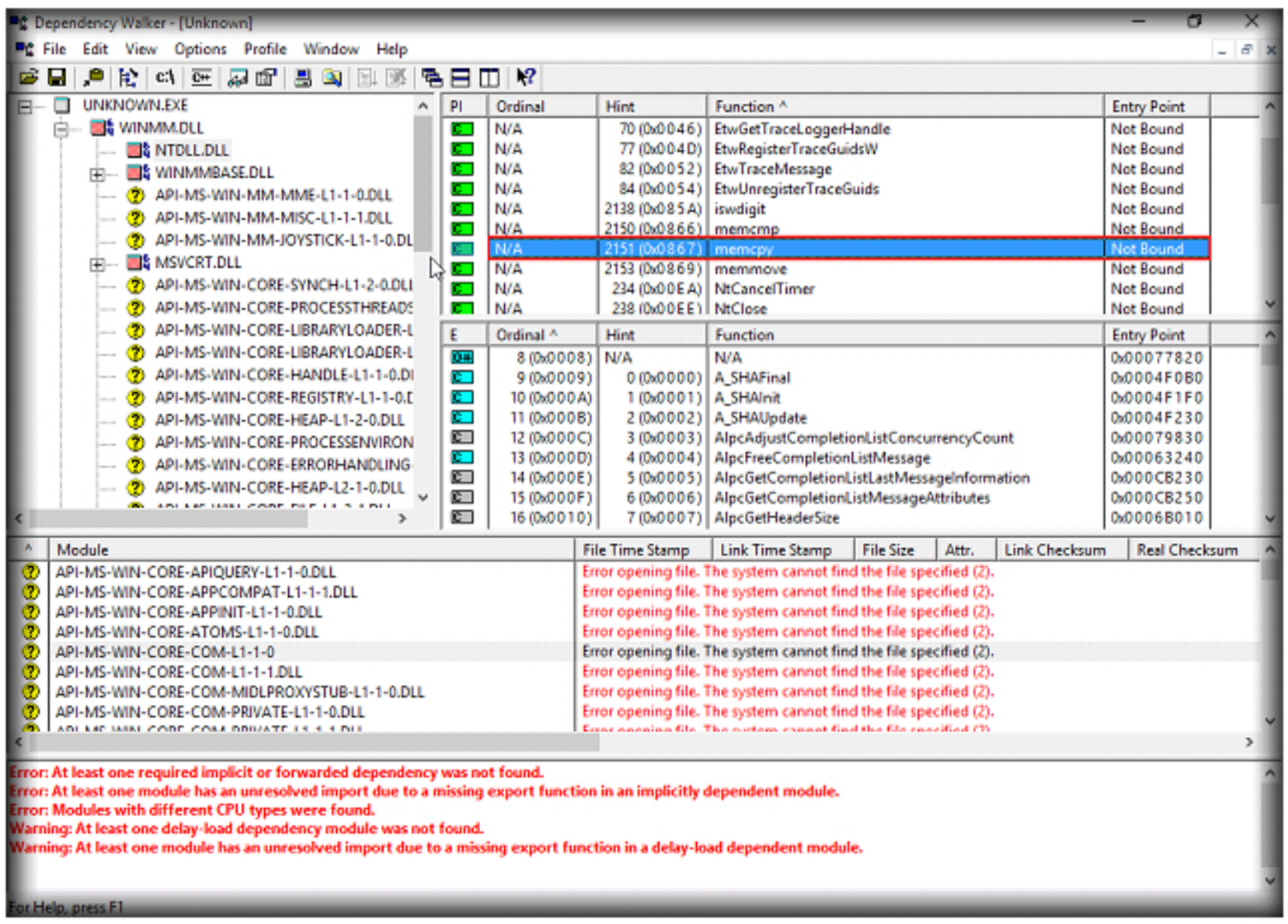


FIGURE 1.29: Examining a specific function in the Dependency Walker tool

52. Go through the libraries of each DLL to find its functions. The **WINMM.DLL** has functions that could trace enabled flags, enabled levels, messages, track guides, etc.

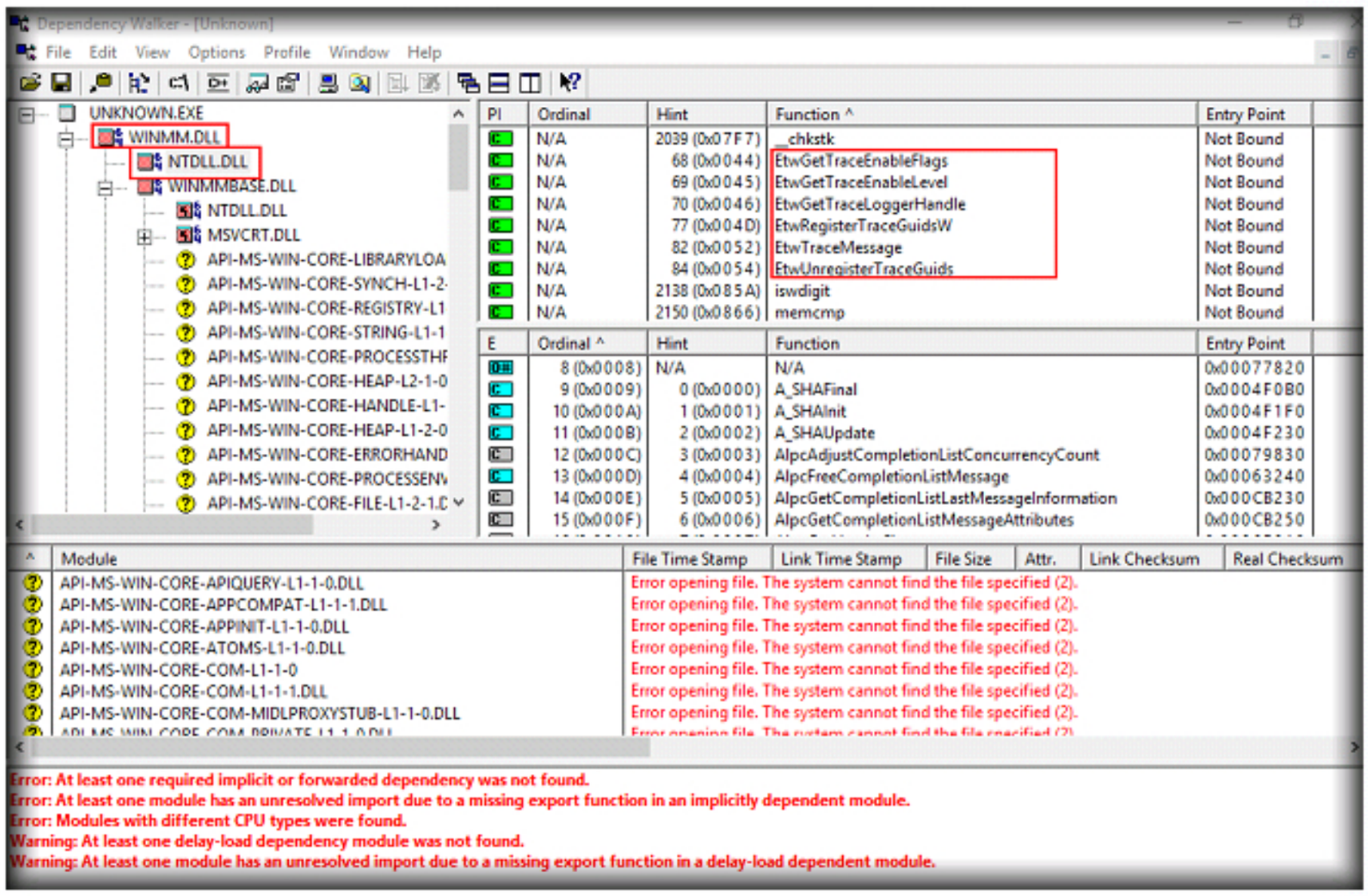


FIGURE 1.30: Viewing functions in WINMM.DLL library

53. These functions represent that the linked libraries would enable the system to trace the flags, messages and other details, and prompt them to the attacker.

54. On finding the file dependencies, exit the application.

TASK 6

Dissemble the file with IDA Pro

55. Investigators can prove that a malware is harmful, only when they can find the functions such as import, copy, connect to a URL and export embedded in their code.

56. IDA Pro is an executable disassembler that helps investigators in gathering information about executables and malware files

57. To install the IDA Pro tool, navigate to **Z:\CHFIv9 Module 11 Malware Forensics\Malware Analysis Tools\IDA Pro** location and double-click the **idafree50.exe**.

58. Follow the installation instructions and after its completion, launch the application from the desktop.

59. Upon launching the application for the first time, an About pop-up appears, click **OK**.

60. Later, an **IDA Pro License** window appears; accept the license agreement by clicking **I Agree**.

61. If a **Help** Guide appears, **Close** it.

62. Click the **New** button to add a new file for analysis.

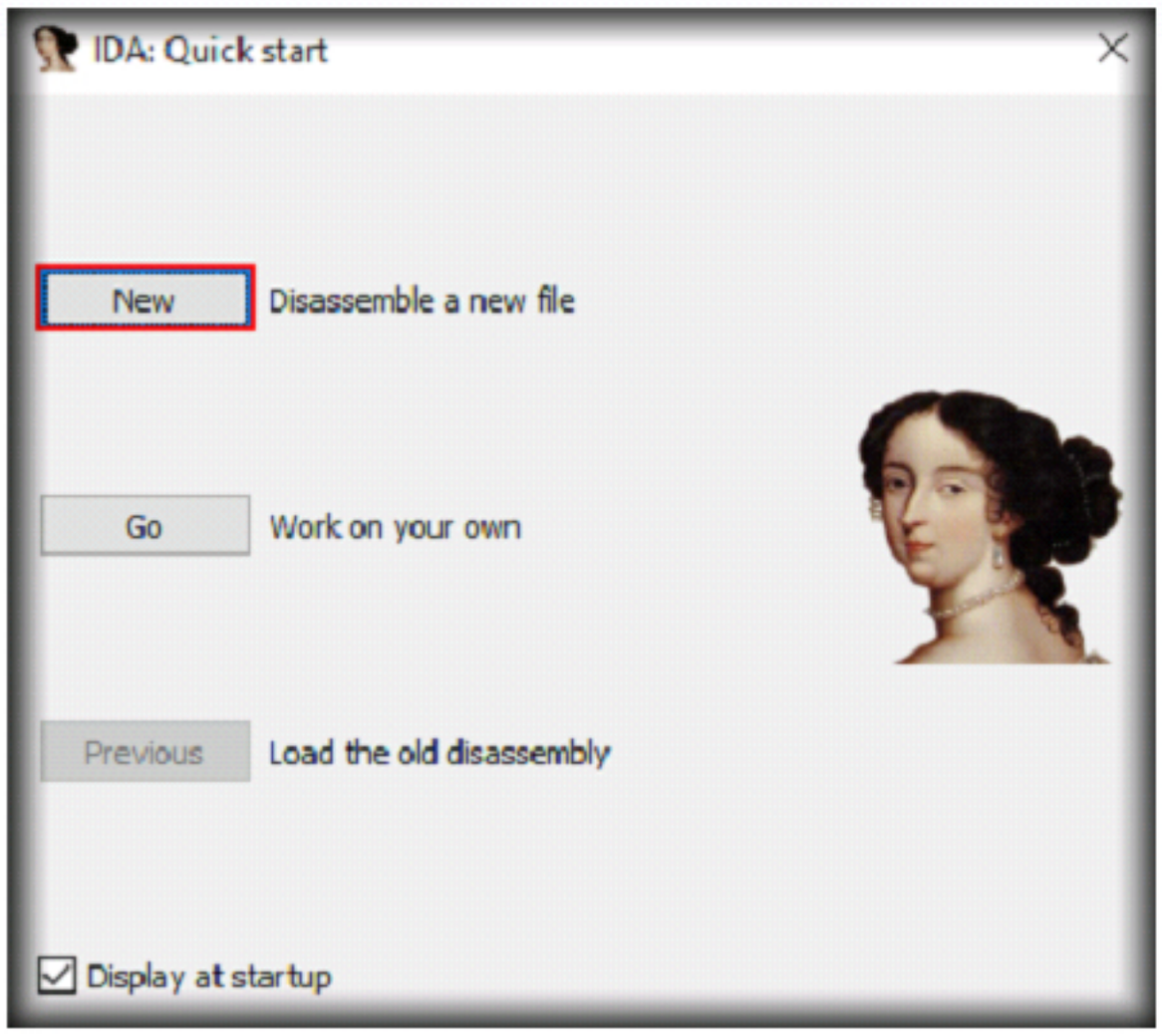


FIGURE 1.31: IDA Pro main window

63. The tool will prompt a dialog box to **Select file to disassemble**. Select **PE Executable** and click **OK**.

64. Navigate to the location **Z:\Evidence Files\Unknown**, select the **Unknown.exe** file and click **Open**.

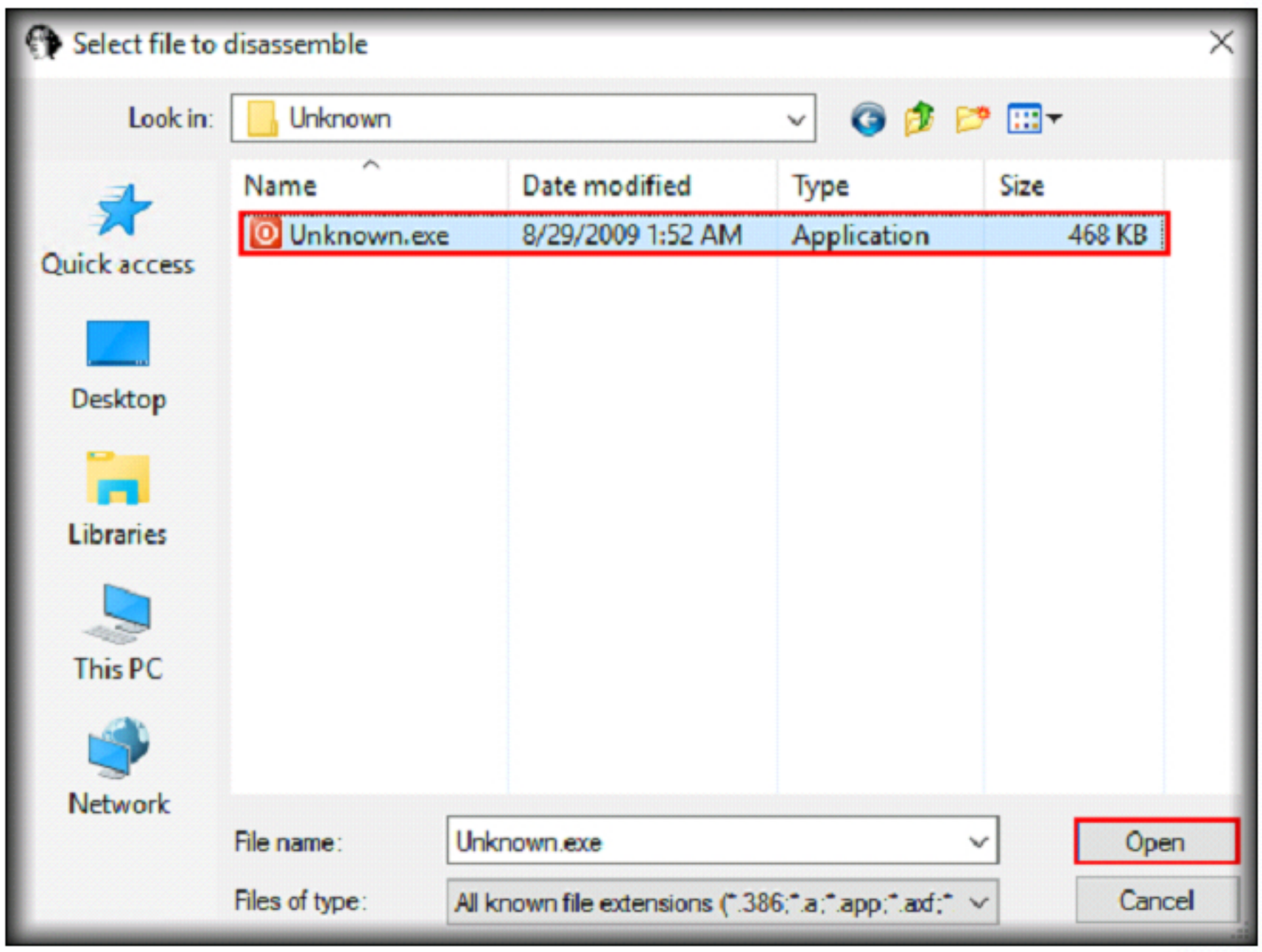


FIGURE 1.32: Navigating to the target file in IDA Pro

65. **PE Executable file loading Wizard** appears, follow the wizard driven steps to load the file.

66. Once the tool loads the file, it displays the internals of the file such as IDA view, hex view, enumerations, imports and exports.

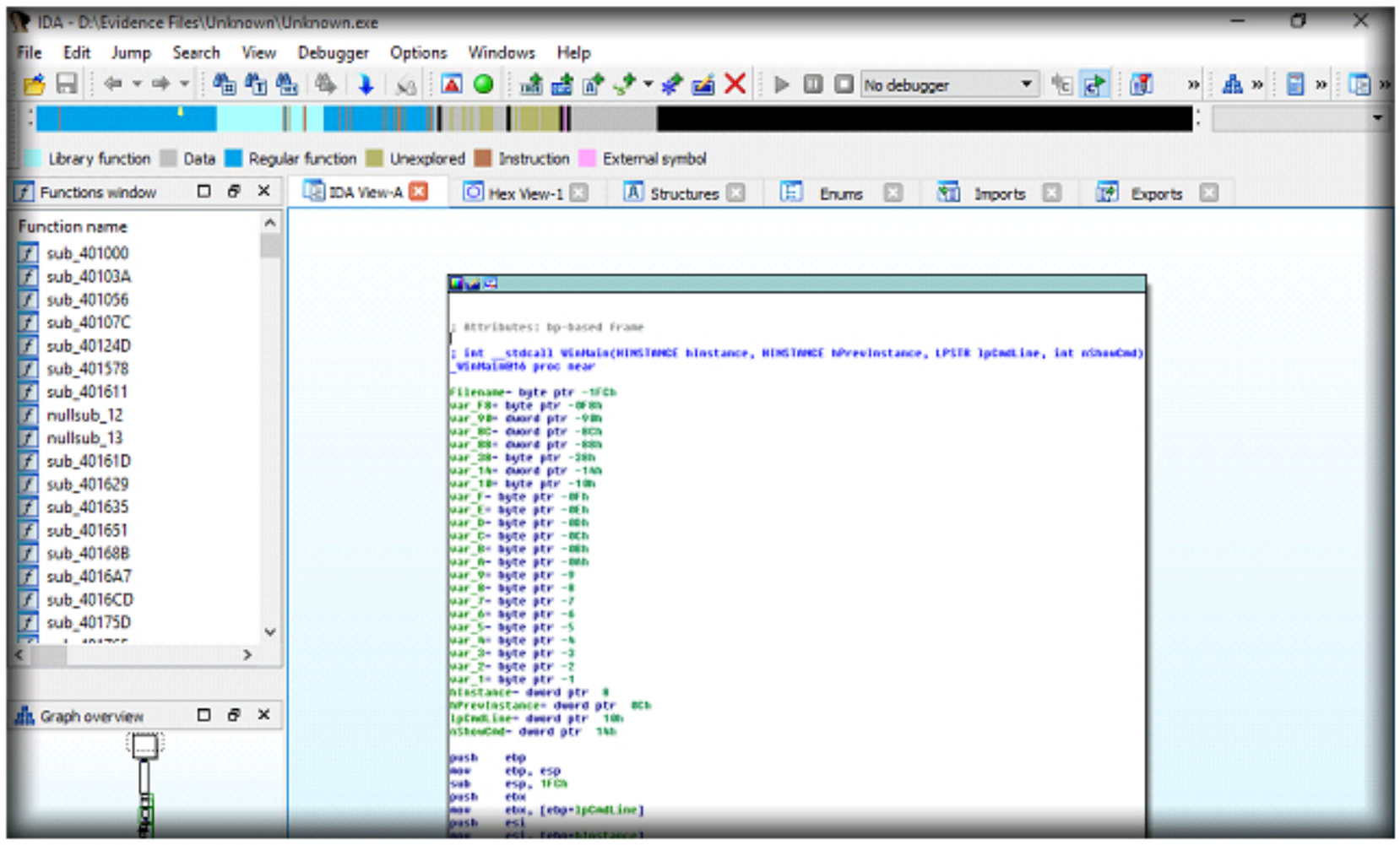


FIGURE 1.33: Examining the internals of the target file

67. Click on the **IDA View-A** tab to view paths of subroutines in the file. The arrows depict the functions calling a subroutine. Each box represents a function or subroutine.

68. To read the text in a subroutine, the investigator needs to right-click in the IDA View-A window and select **Text view** from the drop down list.

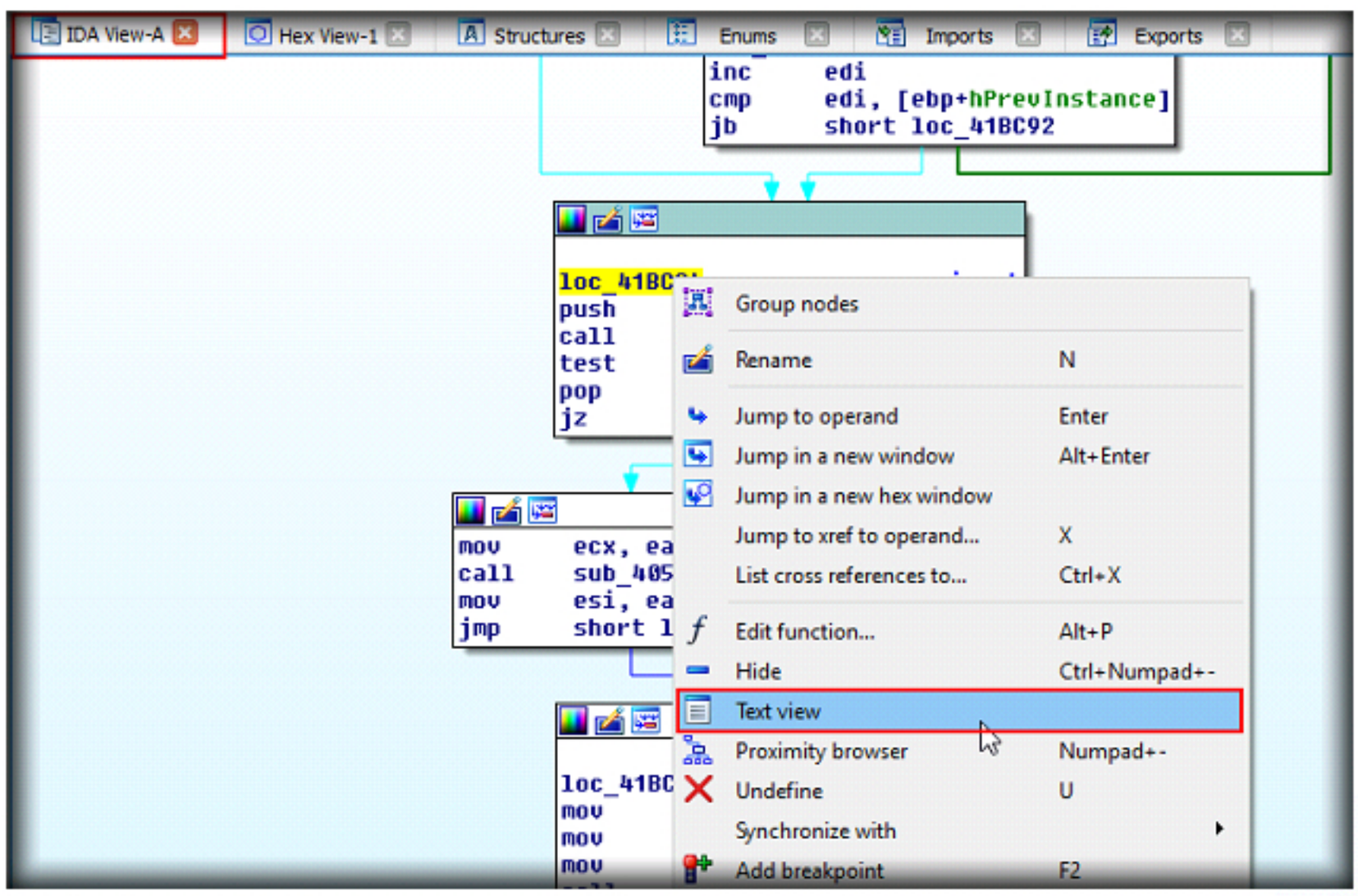


FIGURE 1.34: Reading text in a subroutine

69. Investigators can verify the functions that call a subroutine navigating through them using the **Graph** overview window.

70. The highlighted area represents the text version of a subroutine. The arrows represent the other subroutines that called this subroutine.

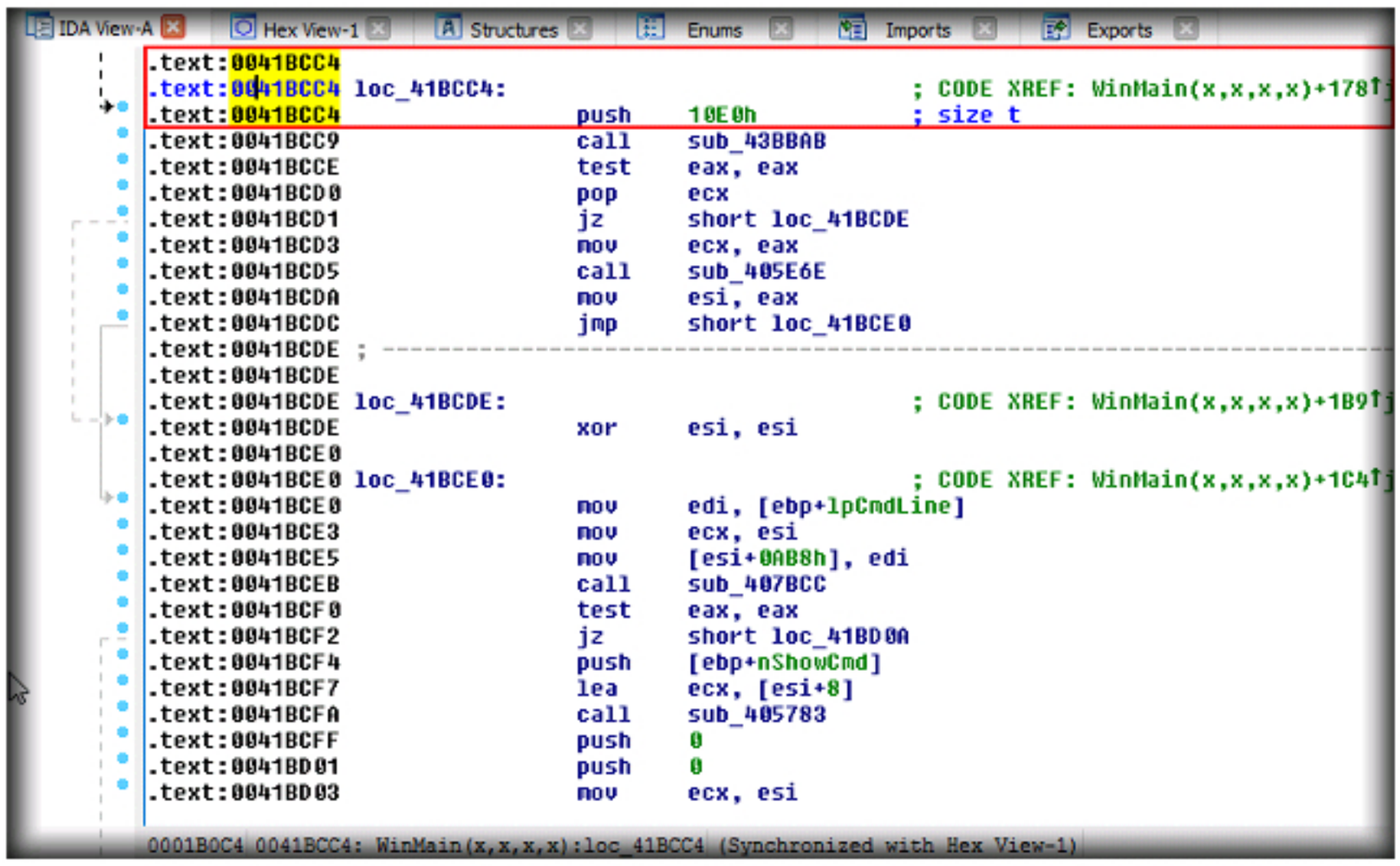


FIGURE 1.35: Examining text in a subroutine

71. Click the **Imports** tab to see all the imports of the file.

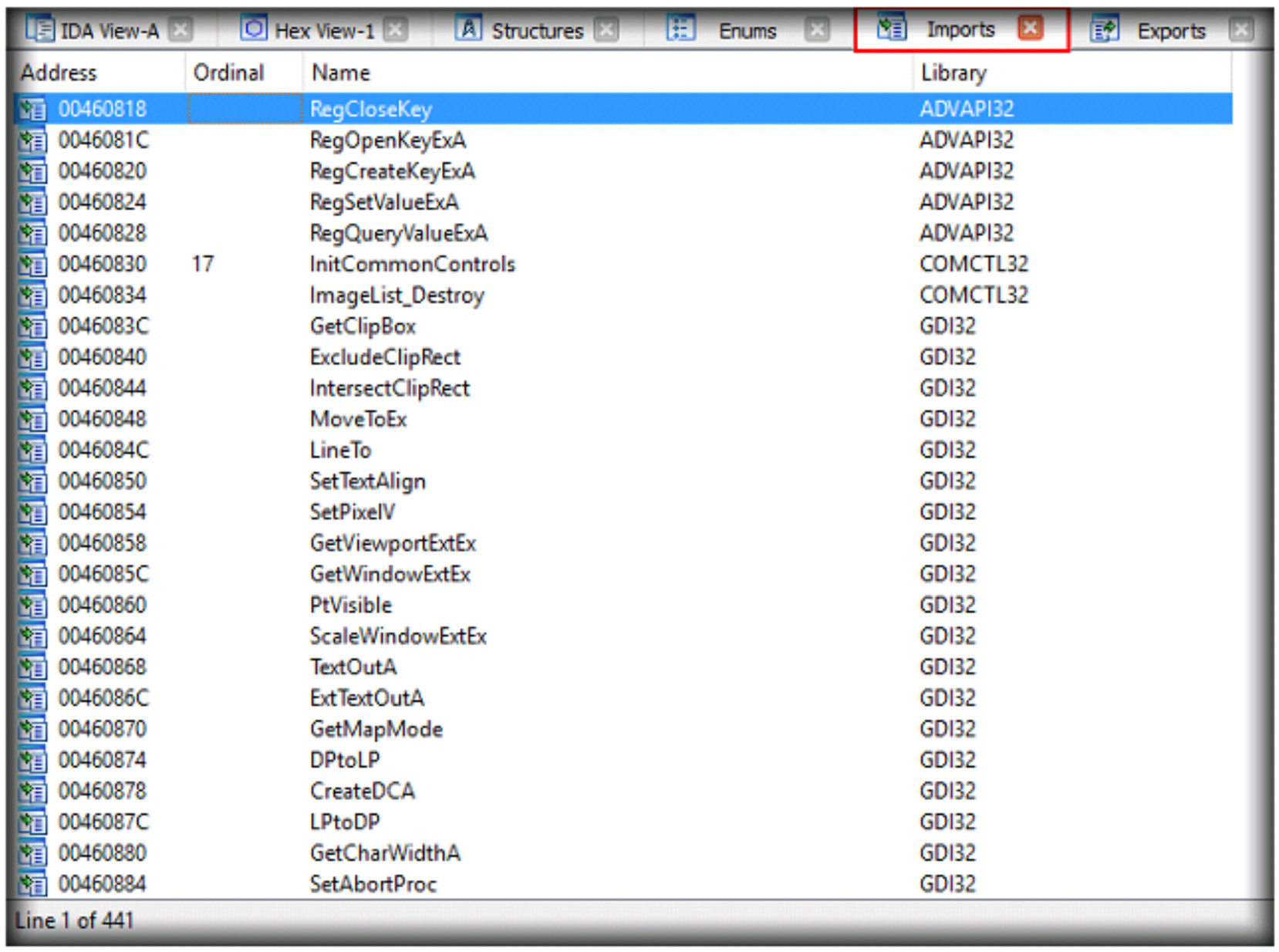


FIGURE 1.36: Viewing all the imports of the file from the Imports tab

72. To filter for any known imports, right click the **Imports** window and select **Quick filter**.

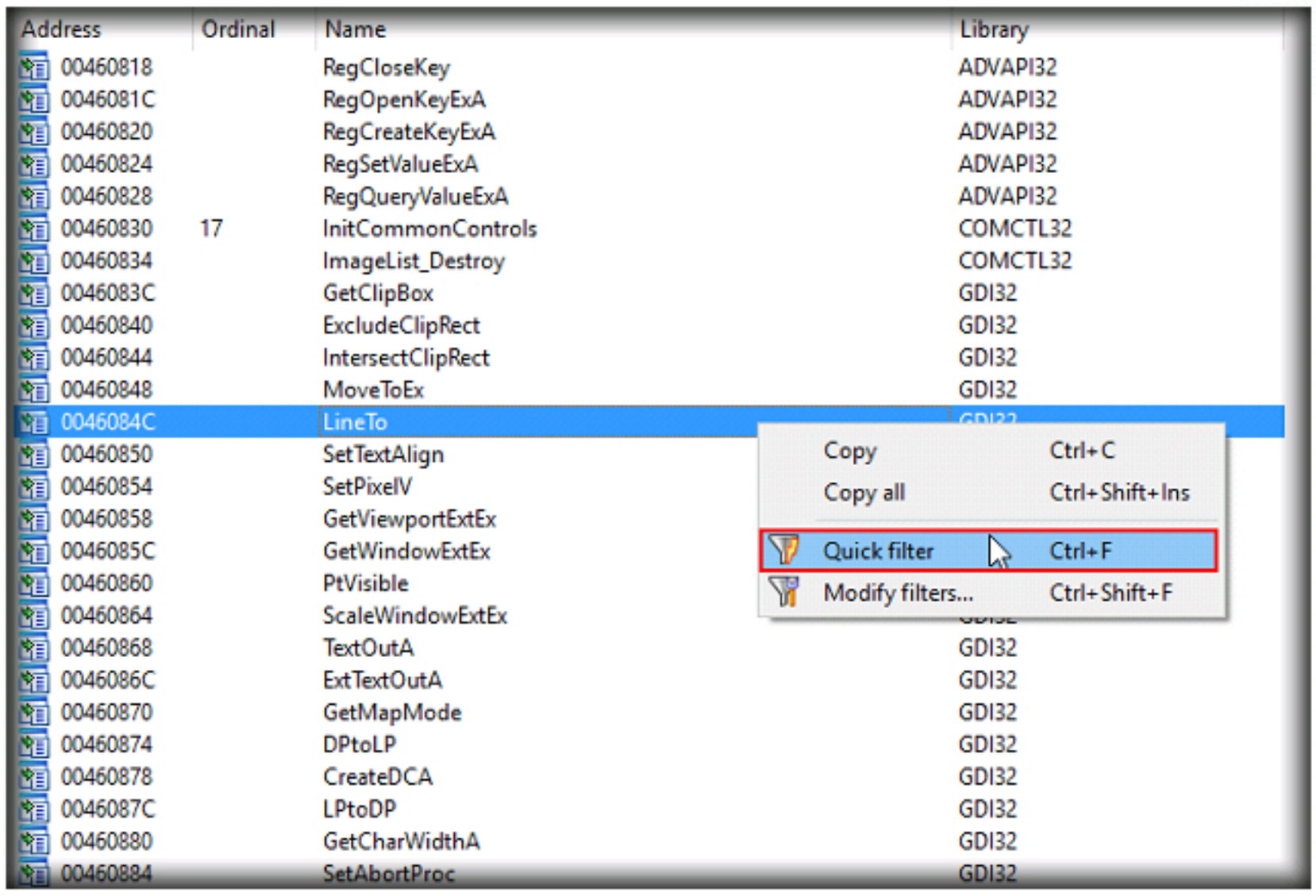


FIGURE 1.37: Filtering known imports

73. The tool will display a search bar under the window, type the names of imports to be searched.

When queried for the local keyword, the tool will display all the functions having the word 'local' in their name. Further analysis reveals that the file tries to gather data such as the system's local time, threads, free space, and unlock the local files as well as free the space.

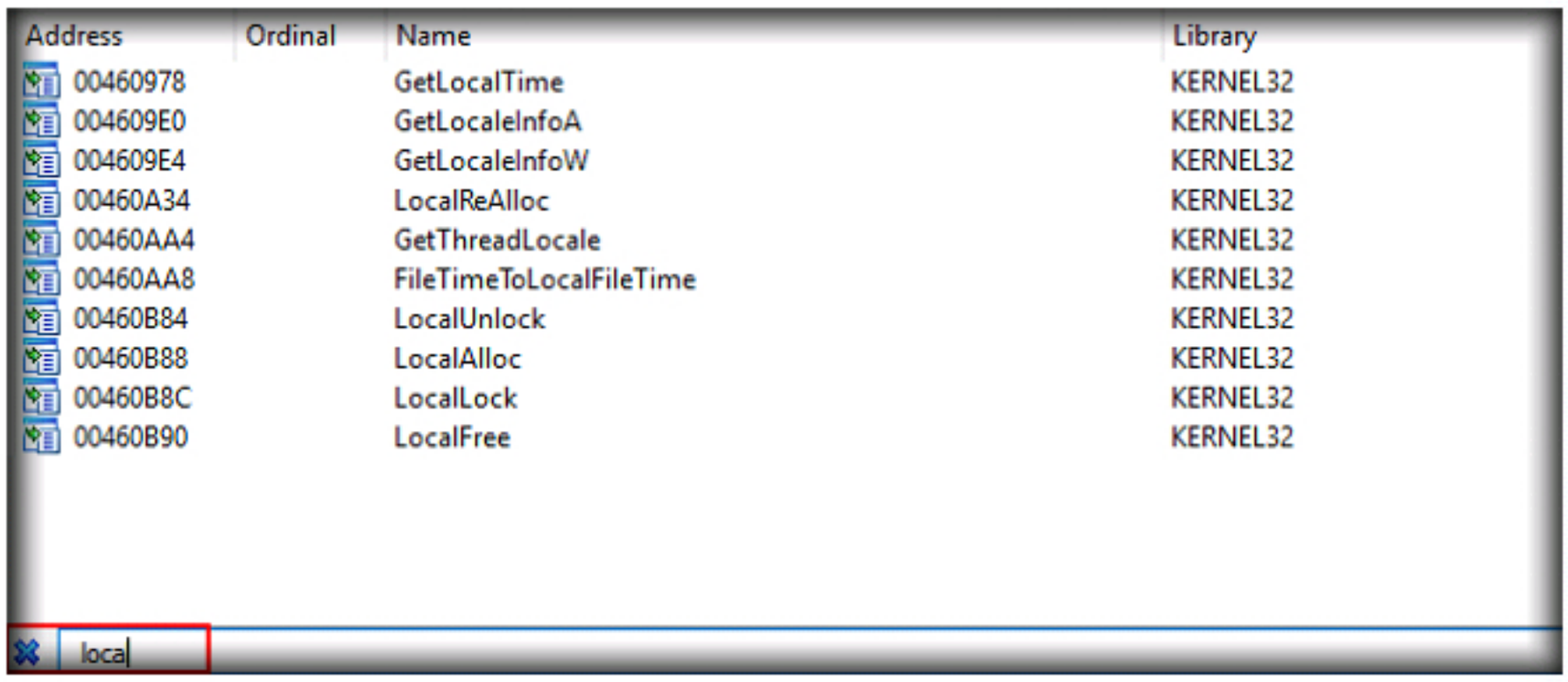


FIGURE 1.38: Querying using the search bar

74. After finding a suspicious import, double-click on its address to observe its IDA view-A.

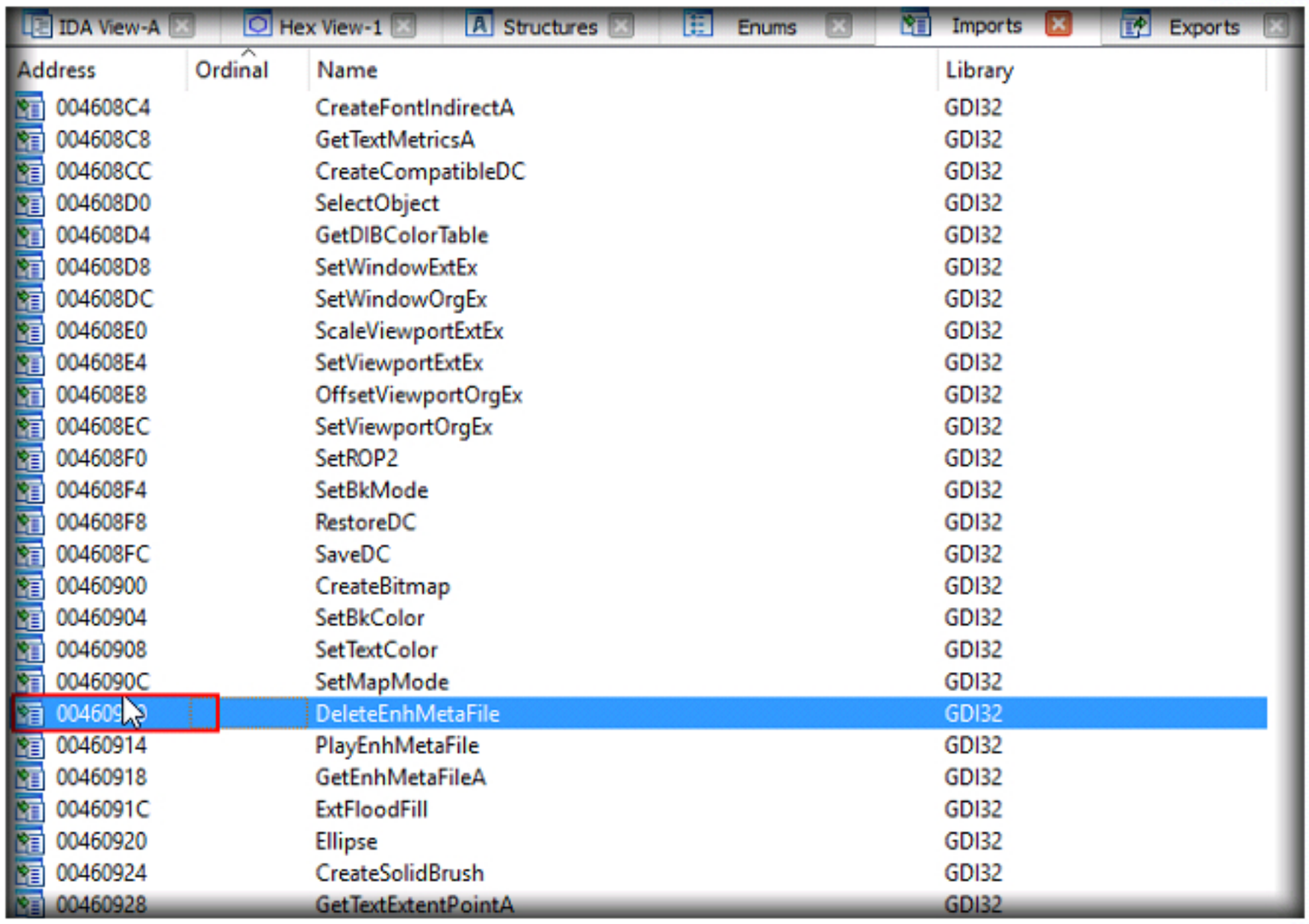


FIGURE 1.39: Observing IDA view-A of the suspicious import

75. The IDA view will display the coding and other subroutines a function includes as well as displays the coding in these subroutines when users hover the cursor over them.

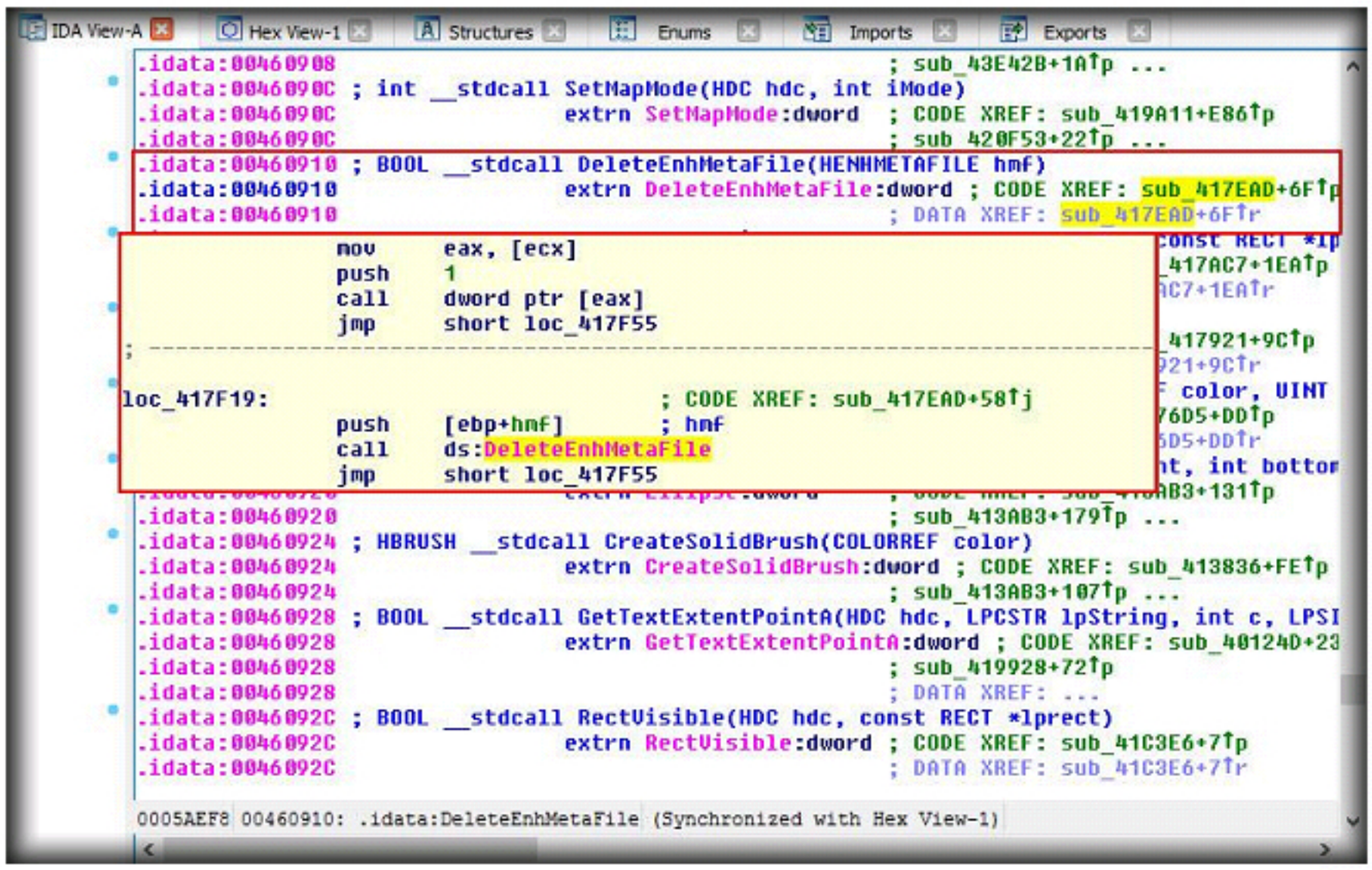


FIGURE 1.40: Examining the coding and other subroutines of the suspicious import

76. Analyze all suspicious subroutines and find their links to determine their complete objective.

Lab Analysis

Document all the suspicious data such as strings, DLLs, functions and packages discovered during the lab.

YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 2

Dynamic Malware Analysis

Dynamic malware analysis is the process of assessing the impact of a malware by running it in a closed environment.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

This lab explains the process of performing dynamic analysis of a malicious file to find the processes it starts, network operations, file changes and other activities. This process will help in finding out the sections of memory the file would influence, IP addresses it would try to connect and functions it would perform on a system.

Lab Objectives

The objective of this lab is to help the student to understand how to perform dynamic malware analysis.

Lab Environment

To carry out this lab, you need:

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv9 Module 11 Malware Forensics

- A computer running **Windows 10** virtual machine.
- ApateDNS located at **Z:\CHFIv9 Module 11 Malware Forensics\Network and Internet Simulation Tools\apateDNS**.
- Netcat (nc.exe) located at **Z:\CHFIv9 Module 11 Malware Forensics\Network and Internet Simulation Tools\nc111nt**.
- Wireshark located at **Z:\CHFIv9 Module 11 Malware Forensics\Network Activity Monitoring Tools\Wireshark**.
- Regshot located at **Z:\CHFIv9 Module 11 Malware Forensics\Registry Entry Monitoring Tools\Regshot**.
- Process Monitor located at **Z:\CHFIv9 Module 11 Malware Forensics\Dynamic Malware Analysis Tools\Process Monitor**.
- Process Explorer located at **Z:\CHFIv9 Module 11 Malware Forensics\Dynamic Malware Analysis Tools\Process Explorer**.

- Administrative privileges to run the tools.

Lab Duration

Time: 20 Minutes

Overview of the Lab

Dynamic malware analysis refers to the process of studying the behavior of the malware by running it in a monitored environment. The environment design should include the tools that can capture every movement of the malware in detail and give feedback to the investigator. Mostly virtual systems act as a base for conducting such experiments.

Lab Tasks

TASK 1

Dynamic Analysis

1. Before starting this lab, install the tools and isolate the virtual machine from any network to prevent malware propagation.
2. Install the ApateDNS tool from **Z:\CHFIv9 Module 11 Malware Forensics\Network and Internet Simulation Tools\apateDNS**, by double-clicking **apateDNS.exe** and following the wizard driven installation steps.
3. After installation, launch the application and enter the IP address for **DNS Reply IP (Default: Current Gatway/DNS)** to complete the setup and click the **Start Server**.
4. The ApateDNS will receive all the requests from the system and will reply to them from the provided IP address.

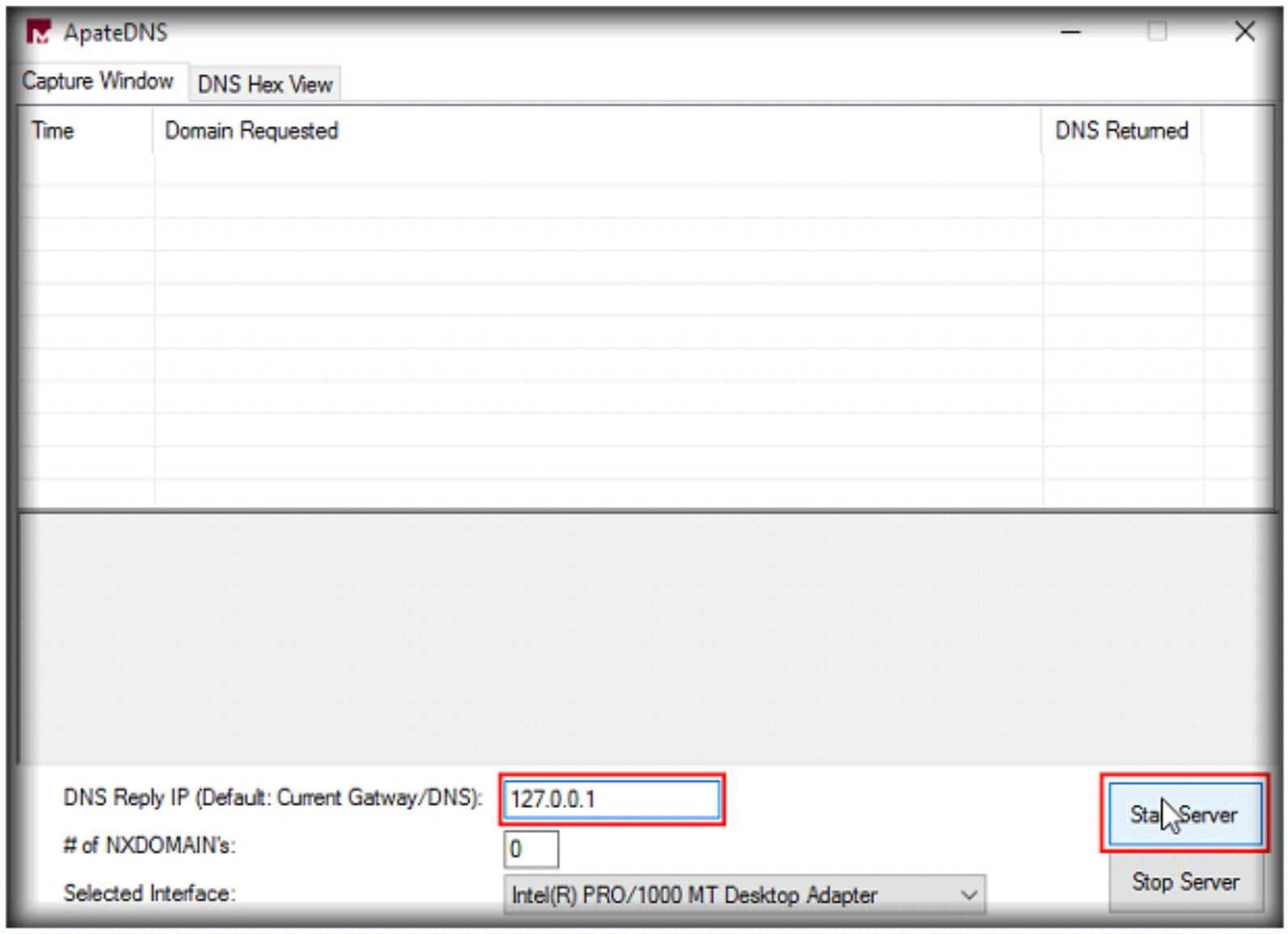


FIGURE 2.1: Running ApateDNS tool by giving a specific IP

5. Use the Command prompt and flush the DNS of the system to remove any previous cache history. Use the command: **ipconfig /flushdns**.

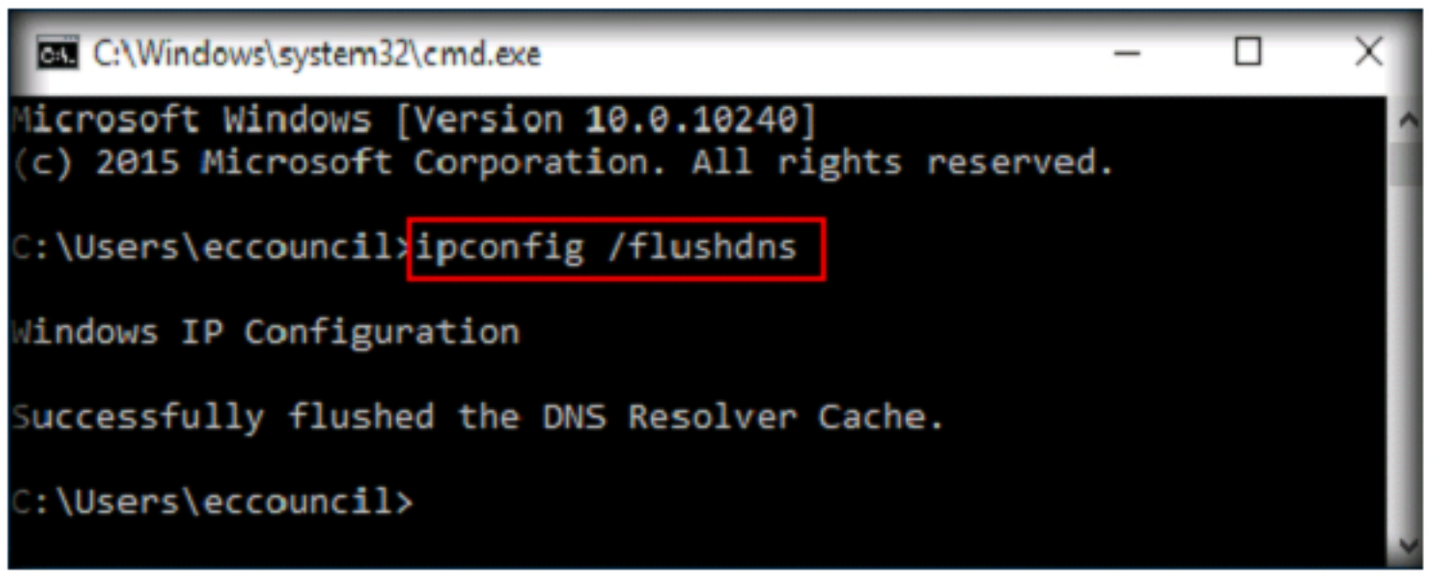


FIGURE 2.2: Flushing the DNS history

6. To launch the ProcMon tool, navigate to the **Z:\CHFIv9 Module 11 Malware Forensics\Dynamic Malware Analysis Tools\Process Monitor** location and double-click the **Procmon.exe** file.

7. **Process Monitor License Agreement** window appears, click **Agree**.

8. Process Monitor GUI appears as shown in the screenshot:

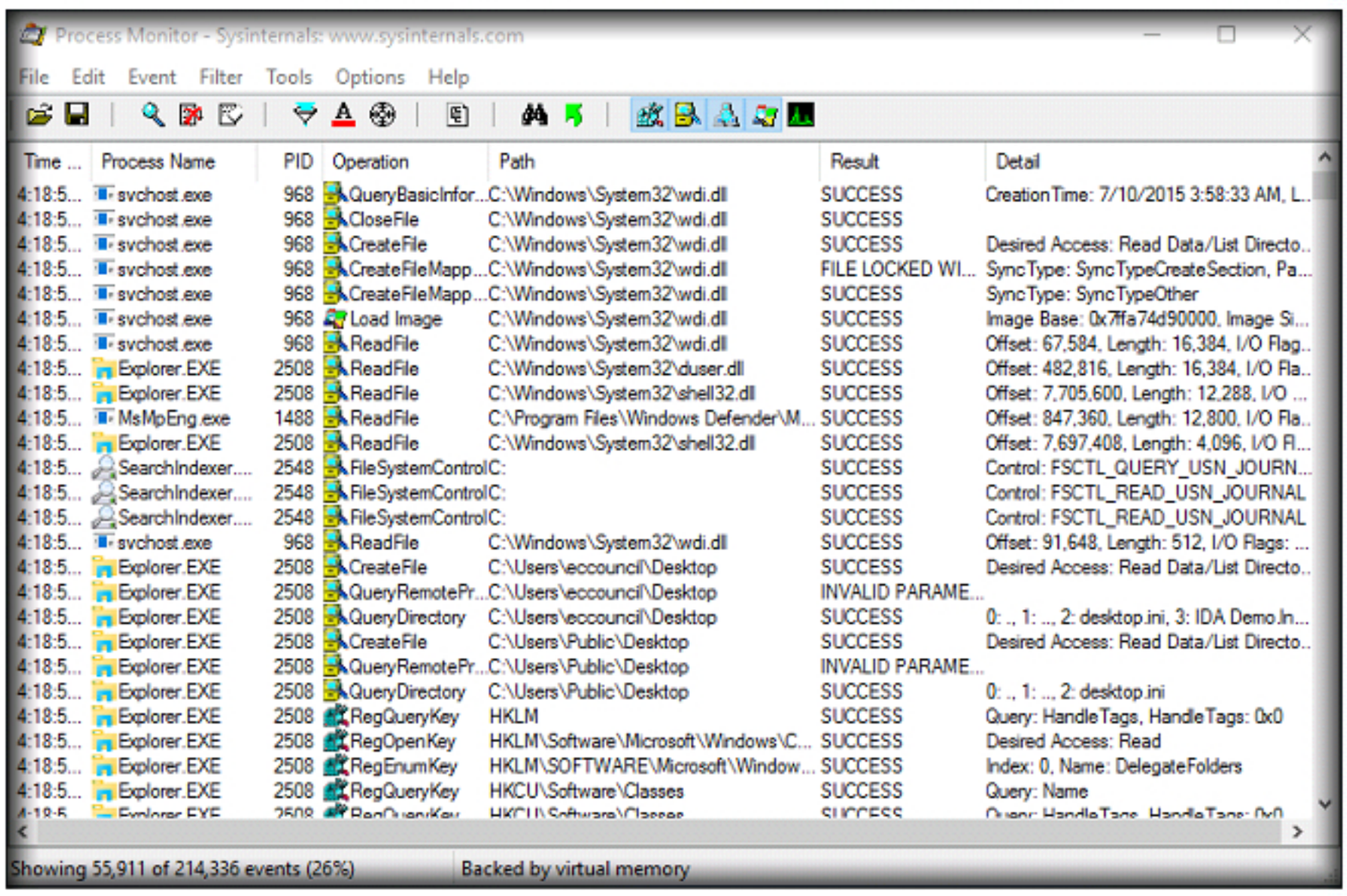


FIGURE 2.3: Launching Process Monitor tool

9. You can personalize the results a process monitor displays using the **Filter** option (shortcut: **Ctrl+L**) from the standard toolbar and specifying the type of file and providing name of the file in the Process Monitor Filter dialog box.

10. **Process Monitor Filter** window appears, select **Process Name** from the drop-down list, the name of the file as **Unknown.exe**, and click **Add** button.

11. Once done, click **OK** button.the

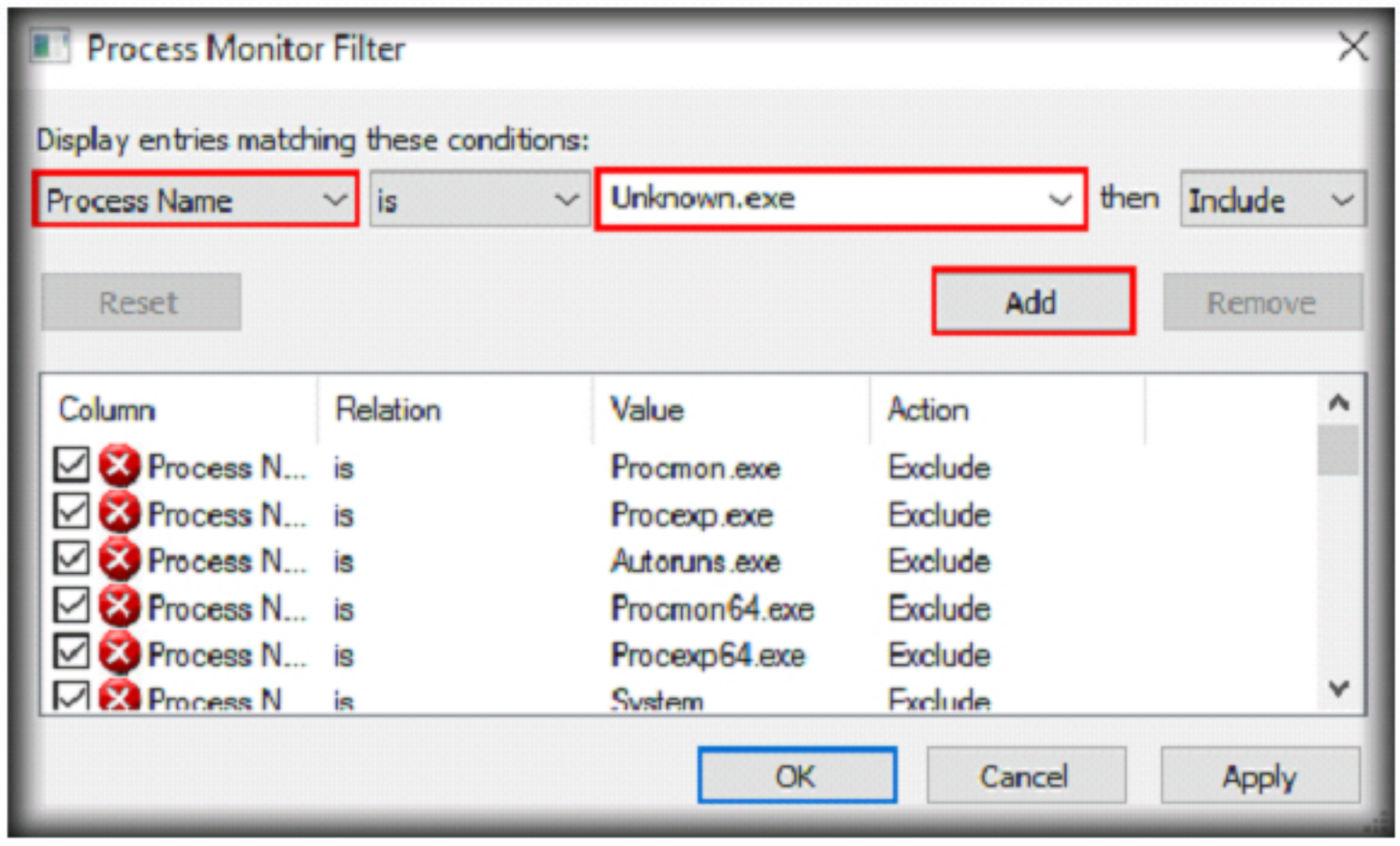


FIGURE 2.4: Adding filter to the process monitor

12. Navigate to **Z:\CHFIv9 Module 11 Malware Forensics\Dynamic Malware Analysis Tools\Process Explorer** and double-click **procexp.exe** to launch the **Process Explorer** tool.

13. **Process Explorer License Agreement** window appears, click **Agree**.

14. The tool will automatically start gathering information about all the running processes.

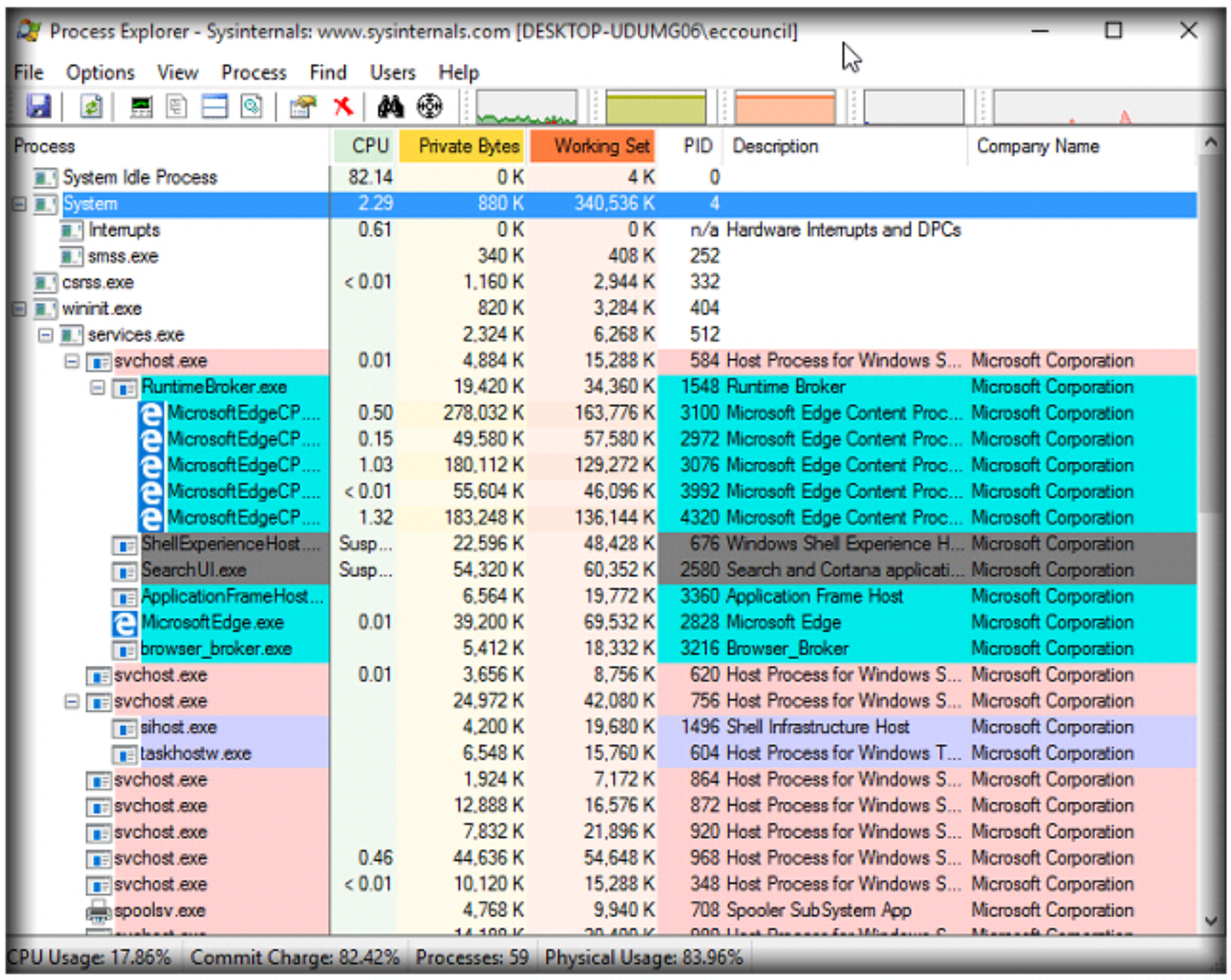


FIGURE 2.5: Launching Process Explorer tool

15. Navigate to **Z:\CHFIv9 Module 11 Malware Forensics\Network Activity Monitoring Tools\Wireshark**, double-click **Wireshark-win64-2.0.4.exe** and follow the wizard driven installation steps to install the application.

16. On completing the installation, launch Wireshark and begin packet capture on an interface which is common to all the other machines connected to Windows 10.

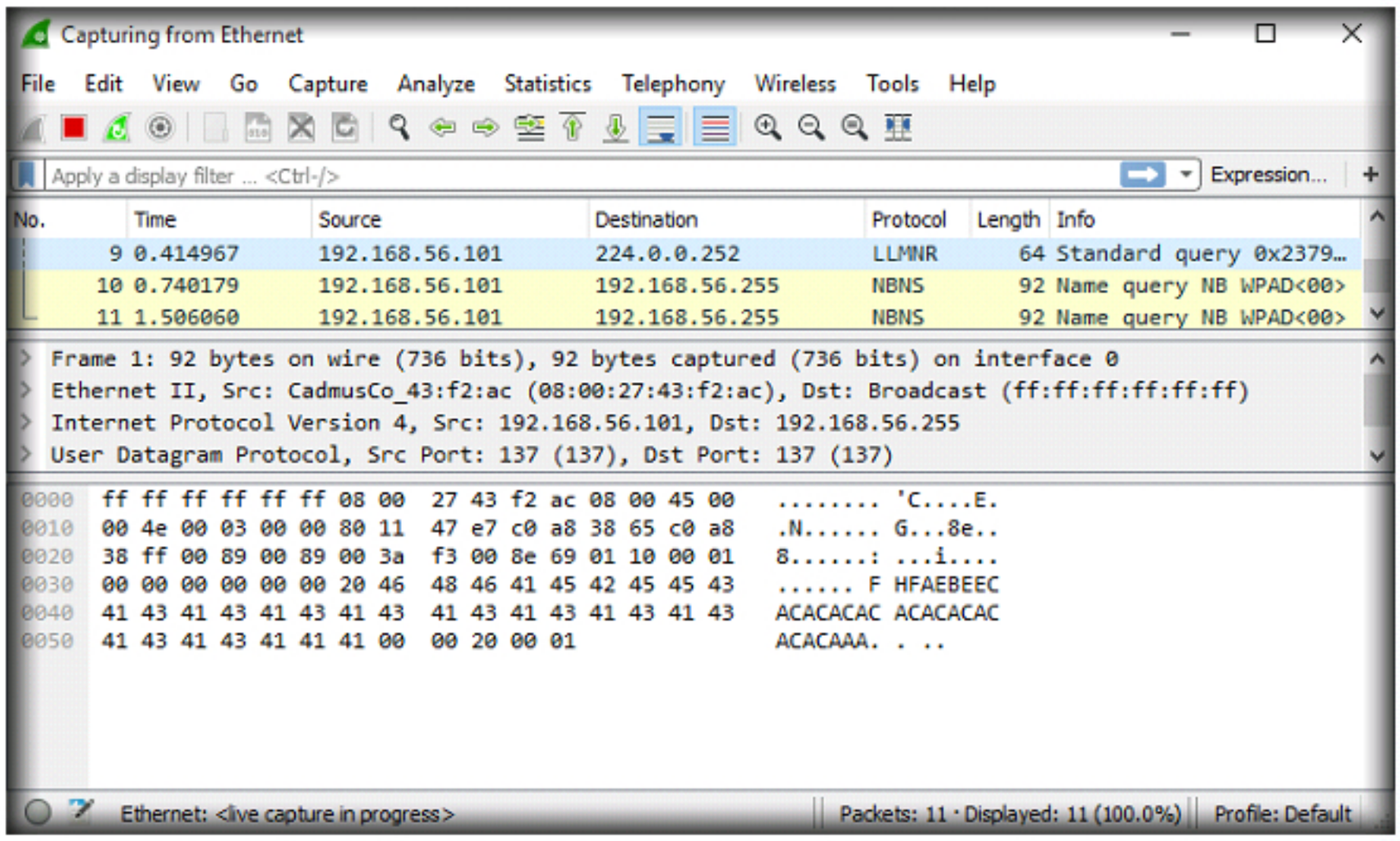


FIGURE 2.6: Launching Wireshark tool

17. Open command prompt. Navigate to the location, **Z:\CHFIv9 Module 11 Malware Forensics\Network and Internet Simulation Tools\nc111nt** and drag and drop the **nc.exe file** onto the command prompt console with the syntax for listening to port 80:

18. Syntax: **Executable File location\Filename -l -p 80**.

19. Press **Enter** and the tool will start recording all the activities performed over port 80.

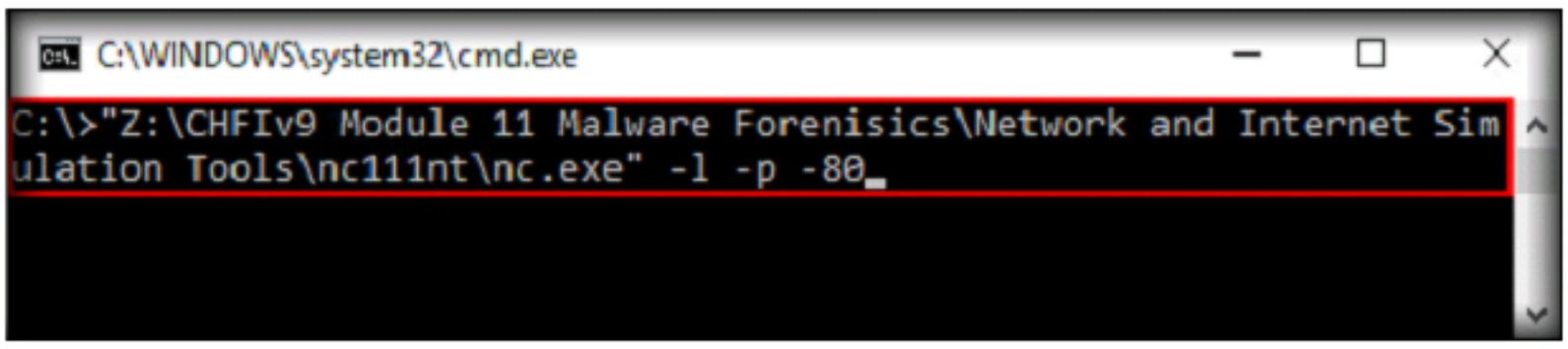


FIGURE 2.7: Running nc.exe file in command prompt

20. Navigate to the location, **Z:\CHFIv9 Module 11 Malware Forensics\Registry Entry Monitoring Tools\Regshot** and double-click the **Regshot-x64-ANSI.exe** file to run the Regshot tool.
21. Click on the **1st shot** button and click the **Shot** option from the list to capture the baseline of the system.

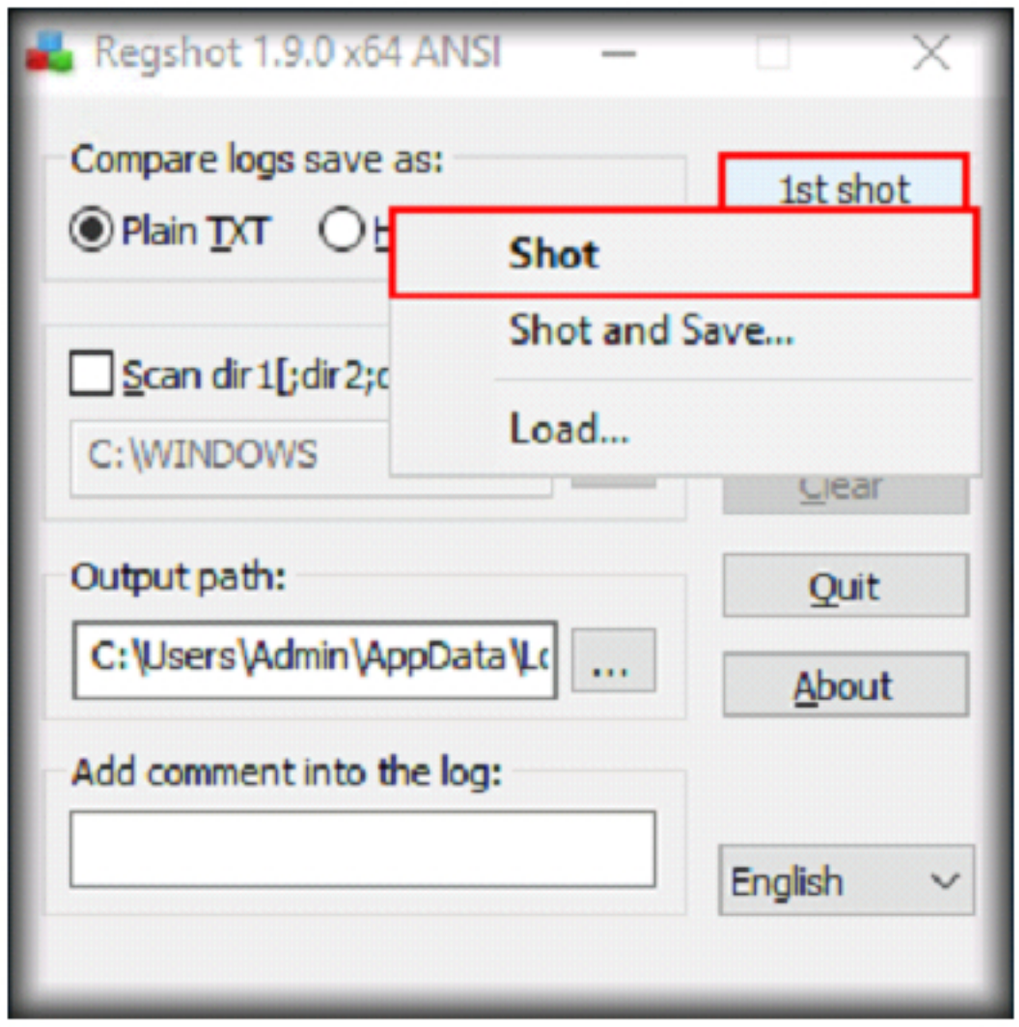


FIGURE 2.8: Capturing baseline of the system using Regshot tool

22. Navigate to the malware location **Z:\Evidence Files\Unknown** and double-click the **Unknown.exe** file to run the malware.

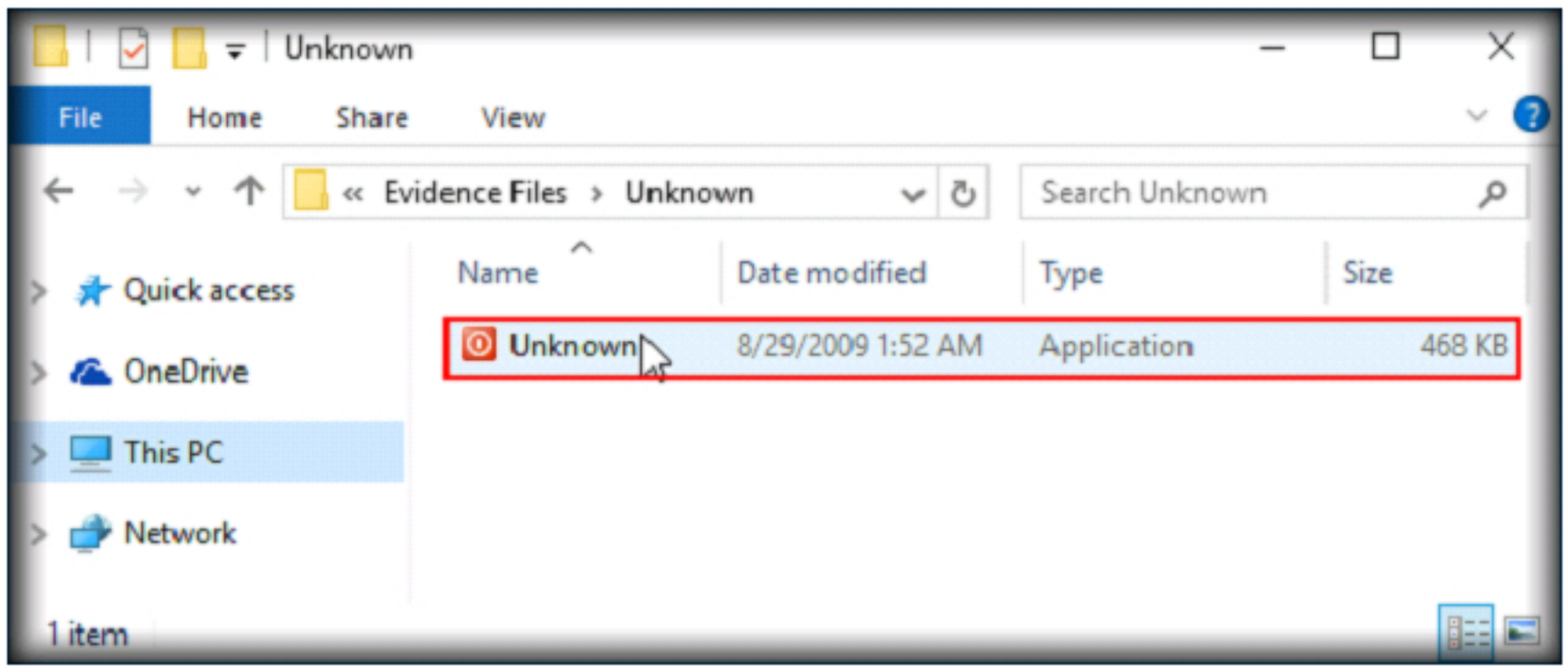


FIGURE 2.9: Executing the Malware using Regshot tool

23. Now, check for the changes tools observed in the system.

24. Go to the **Regshot** tool, click on the **2nd shot** button and click the **Shot** option from the list to capture another baseline of the system.

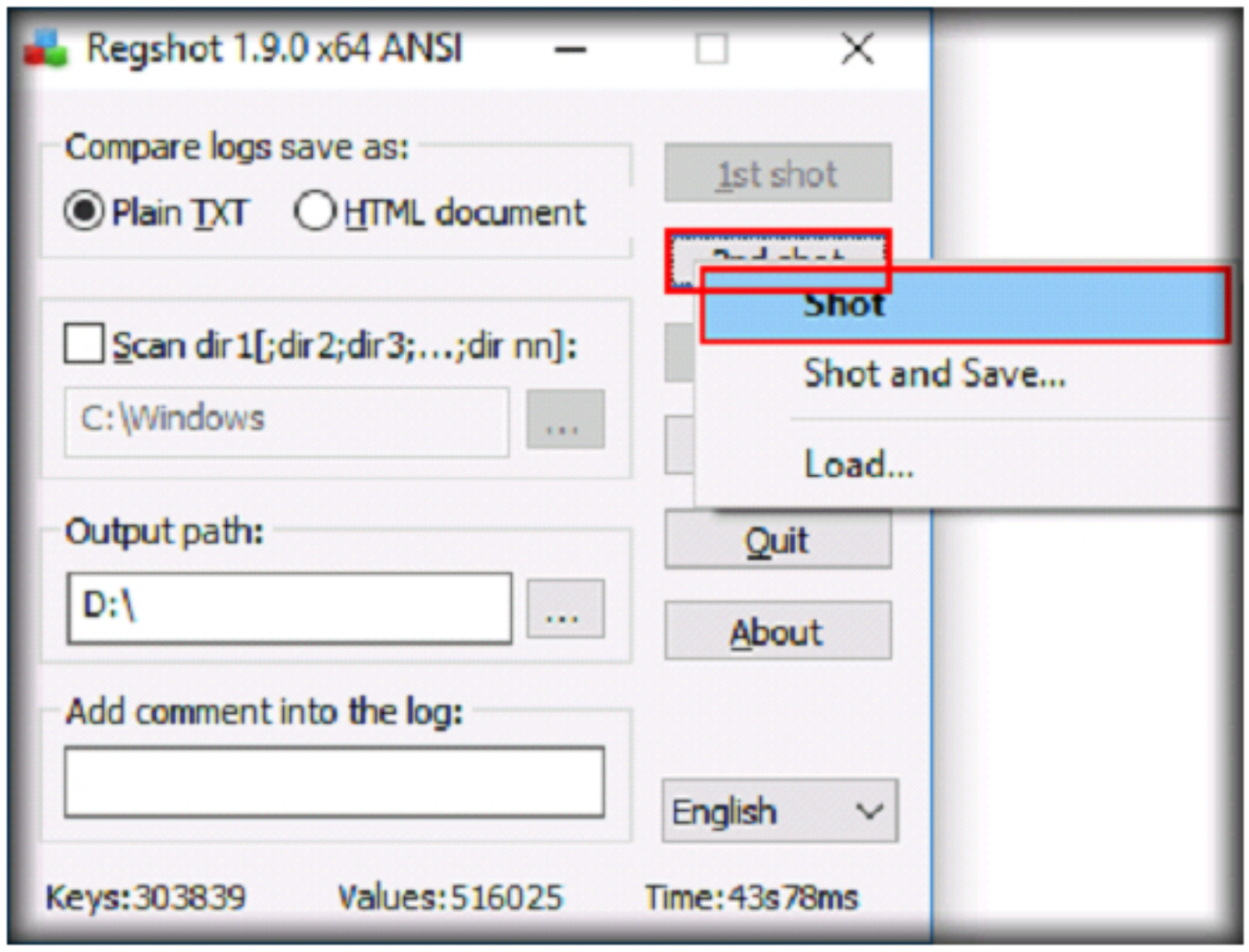


FIGURE 2.10: Capturing 2nd shot of the system

25. The tool will take a second shot of the complete system and enable the **Compare** button. Click the **Compare** button to compare the system status during the first shot with the status of the second shot.

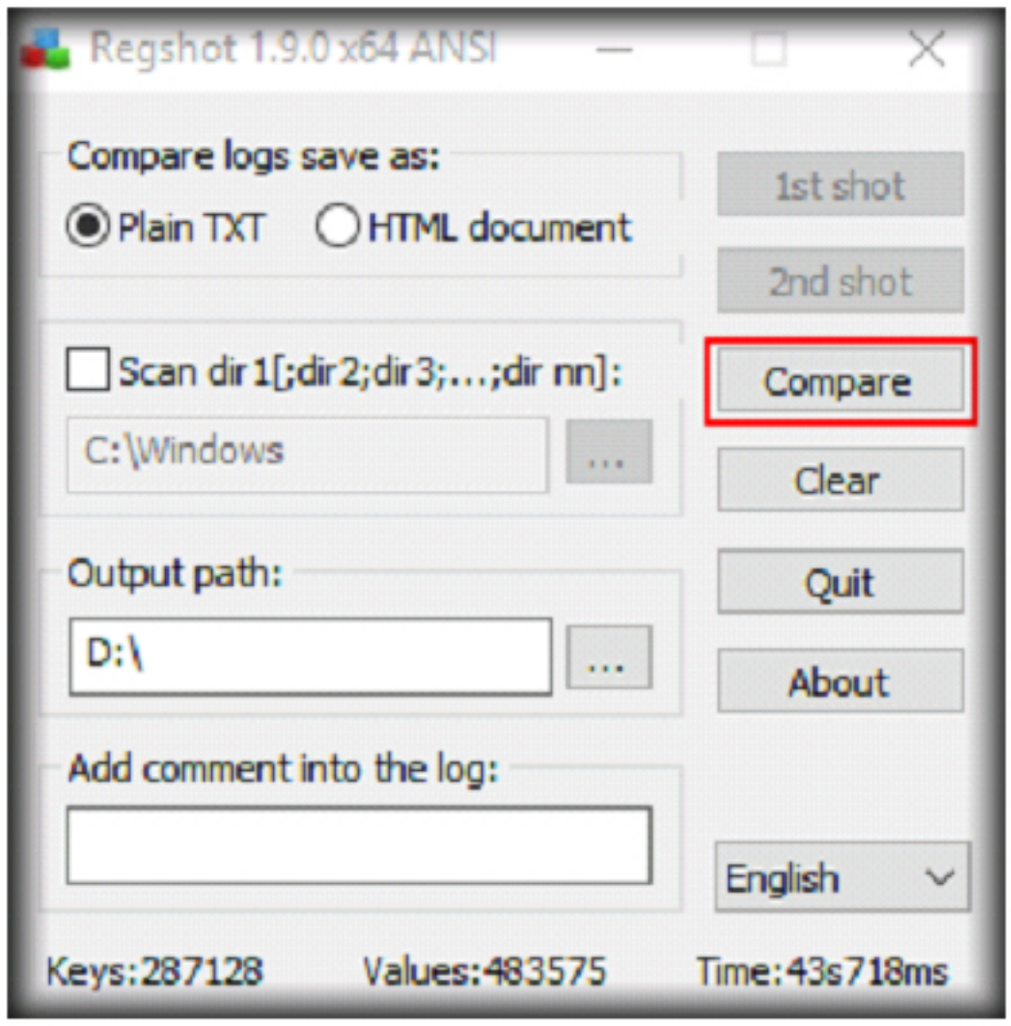


FIGURE 2.11: Comparing the 1st and 2nd shots

26. On completing the comparison, the tool will display a notepad containing information about all the changes to the system made by a suspicious file. The information includes file location and the hex value of the changes.

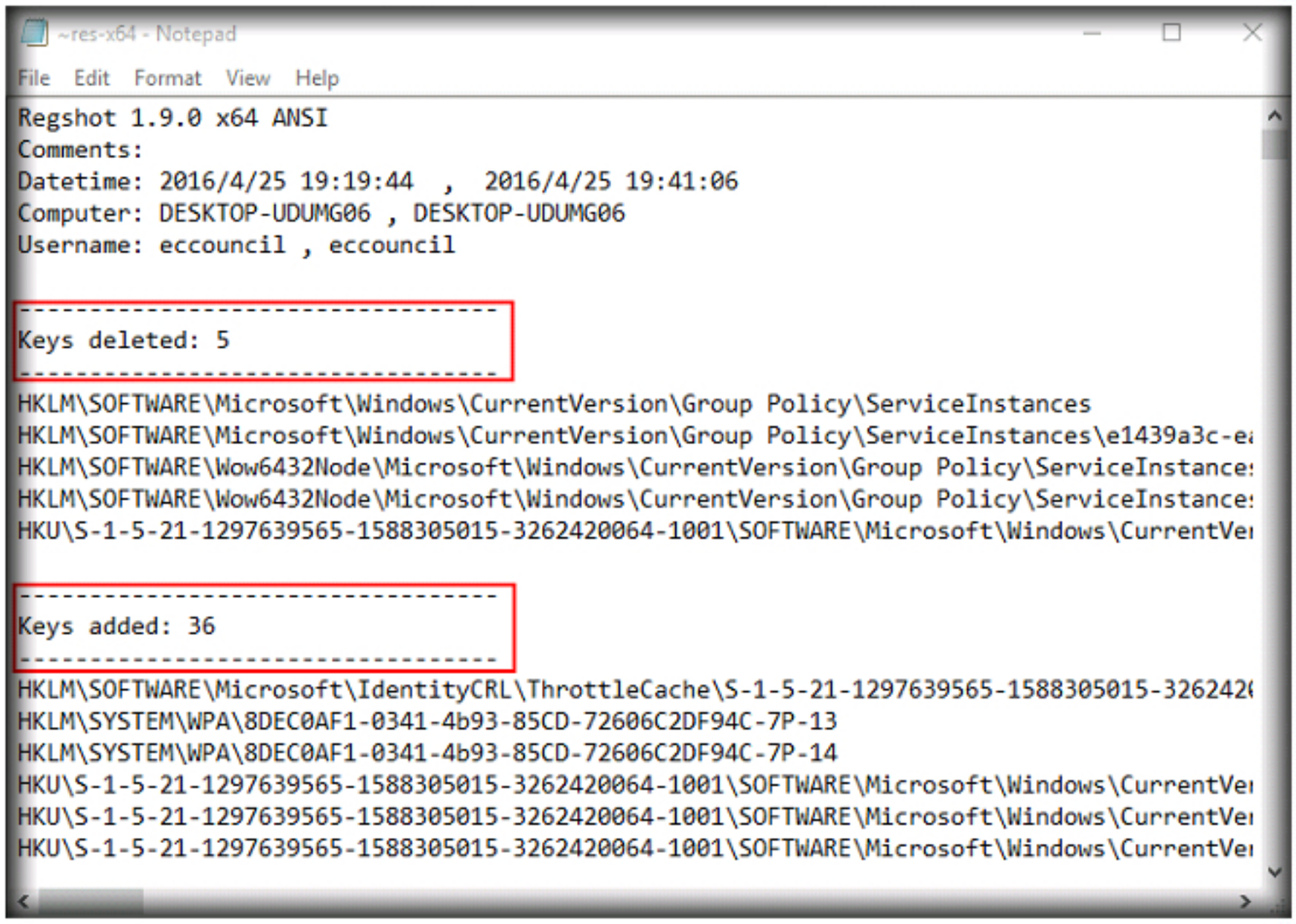


FIGURE 2.12: Examining the output of Regshot

27. Switch to Wireshark GUI. Wireshark provides the information about the connections malware tried to make as well as the data packets it tried to send or receive as shown in the screenshot:

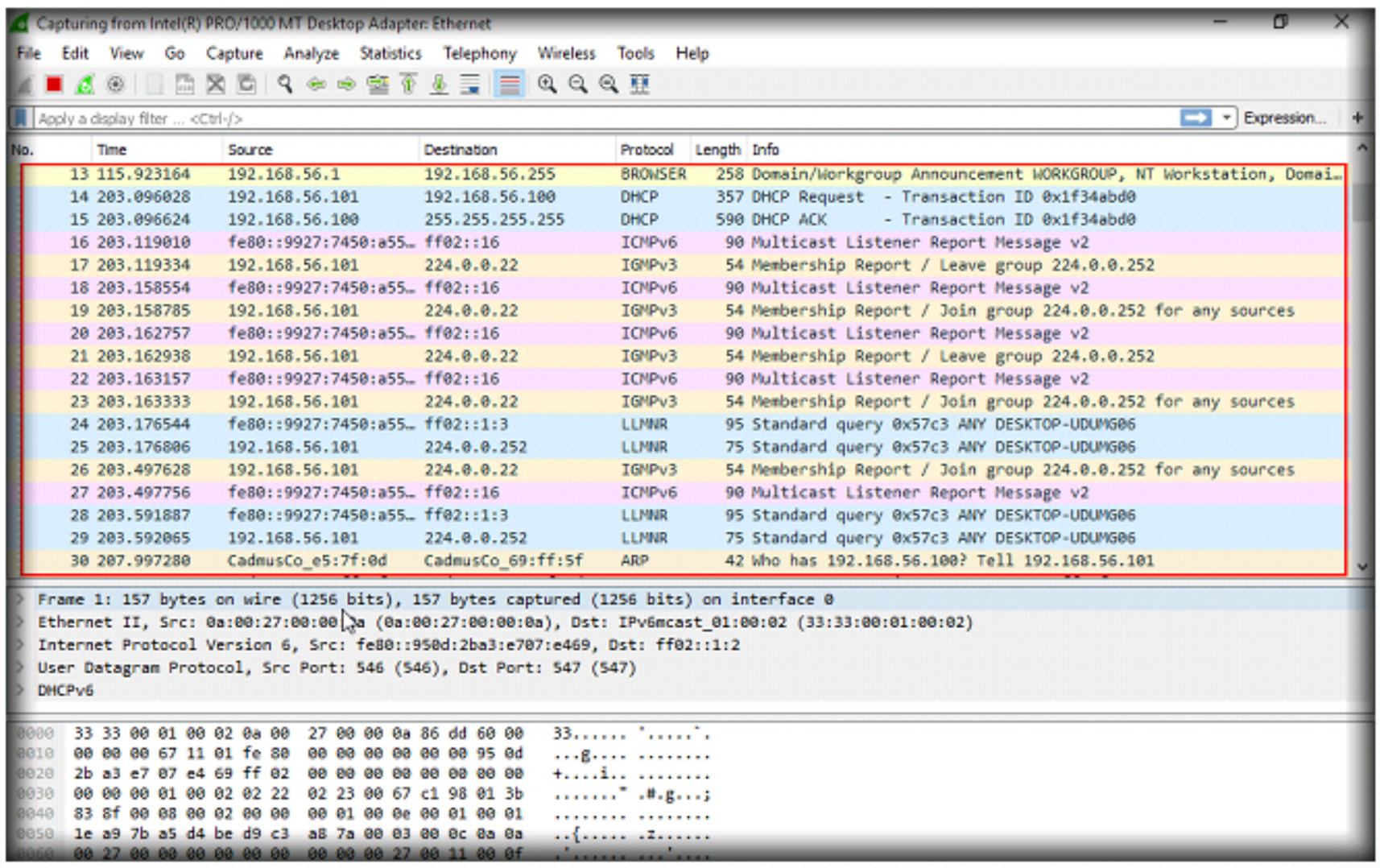


FIGURE 2.13: Examining network traffic using Wireshark tool

28. Check the **Process Monitor**. The tool will provide details about all the functions, processes, services and activities; the malware has started. It will also provide information about the flags it enabled and disabled.

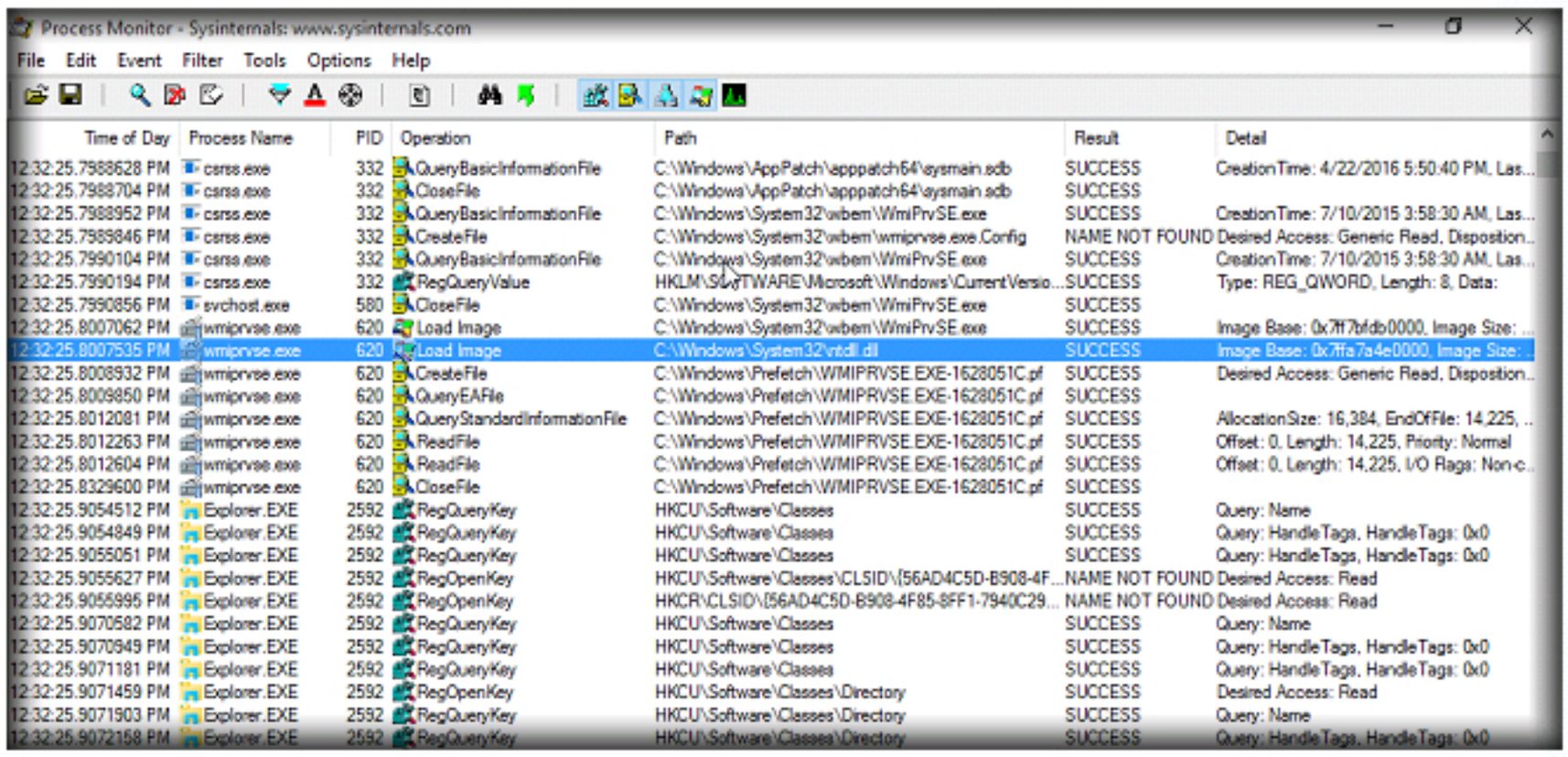


FIGURE 2.14: Output of Process Monitor

29. The Process Explorer displays details of all the process including the ones the suspicious file initiated. Check for the unmarked processes that do not have a company name.

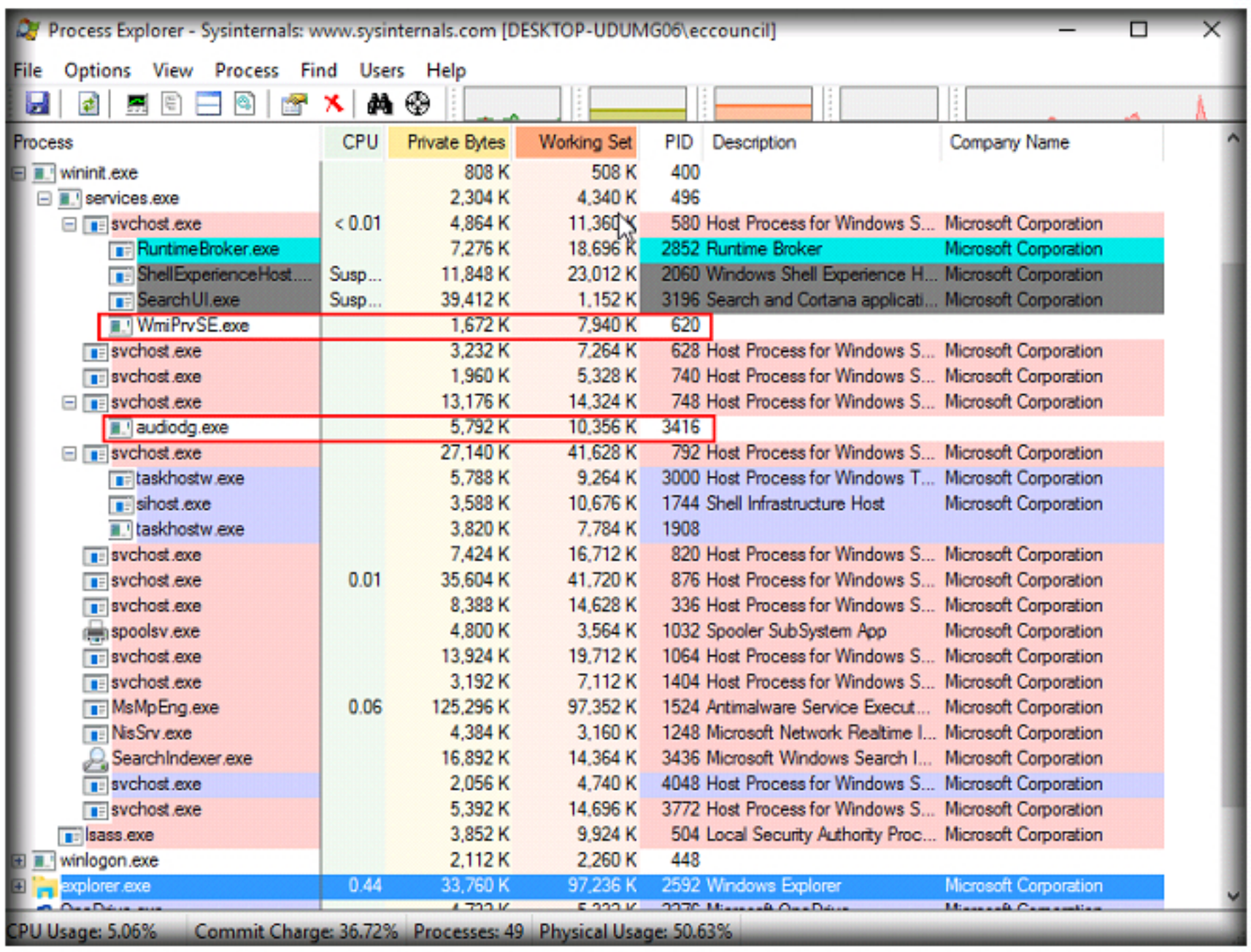


FIGURE 2.15: Output of Process Explorer

30. The **ApateDNS** tool displays all the DNS requests from the malware. It will show the IPv4 and IPv6 addresses of domains, the malware tried to connect with.

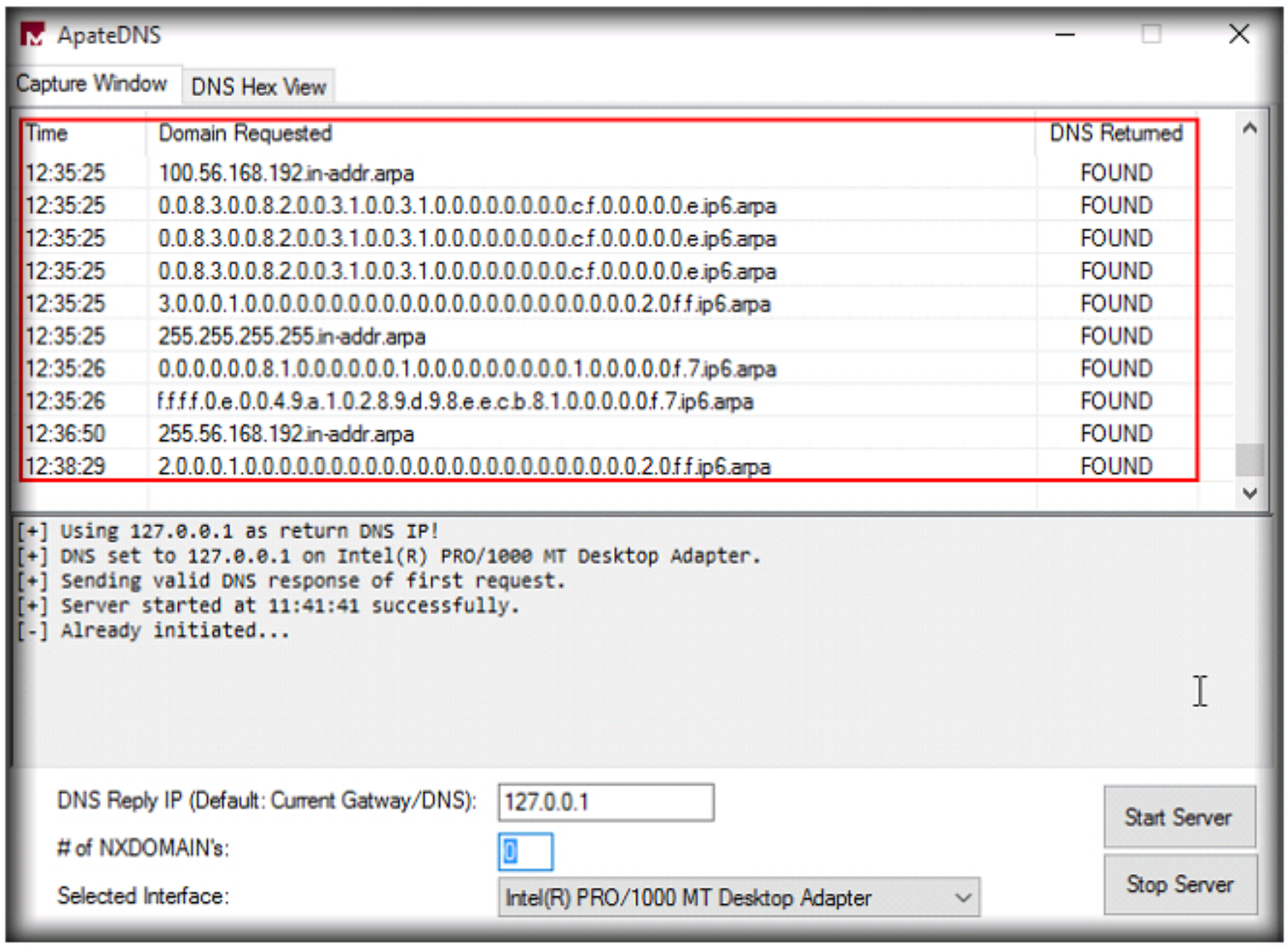


FIGURE 2.16: Output of ApateDNS

Lab Analysis

Document all the findings obtained during this lab.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab

3

Analyze Malicious PDF File

Portable Document Format (PDF) is a file format that supports structured storage and embeds graphics, images, strings, links, etc. These characteristics of the format allow perpetrators to embed malicious strings and scripts with the document.

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

Lab Scenario

Attackers can hide malware in PDF files, and it starts executing the program when users try to open the file. This type of malware can avert system and network security easily and install malicious programs on the system. Investigators need to run different scans on a suspicious PDF file to identify the presence of malicious scripts and extract them to find their impact.

Lab Objectives

This lab will give you real-time experience in identifying the malicious PDF files and assessing their impact on the system and the network.

Lab Environment

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv9 Module 11 Malware Forensics

To carry out this lab, you need:

- PDF scanning tools, such as PDFid, peepdf and PDF Stream Dumper, located at **Z:\CHFIv9 Module 11 Malware Forensics**.
- A computer running Windows Server 2012 Host machine or Oracle Virtual Box.
- A Windows 10 virtual machine.
- Python interpreter installed on the Windows 10 VM.
- A web browser with Internet access.
- Administrative privileges to run the tools.

Lab Duration

Time: 25 Minutes

Overview of Analyzing Malicious PDF

PDFs are the most commonly shared files due to their small size and platform independence. Attackers use these PDFs to send hidden malicious scripts. An investigator must be aware of the process of identifying a malicious PDF file.

Lab Tasks

TASK 1

Analyze Malicious PDF

1. In this lab, your primary task is to upload the suspicious PDF file to online virus databases, such as www.virustotal.com and check if these databases contain signatures of virus in the file which you are going to upload.
2. Login to Windows 10 machine, launch a web browser and browse https://www.virustotal.com. VirusTotal homepage appears, click on **Choose File** to upload the files.

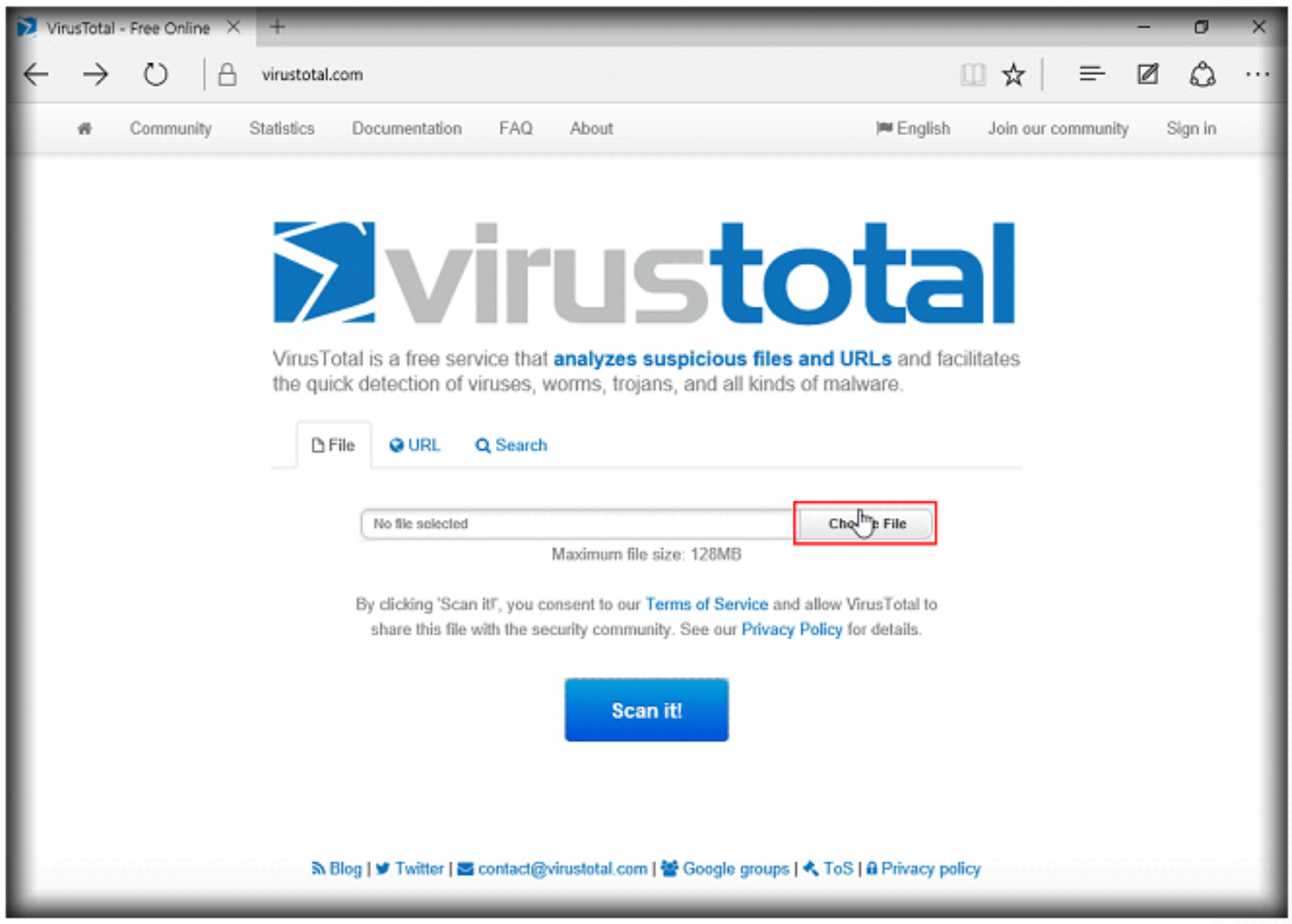


FIGURE 3.1: VirusTotal home page

3. **Open** window appears, navigate to the location, **Z:\Evidence Files**, select the suspicious file (**infected.pdf**) and click **Open** to upload the file.

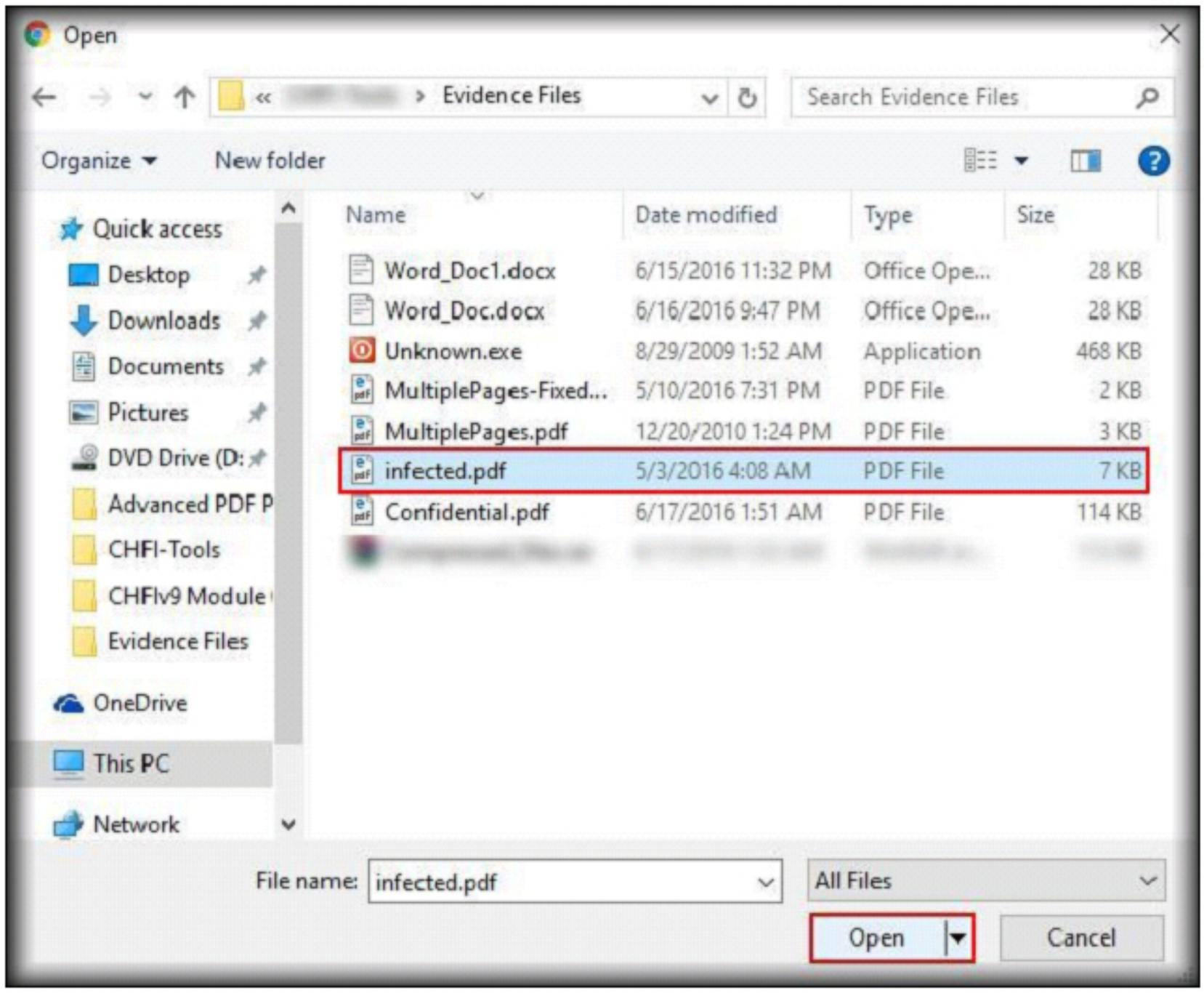


FIGURE 3.2: Navigating to the Evidence Files folder and selecting the suspicious file

4. Click the **Scan it!** button on the web page.

FIGURE 3.3: Scanning the suspicious file

5. The website will scan the database for similar files and display the result including the SHA256 hash value of the file, Detection ratio as well as the antivirus programs that identify the file as suspicious.

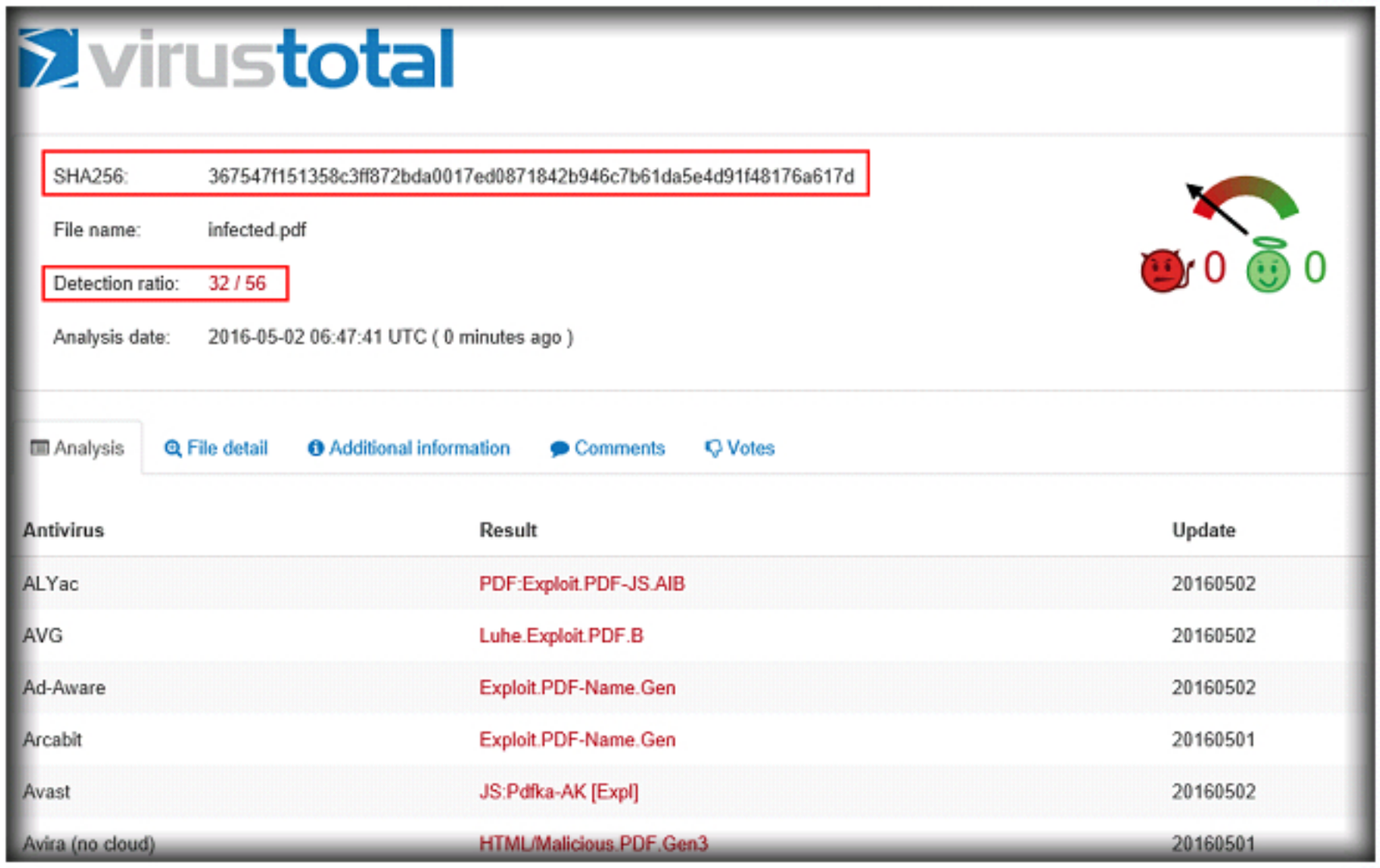


FIGURE 3.4: Analyzing the results after completion of scanning

6. The website also has an inbuilt version of the PDF scanning tool, **PDFiD**. Click on the File detail tab to open further analysis details.
7. In this tab, the website displays PDFiD information, which specifies the details of contents that make the file malicious.

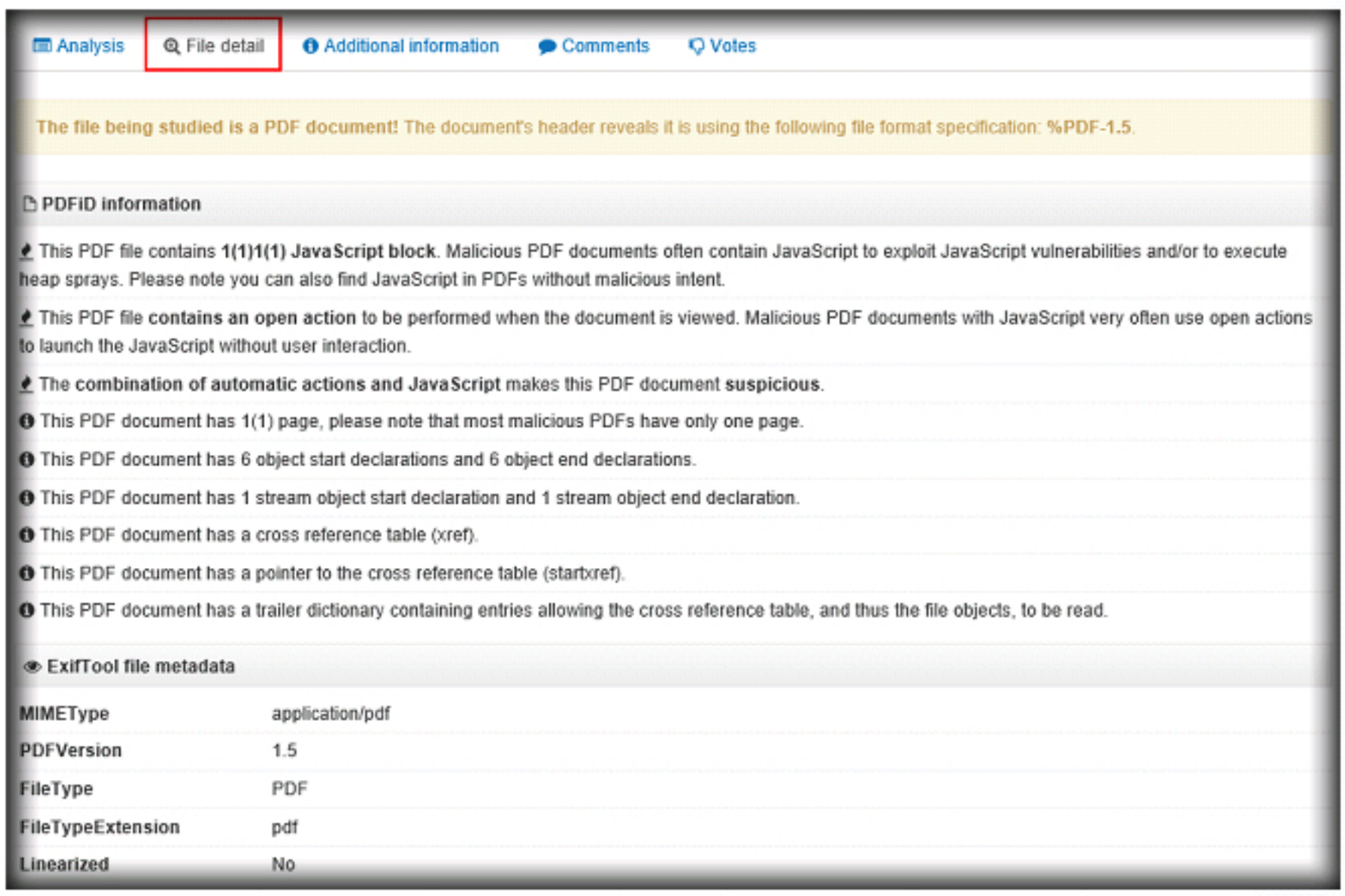


FIGURE 3.5: Viewing the results from the File detail tab

8. Click on the Additional information tab to view more details of the file. This tab displays information such as MD5 hash value and the tags that render the file as malicious.

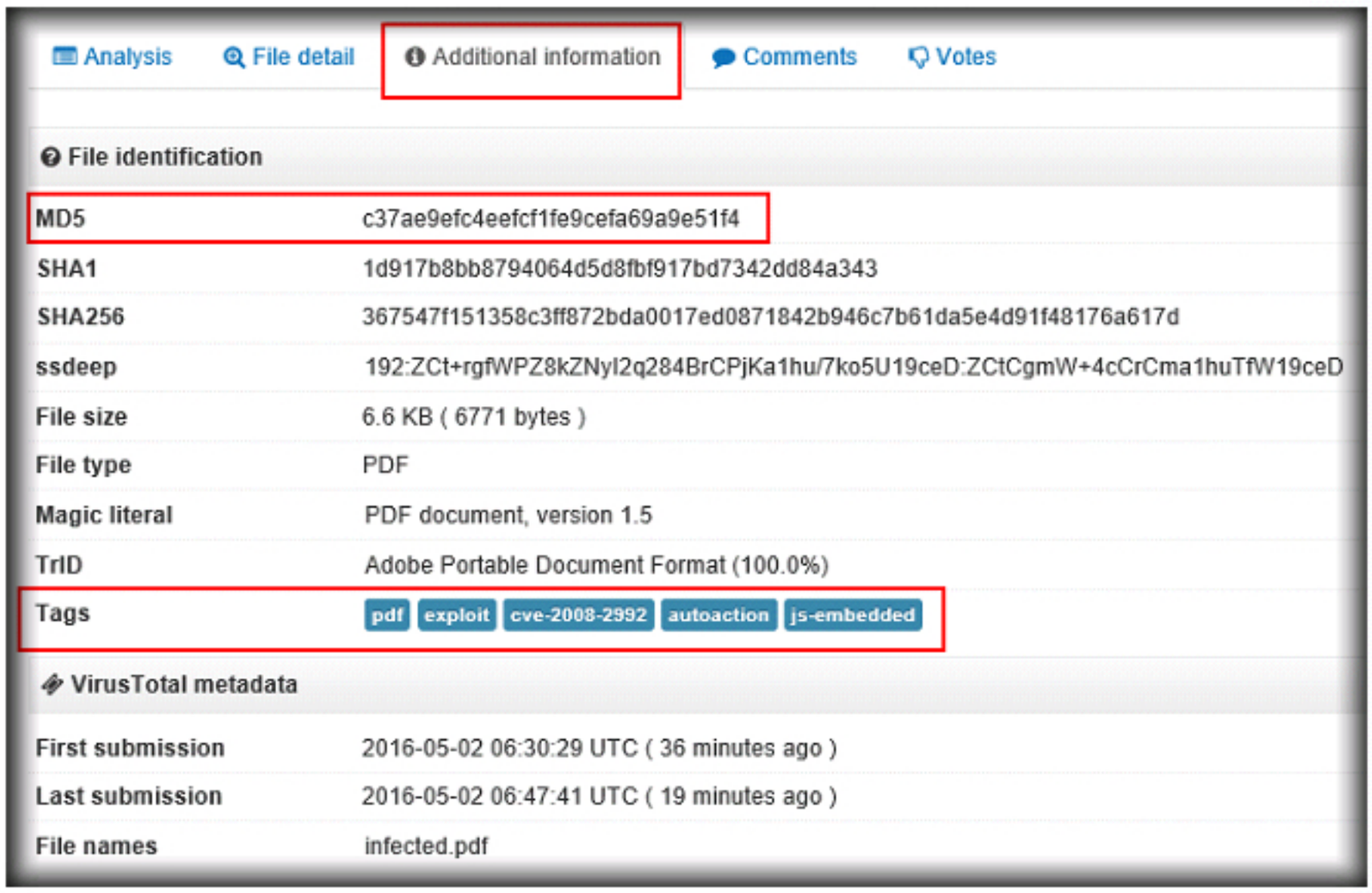


FIGURE 3.6: Viewing the results from the Additional information tab

9. Investigators need to analyze the structure of the suspicious PDF file to check the headers, objects, scripts and end of the file. To view the structure of a PDF file, use the python based tool, PDFiD, which comes pre-installed with Kali Linux virtual machine.
10. So, login to Kali Linux machine, Launch a command line terminal and type **pdfid{Space}**.
11. Now, navigate to the Home folder, click **Other Locations** in the left pane.
12. Other Locations window appears, displaying the Machine Names of the machines that are found on the network.
13. Double-click the Windows machine Name associated with Windows Server 2012 which contains CHFI-Tools, to connect to the machine. In this lab, the Windows machine selected is **win-v75vghtull**.

Note:

a) The machine name might vary in your lab environment

b) If you are unable to find the machines, click on **Home** and then, click on **Other Locations**

14. All the drives associated with the machine appear. Navigate to **E:\Evidence Files**.

15. Now, keeping the window intact drag and drop **infected.pdf** file onto the terminal, and press **Enter**.

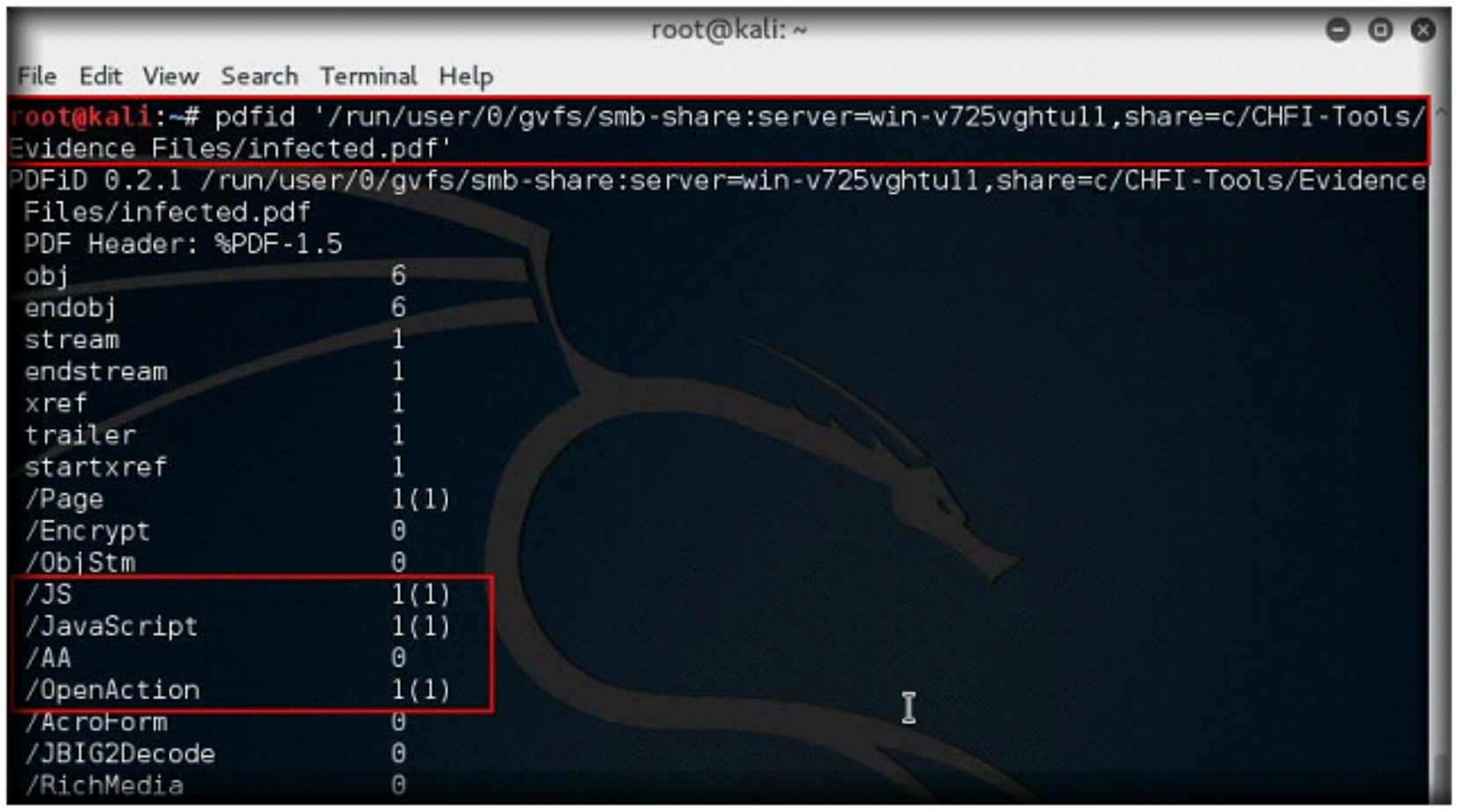


FIGURE 3.7: Running infected.pdf file in Linux terminal using pdfid

16. The tool will display the structure of PDF starting with the header containing the version of PDF used in making the file, contents of the file, end of the file (EOF).

17. The tool will also display all the scripts present in a file including malicious scripts such as JavaScript and OpenAction.

18. Now, we shall use Python-based pdf-parser tool to look at the elements of the PDF file.

19. Type **pdf-parser{Space}** in the terminal, drag and drop **infected.pdf** onto the terminal and press **Enter**. You can observe that the file contains OpenAction script in it as shown in the screenshot:

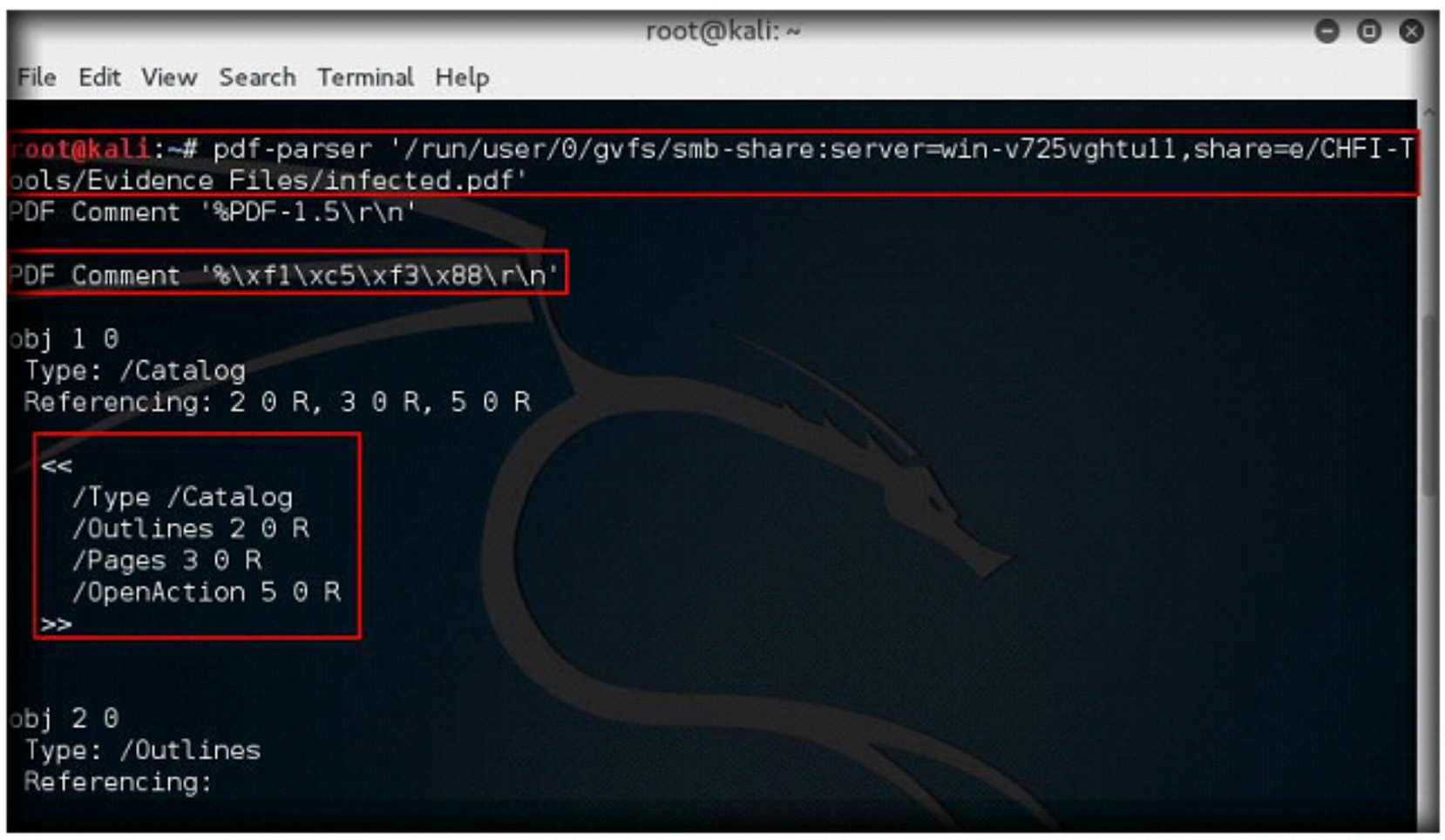


FIGURE 3.8: Running infected.pdf file in Linux terminal using pdf-parser

20. Scroll down to see the object embedded with the javascript. Other details include length and filter of the PDF file.

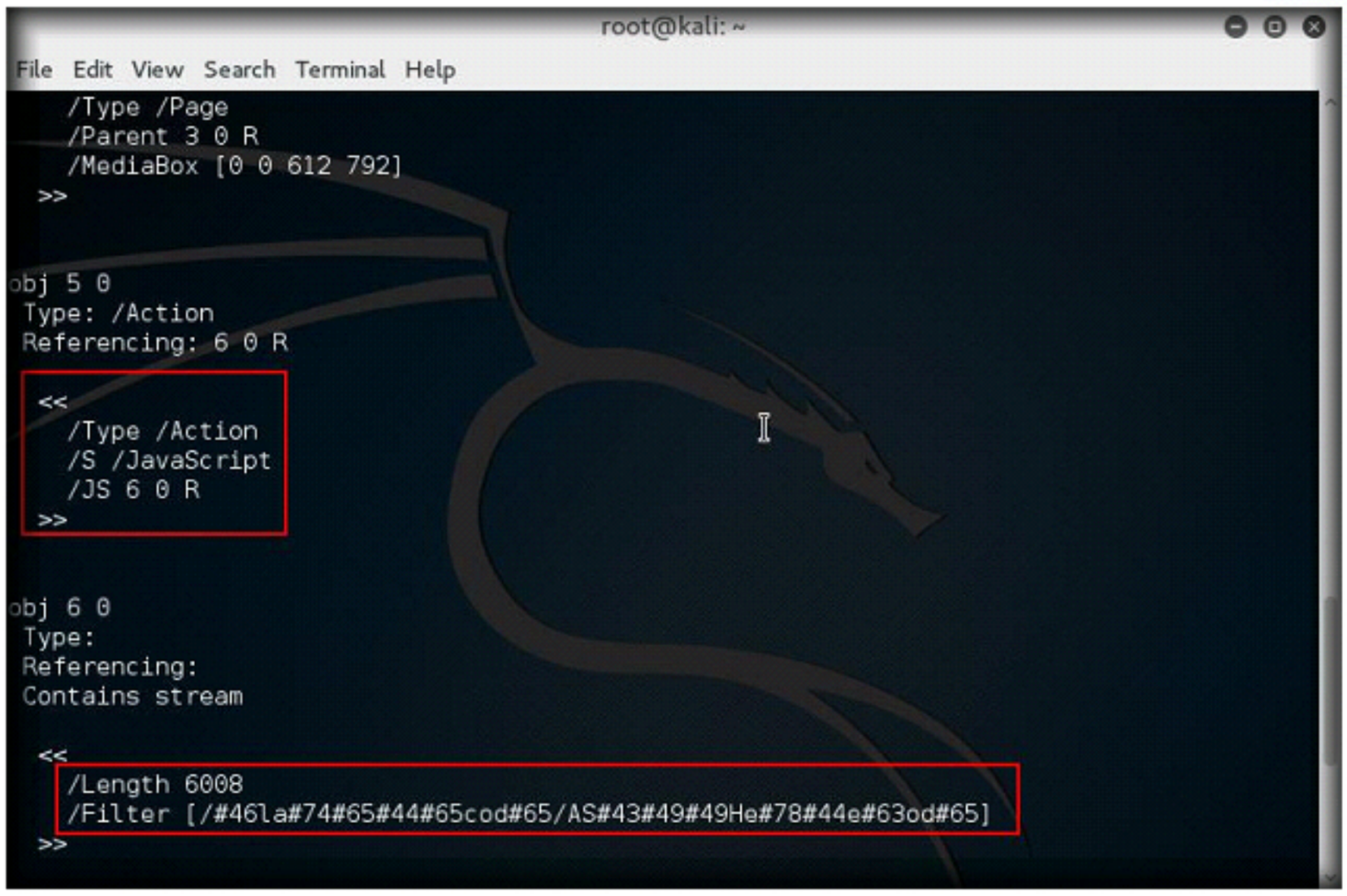


FIGURE 3.9: Viewing the object embedded with the javascript

21. To view the statistics of the file, type **pdf-parser --stats{Space}**, drag and drop **infected.pdf** onto the terminal and press **Enter**. This displays the statistics of the file as shown in the screenshot:

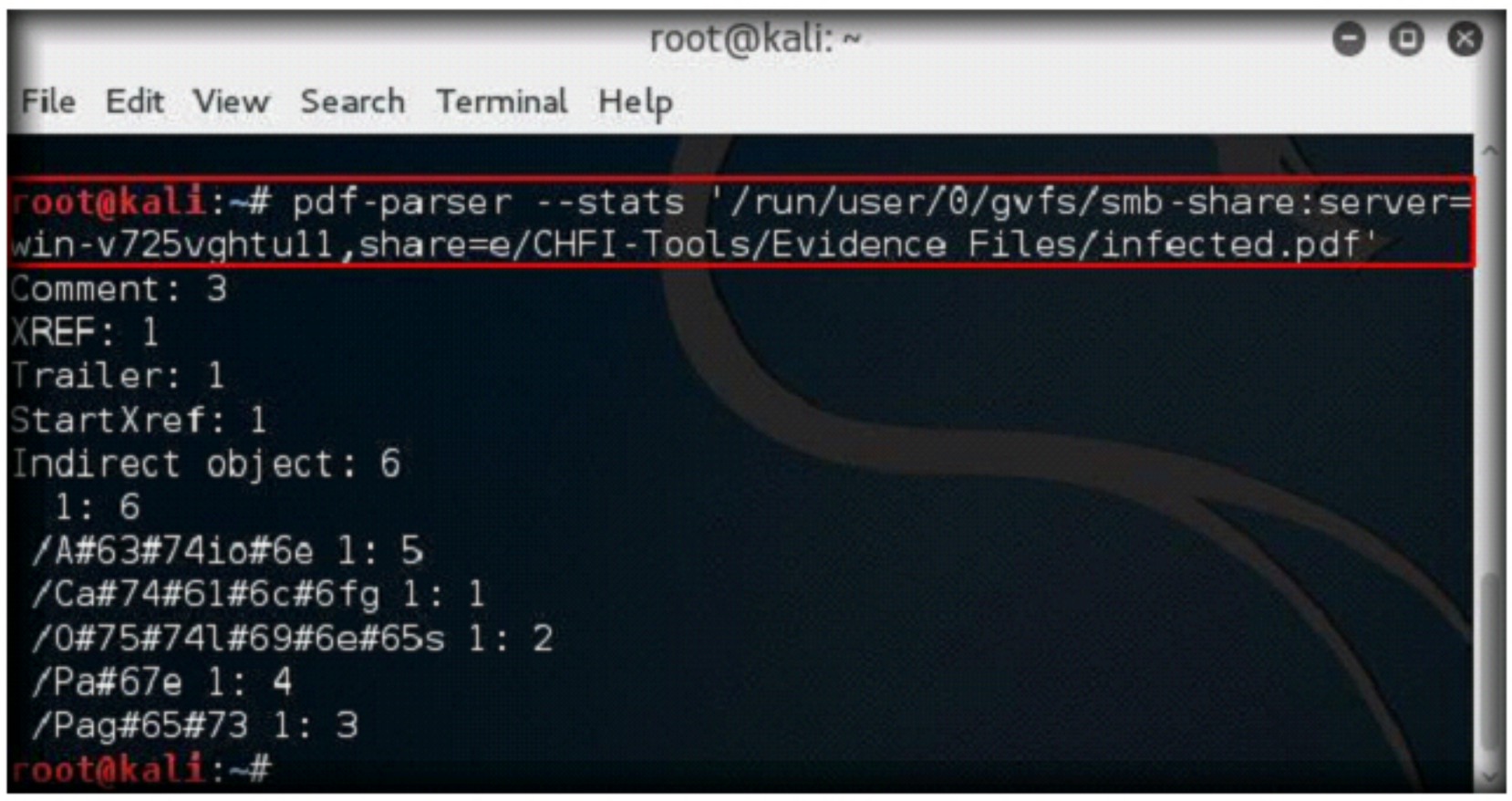


FIGURE 3.10: Executing a sample command

22. Investigators need to assess the PDF file to view the JavaScript, low-level PDF headers, objects, and shellcode present in it. Use the PDF Stream Dumper to analyze the file.

23. Navigate to the location, **Z:\CHFIv9 Module 11 Malware Forensics\Malicious Document Analysis Tools\PDFStreamDumper** and double-click the **PDFStreamDumper_Setup.exe** file to install the tool.

24. During installation, you will be prompted to download and install **AC3 Sorcerer Web Install** application. Click **No Thanks - Exit** button to avoid the application installation.

25. Launch the tool from **Desktop** or **System app menu**. Click the **Load** option from the toolbar and select **Pdf File** from the provided list.

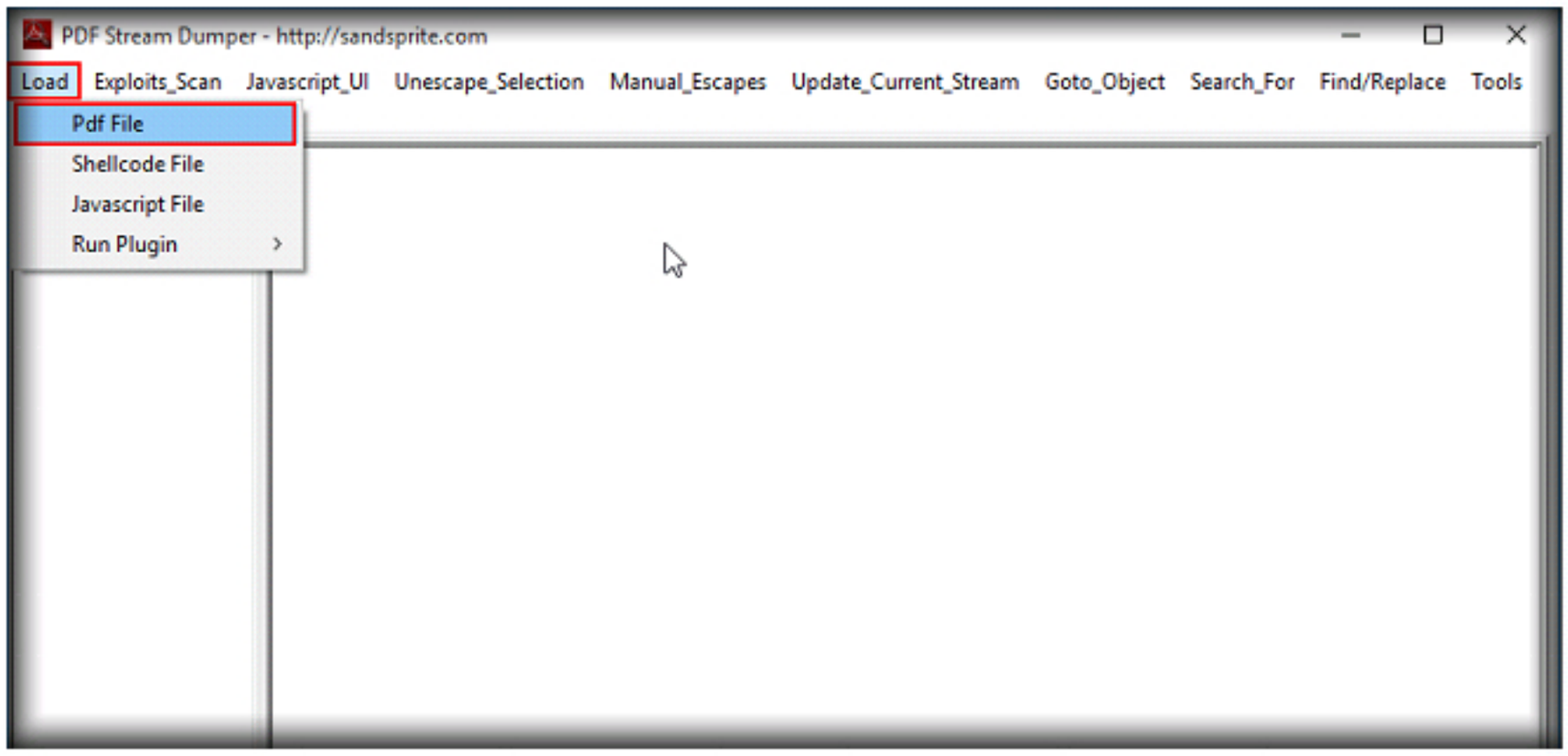


FIGURE 3.11: Launching and loading Pdf file into PDF Stream Dumper tool

26. Navigate to the suspicious file's location **Z:\Evidence Files**, select **infected.pdf** and click **Open** to upload the file to PDF Stream Dumper.

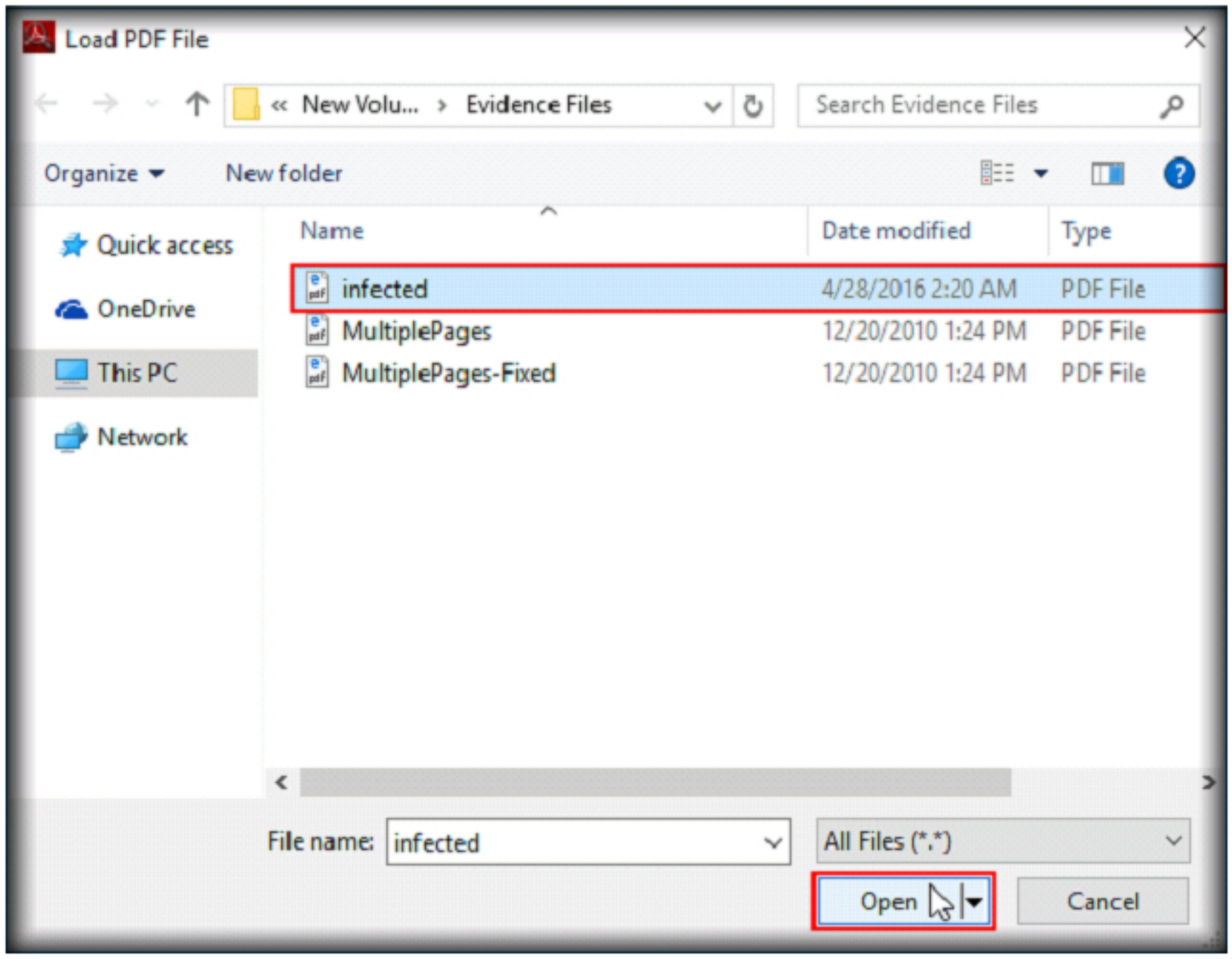


FIGURE 3.12: Navigating to the Evidence Files folder and selecting the suspicious Pdf file

27. **PDF Stream Dumper** analyzes the PDF file and displays its contents such as objects, streams, encrypted or obfuscated elements, etc. Other details include the HexDumps and stream details of the selected object, as well as the internal contents with structure.

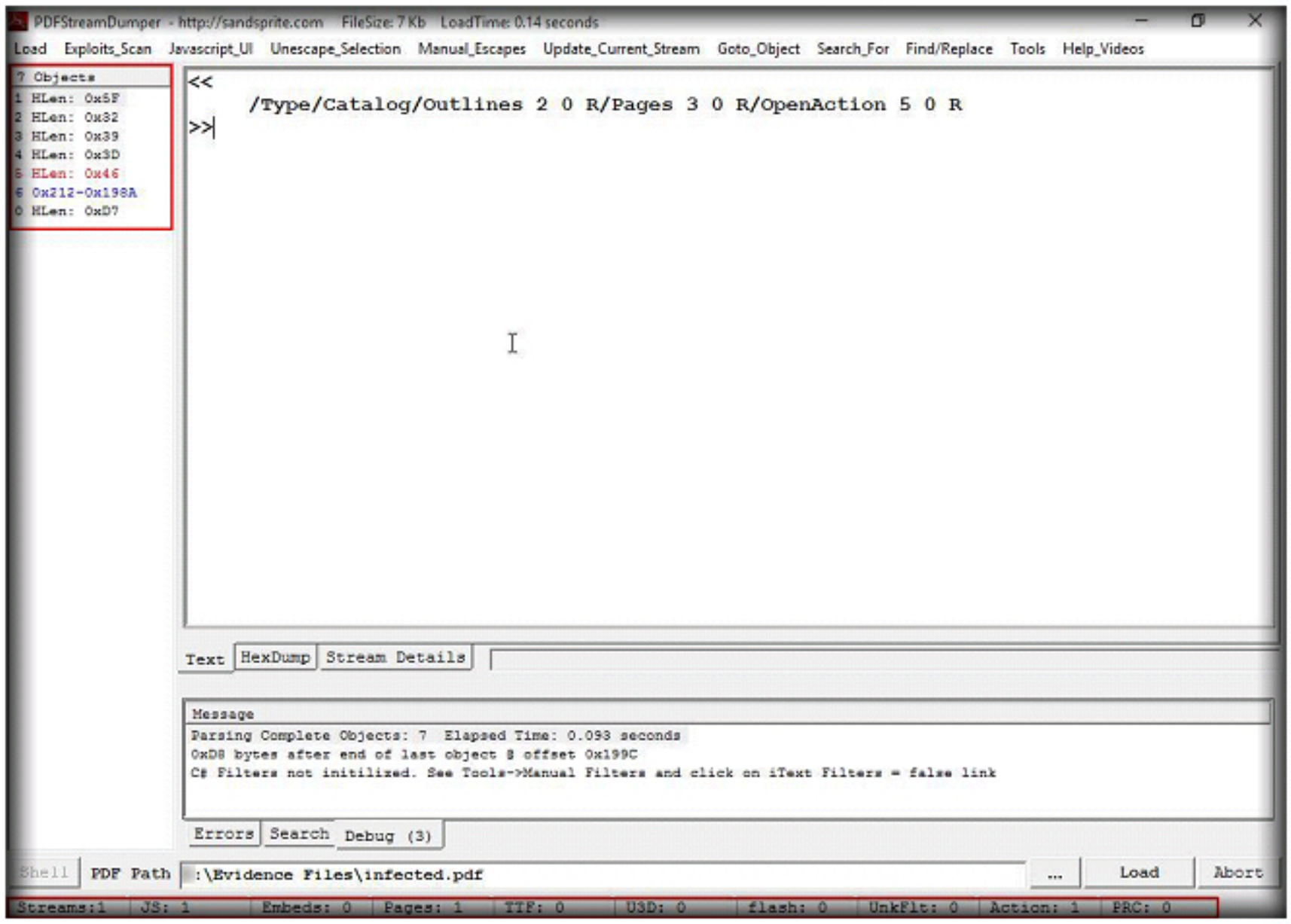


FIGURE 3.13: Viewing the contents of the suspicious Pdf file

28. The tool displays objects in different colors depending on its contents. To view the color specifications, click the **Tools** option from the toolbar and select the option **About Listview Colors**.

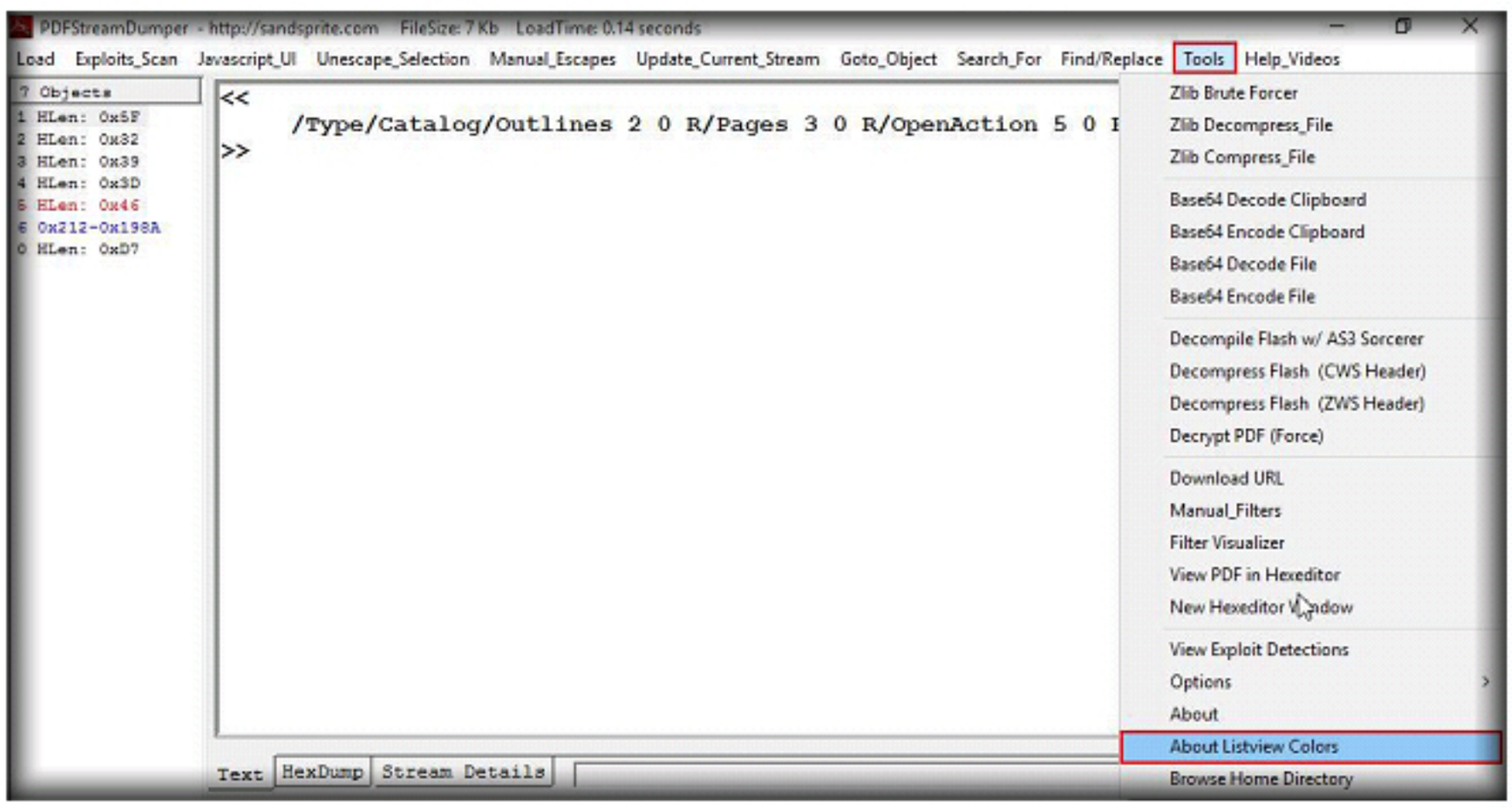


FIGURE 3.14: Screenshot representing the process to view the color specification

29. **PDFStreamDumper** dialog-box appears, displaying the color specification. Click **OK** to close the dialog-box.

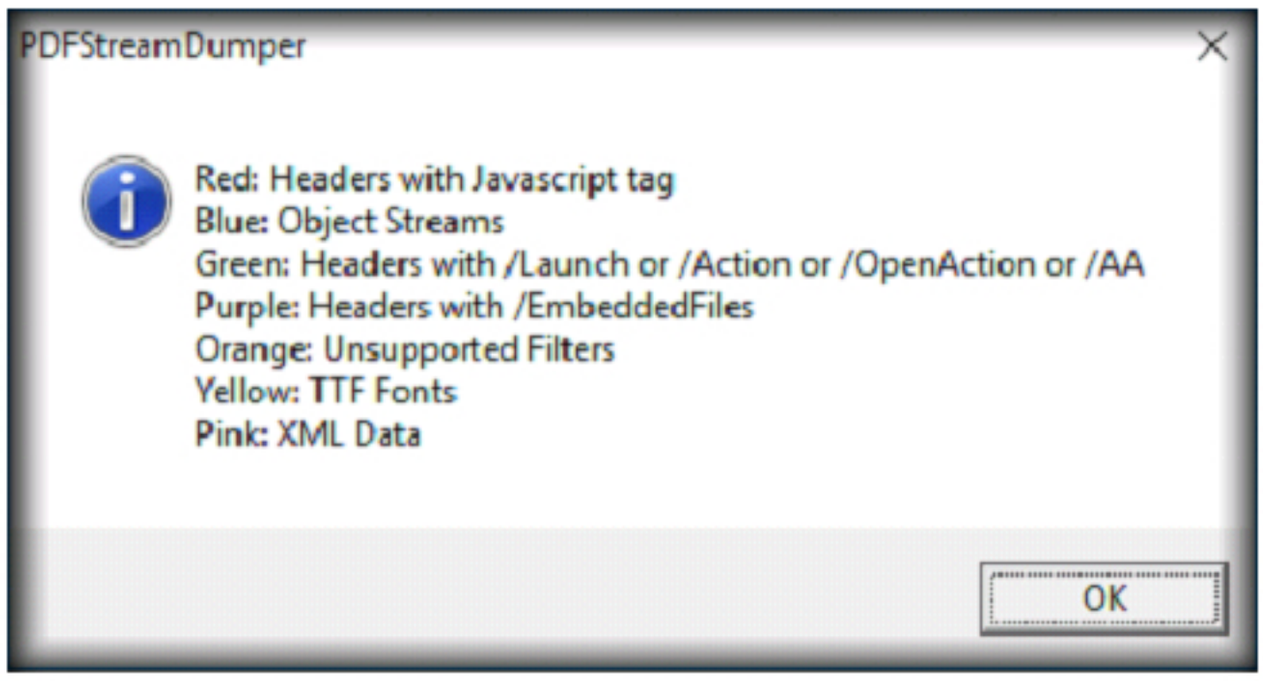


FIGURE 3.15: Screenshot representing the color specification

30. View the objects one by one to find their contents.
31. Analysis of the fifth object reveals that it contains a JavaScript header with the stream starting in the sixth object.

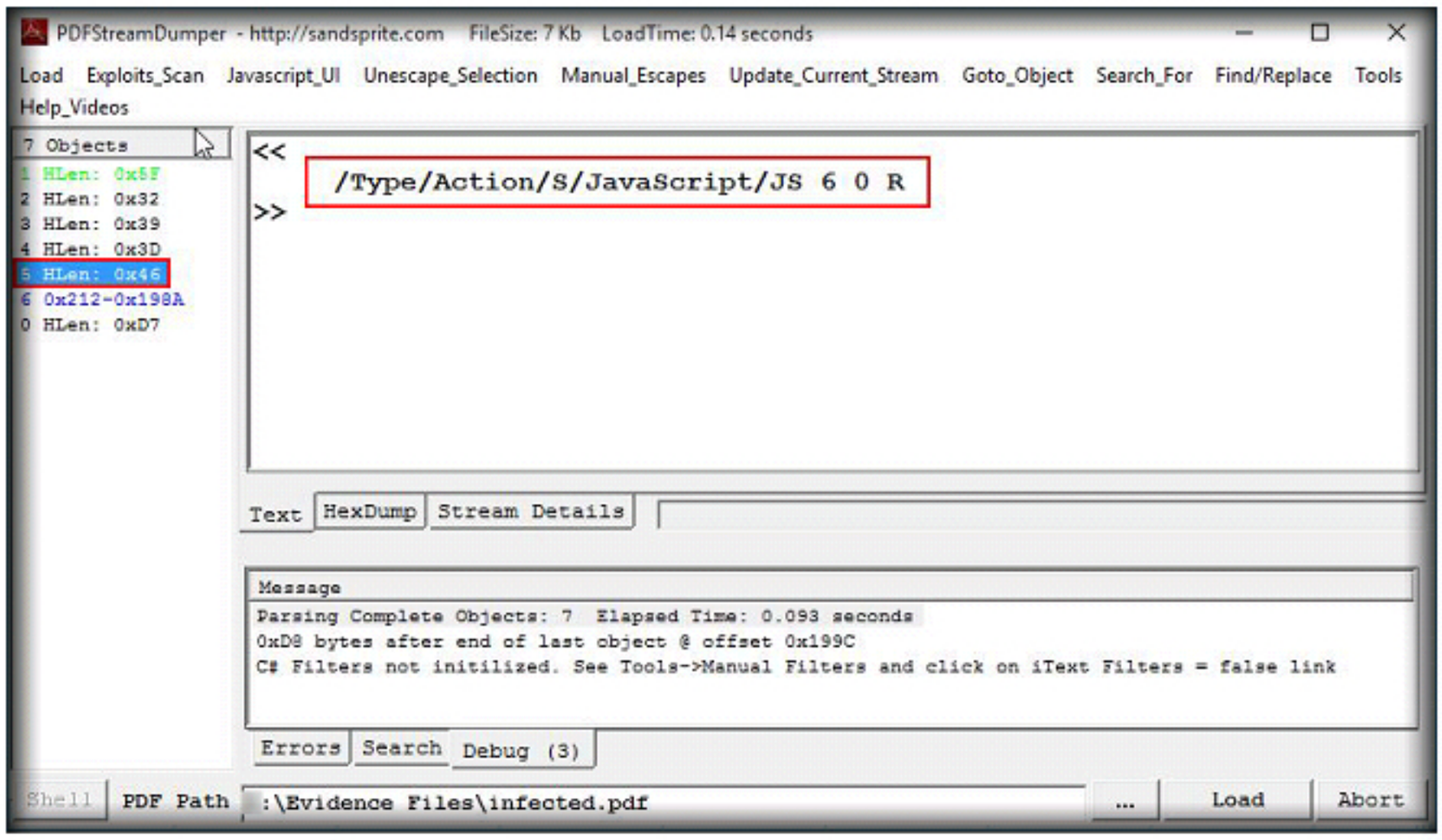


FIGURE 3.16: Analyzing the fifth object

32. To analyze the contents go to sixth stream by clicking on it from the left side object menu.

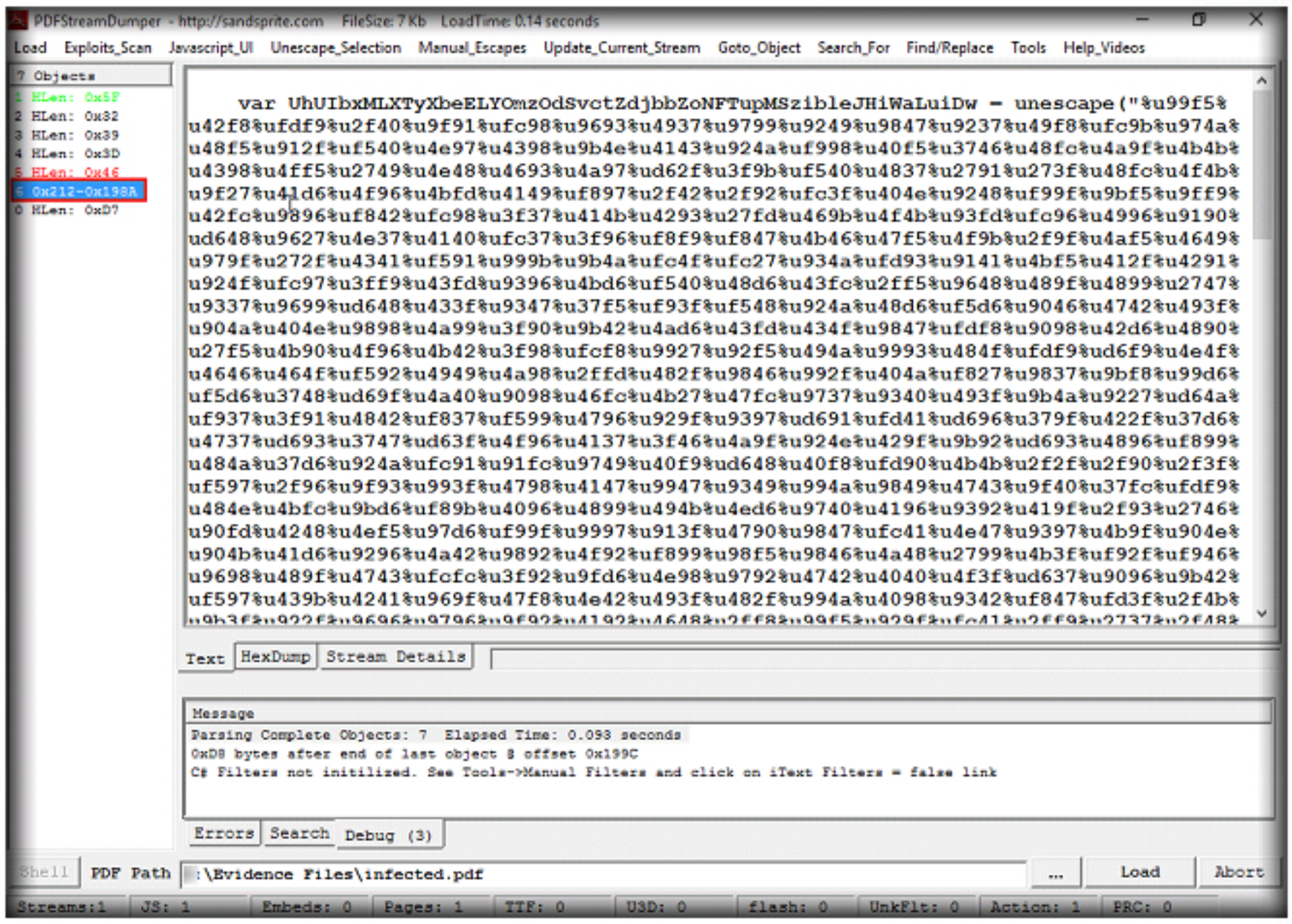


FIGURE 3.17: Analyzing the sixth stream

33. Scroll down to check for any readable strings from the Text of the suspicious object. One can notice that it uses the **util.printf()** command that can display the gathered system data.

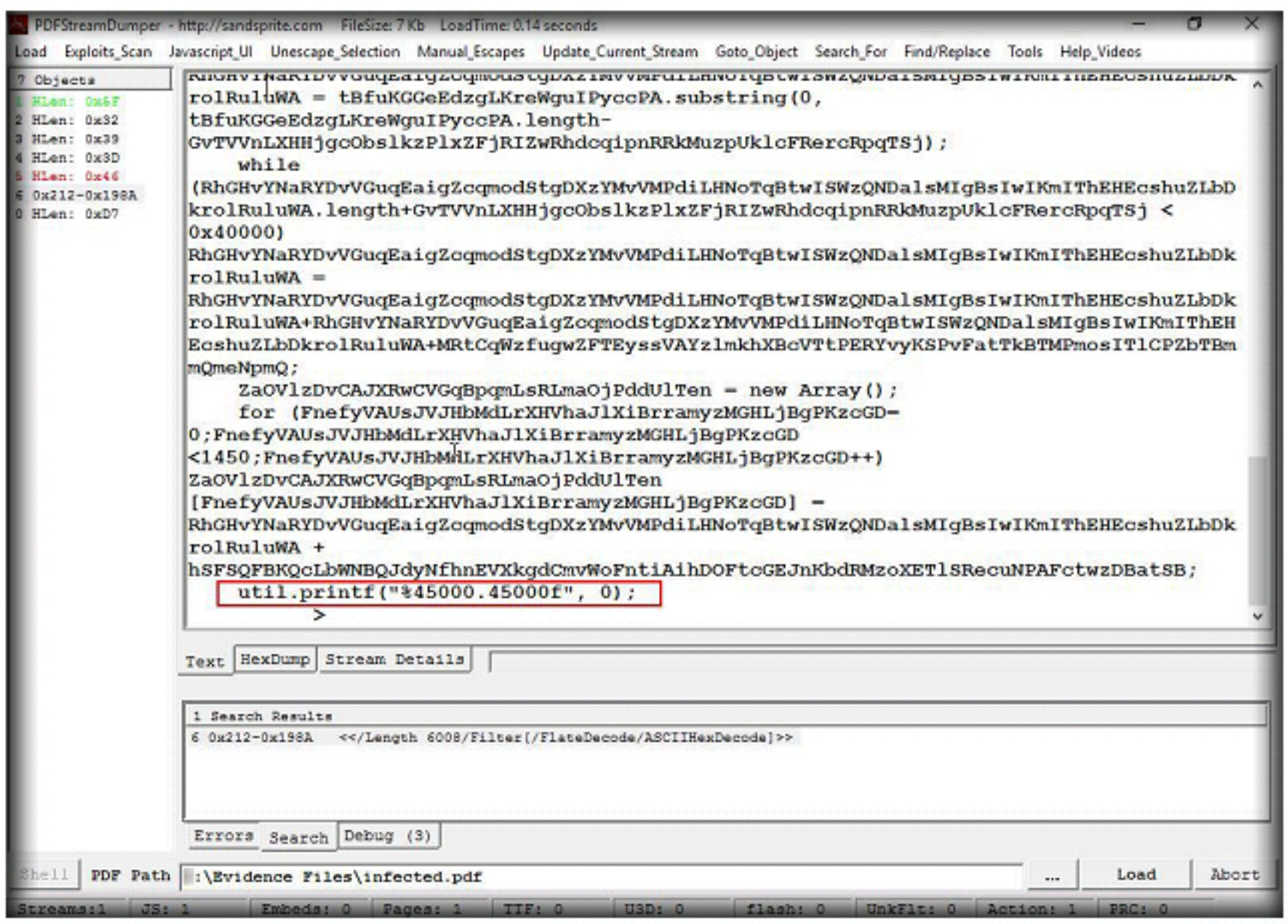


FIGURE 3.18: Checking for any readable strings from the Text of the suspicious object

34. For gathering further details on other exploits, click the **Exploits_Scan** option from the toolbar.

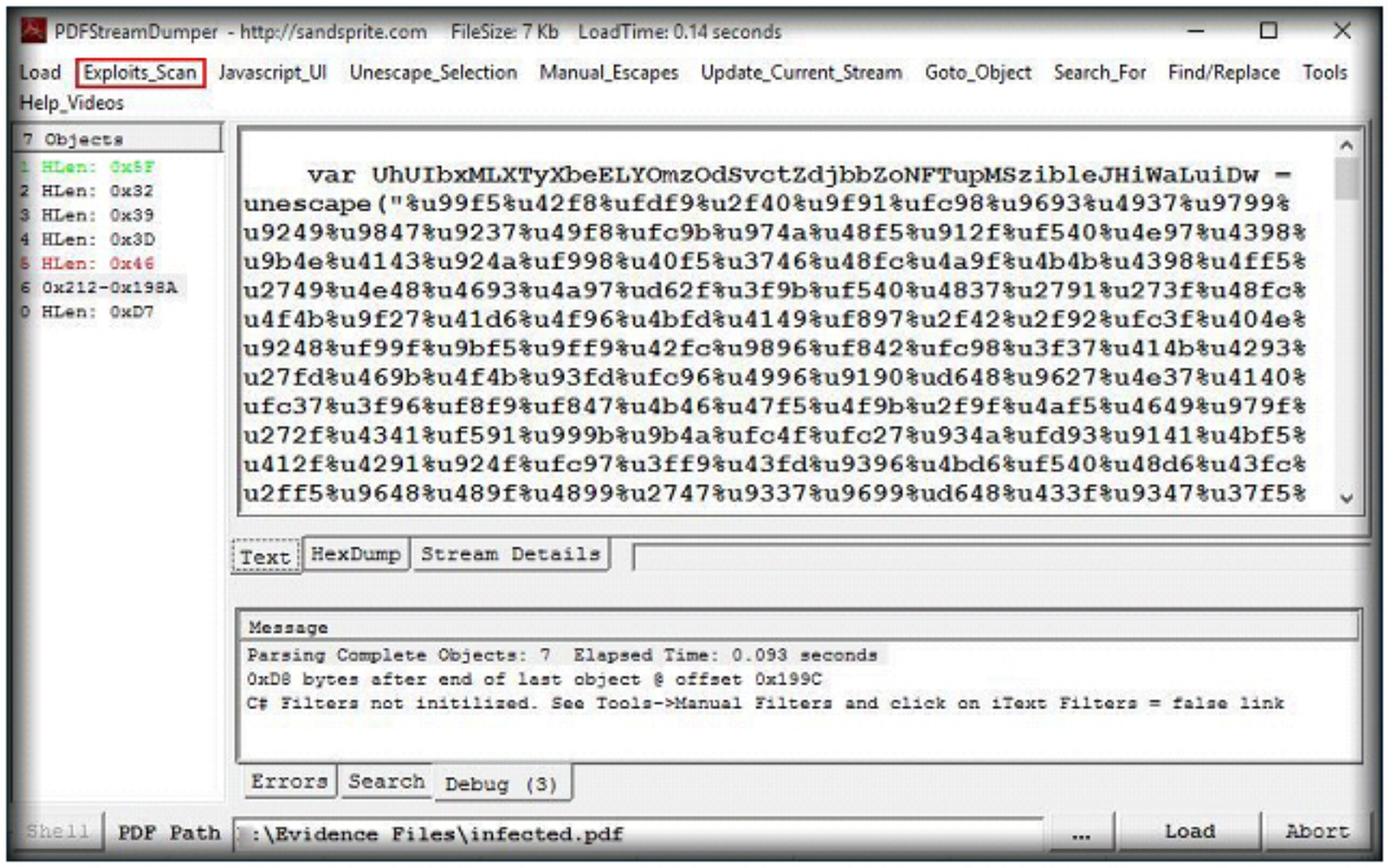


FIGURE 3.19: Screenshot representing Exploits_Scan option

35. The tool will scan the object for any available exploits and display them in a new window. The result includes the type of exploit used, date of embedding the exploit into the PDF and the serial number of the stream containing the exploit. The exploit also marks the same util.printf command as malicious.

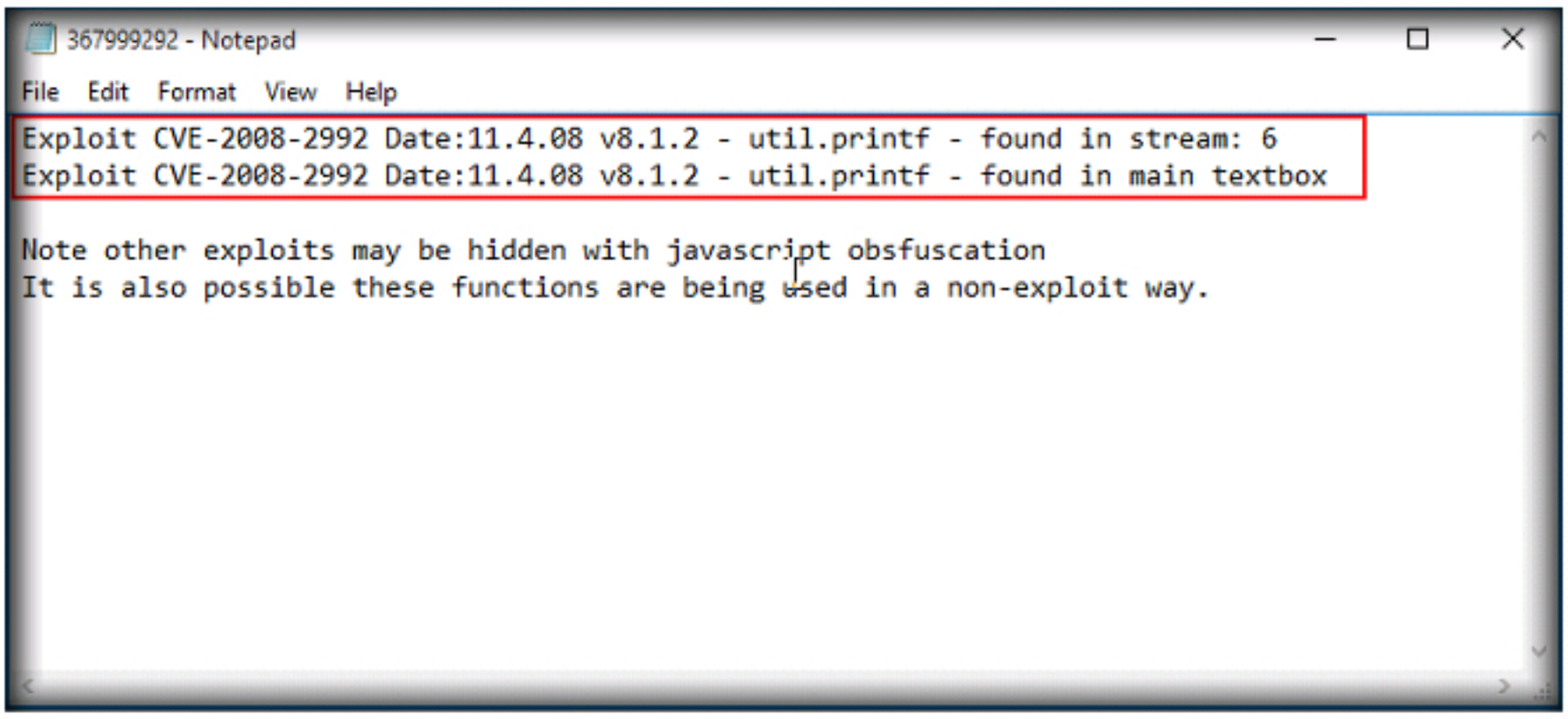


FIGURE 3.20: Viewing further details from Exploits_Scan option

36. To be sure, check for the vulnerability across online security databases such as **https://vulners.com**. Go to the website by typing the URL in a browser. Give the vulnerability name in the search box provided and click on the button with Arrow symbol on it.

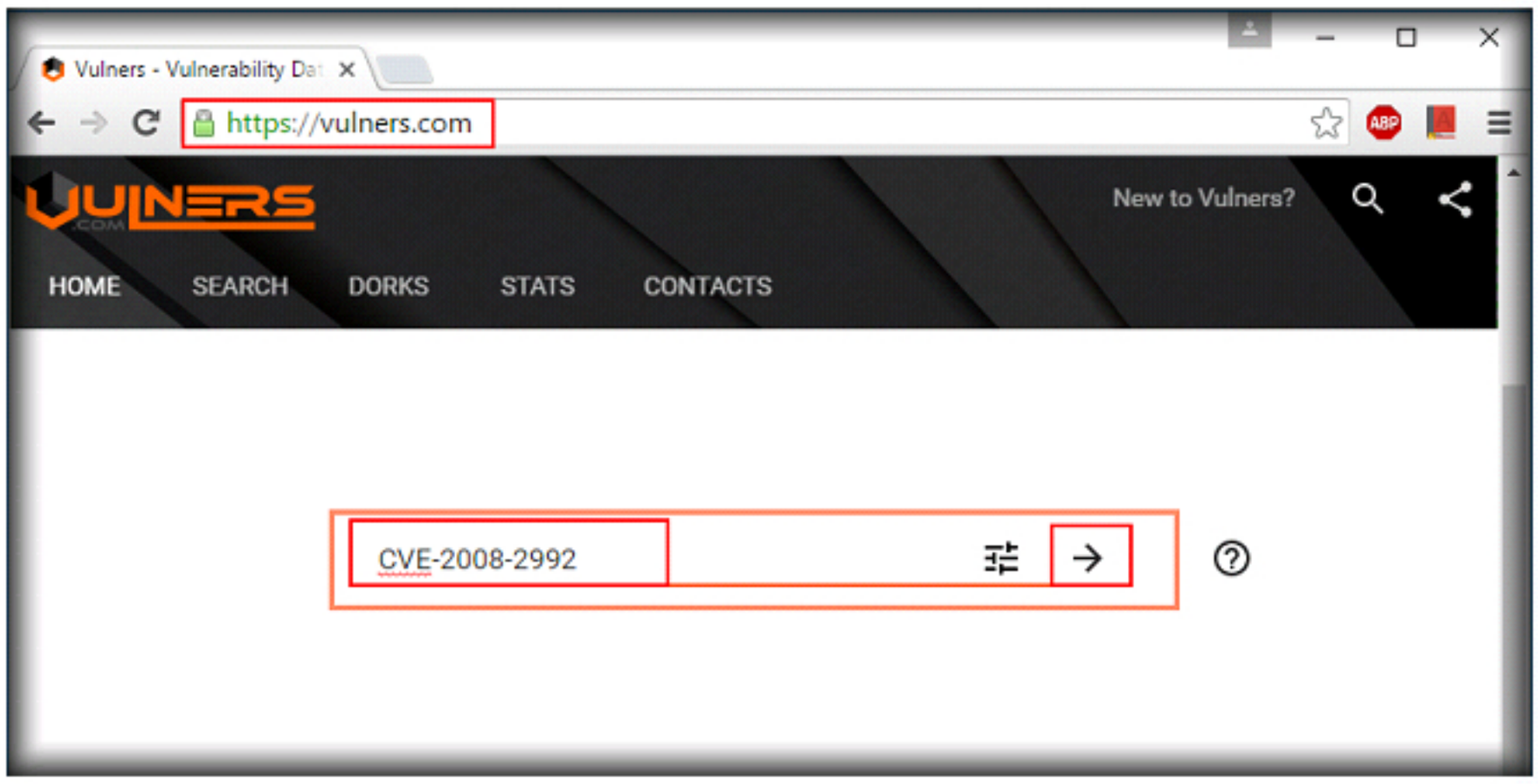


FIGURE 3.21: Checking for vulnerability across an online security database

37. Click on any of the results to have detailed information about the vulnerability.

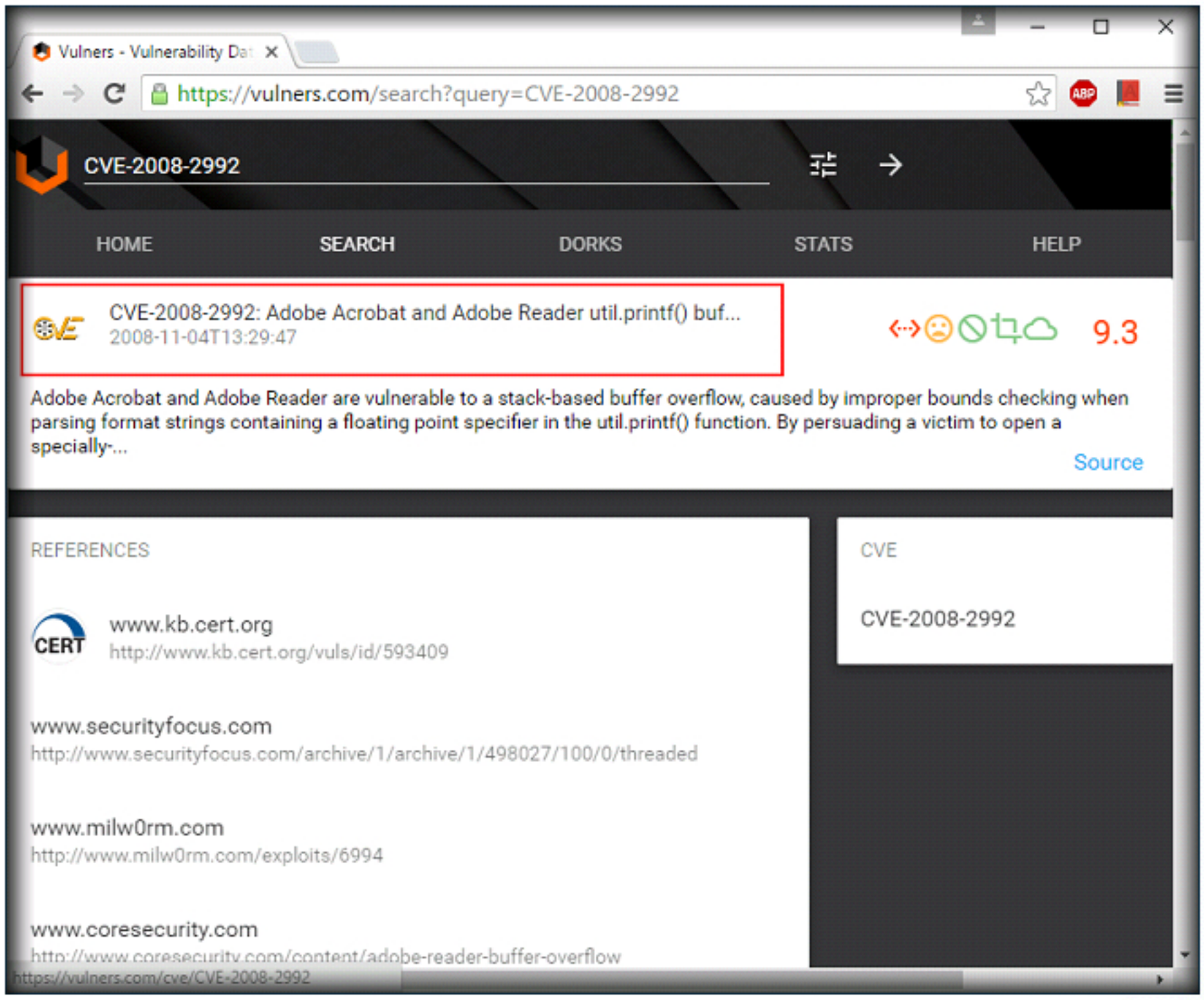


FIGURE 3.22: Viewing the detailed information about the vulnerability

38. Analyze the JavaScript in the object by clicking on the **Javascript_UI** option from the toolbar.

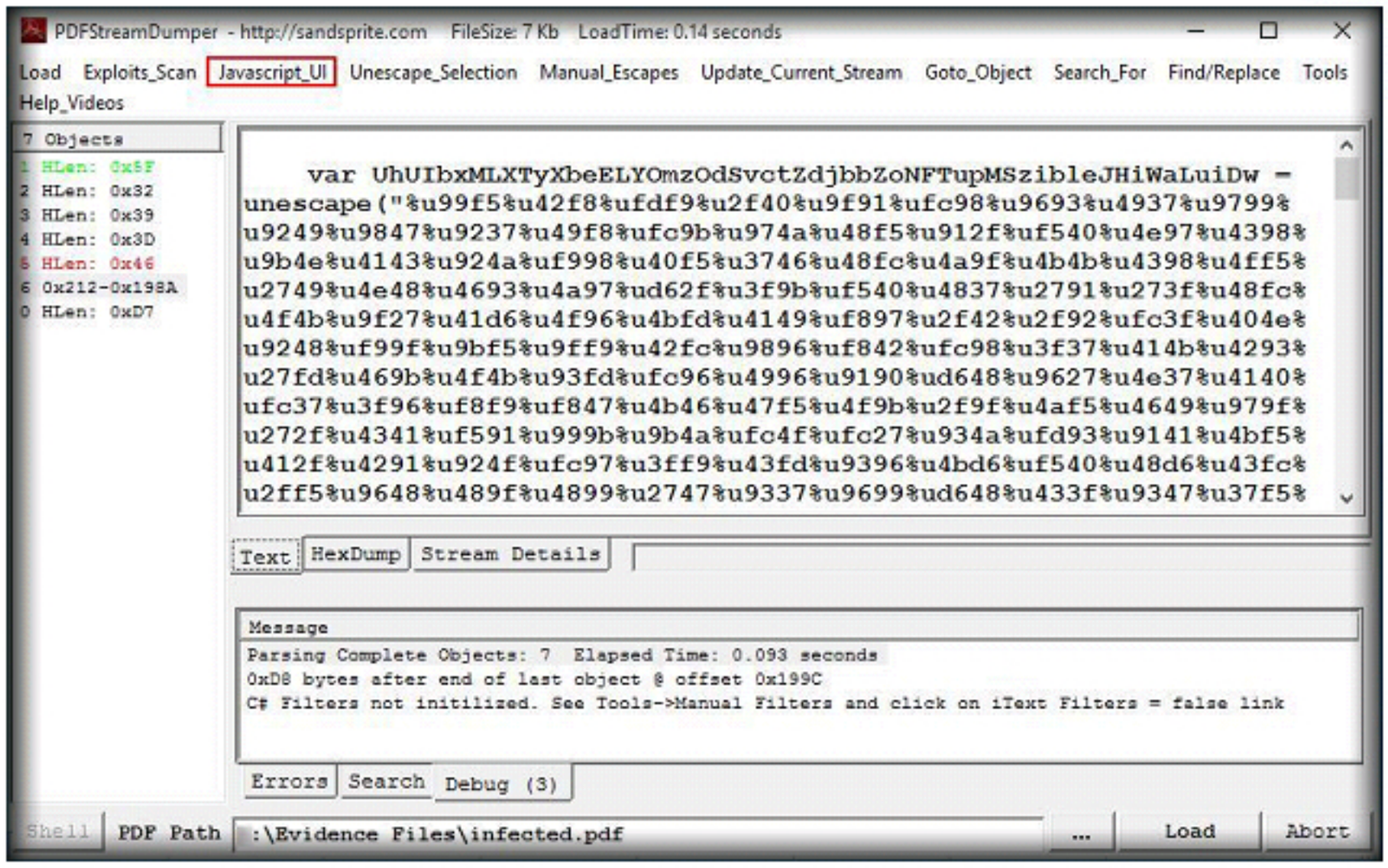


FIGURE 3.23: Screenshot showing the Javascript_UI option in the toolbar

39. The tool will display the Javascript_UI in another window. Click **<-- to clipboard** link present under the toolbar to save a copy of the scripts as analysis steps tend to modify the original script. The tool will display the saved script in the left side menu.

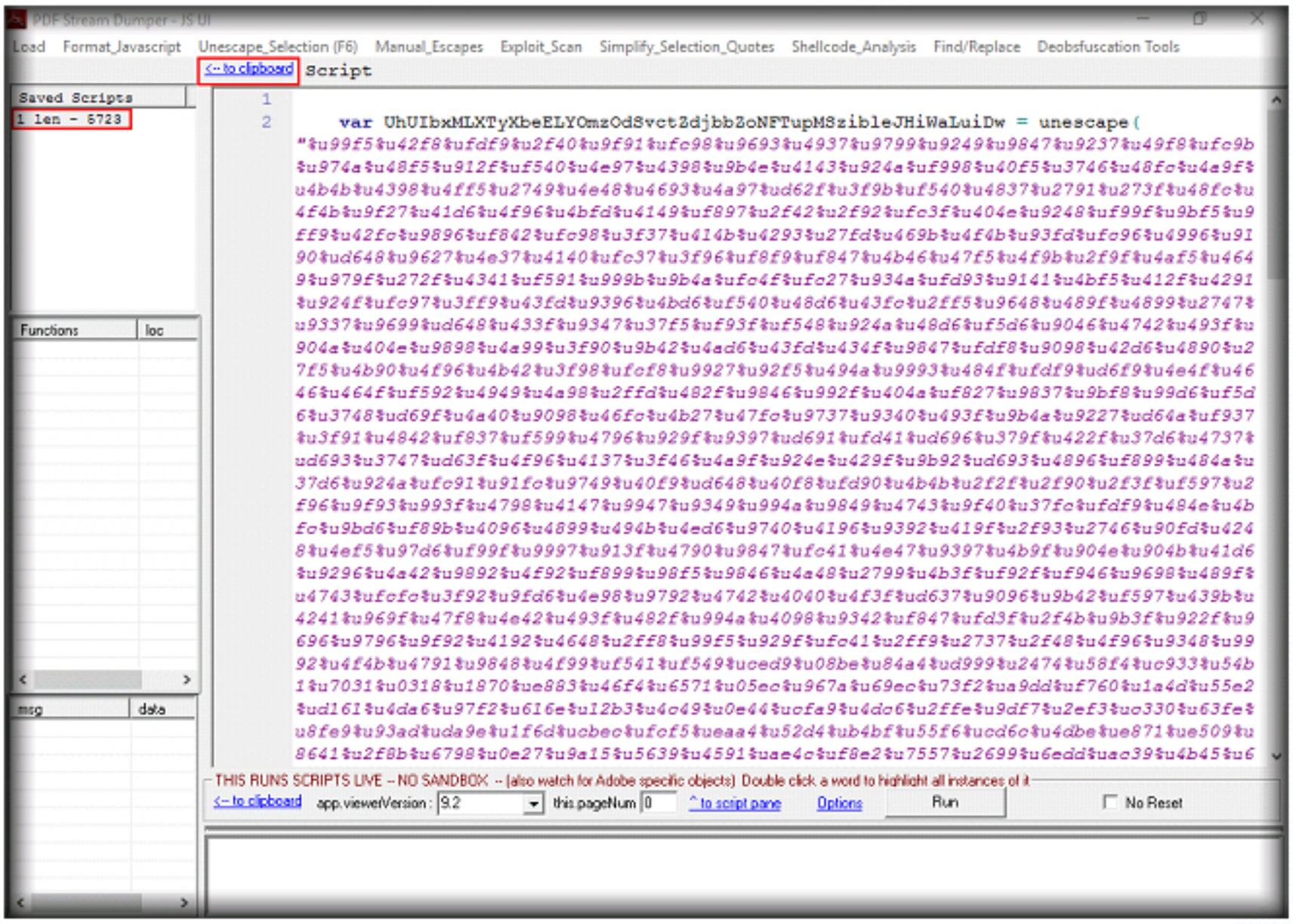


FIGURE 3.24: Viewing the results of the Javascript_UI option

40. The Javascript_UI will help the reader differentiate the commands and codes by offering color specifications. Scroll to check the document for suspicious commands and strings.

FIGURE 3.25: Checking the document for suspicious commands and strings

41. The **util.printf( )** command is a vulnerability found in PDF files that can allow attackers to execute arbitrary code on the system, leading to buffer overflow attack.

Lab Analysis

Document all the results and reports gathered during the lab.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab

4

Scanning PDF files using online Resources

Malware Tracker is an online malware analysis tool that automates the static analysis of PDF, and common document formats.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Scanning PDFs and other documents for malicious scripts is challenging task due to daily advances in technologies and a tremendous increase in the number of ways an attacker can use these documents to perform the attack. Investigators need to be up to date with all the databases for any recent malicious findings. Sites such as www.malwaretracker.com help in using online databases, which are up to date to analyze the attacks and identify the vulnerabilities used to perform them.

Lab Objectives

The objective of this lab is to help students scan the PDFs using www.malwaretracker.com and identify the missing links that can prove the documents to be malicious.

Lab Environment

To perform this lab, you need:

- A computer running Windows 2012 Server as the host machine
- Windows 10 running on a virtual machine

Lab Duration

Time: 5 Minutes

Overview of Malware Tracker

Malware Tracker provides malware analysis, forensics, and security solutions as well as helps to detect malicious document attachments and to automate the static analysis of PDF, and common documents.

Lab Tasks

TASK 1

Scan PDF Online

1. Launch a web browser, type the URL **https://www.malwaretracker.com** in the address bar, and press **Enter**.
2. Click the **scan PDF** button from options menu bar.

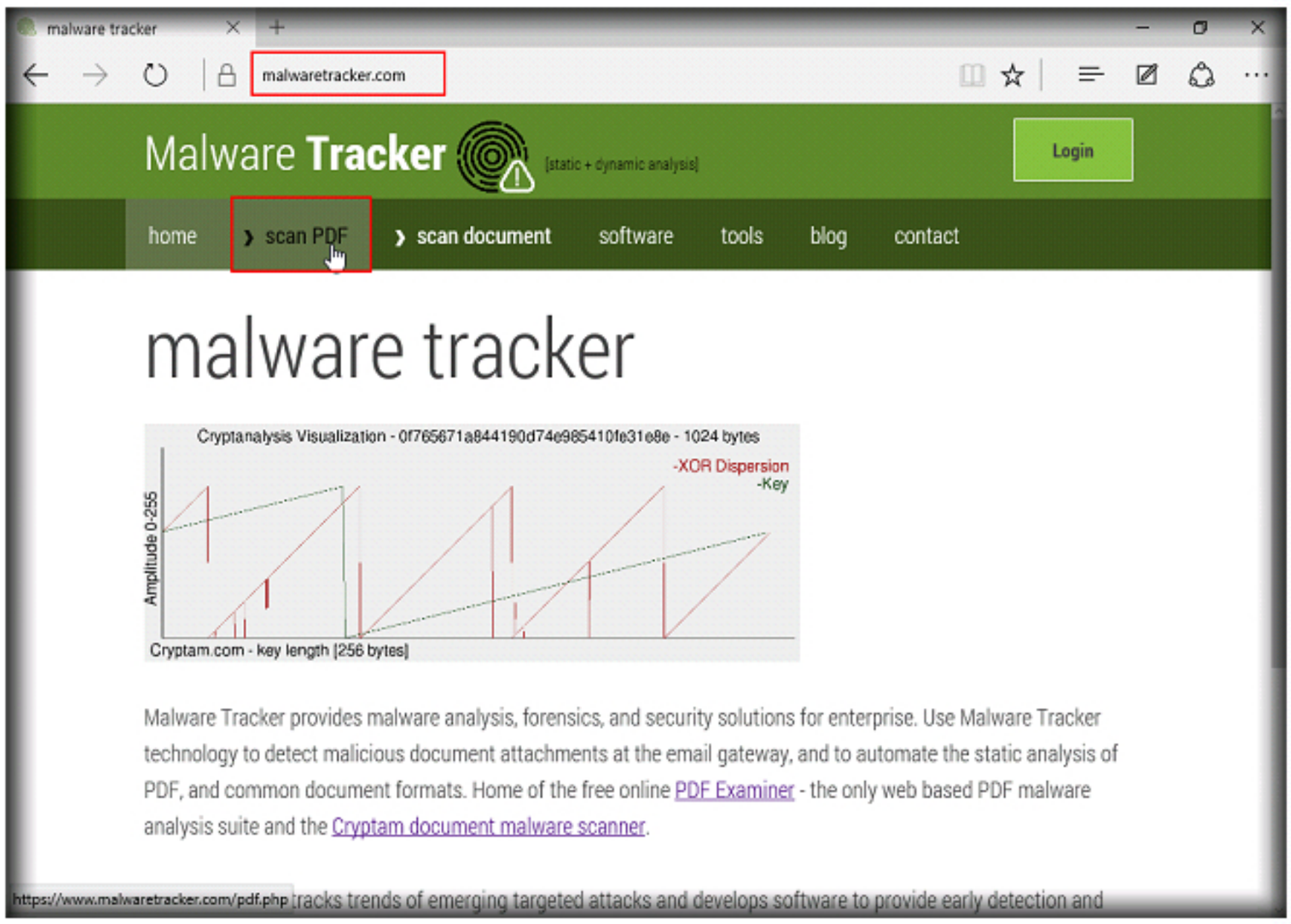


FIGURE 4.1: Malware Tracker home page

3. The website will navigate to a PDF examiner section. Click the **Browse...** button across the "**PDF sample:**" to upload the suspicious file.

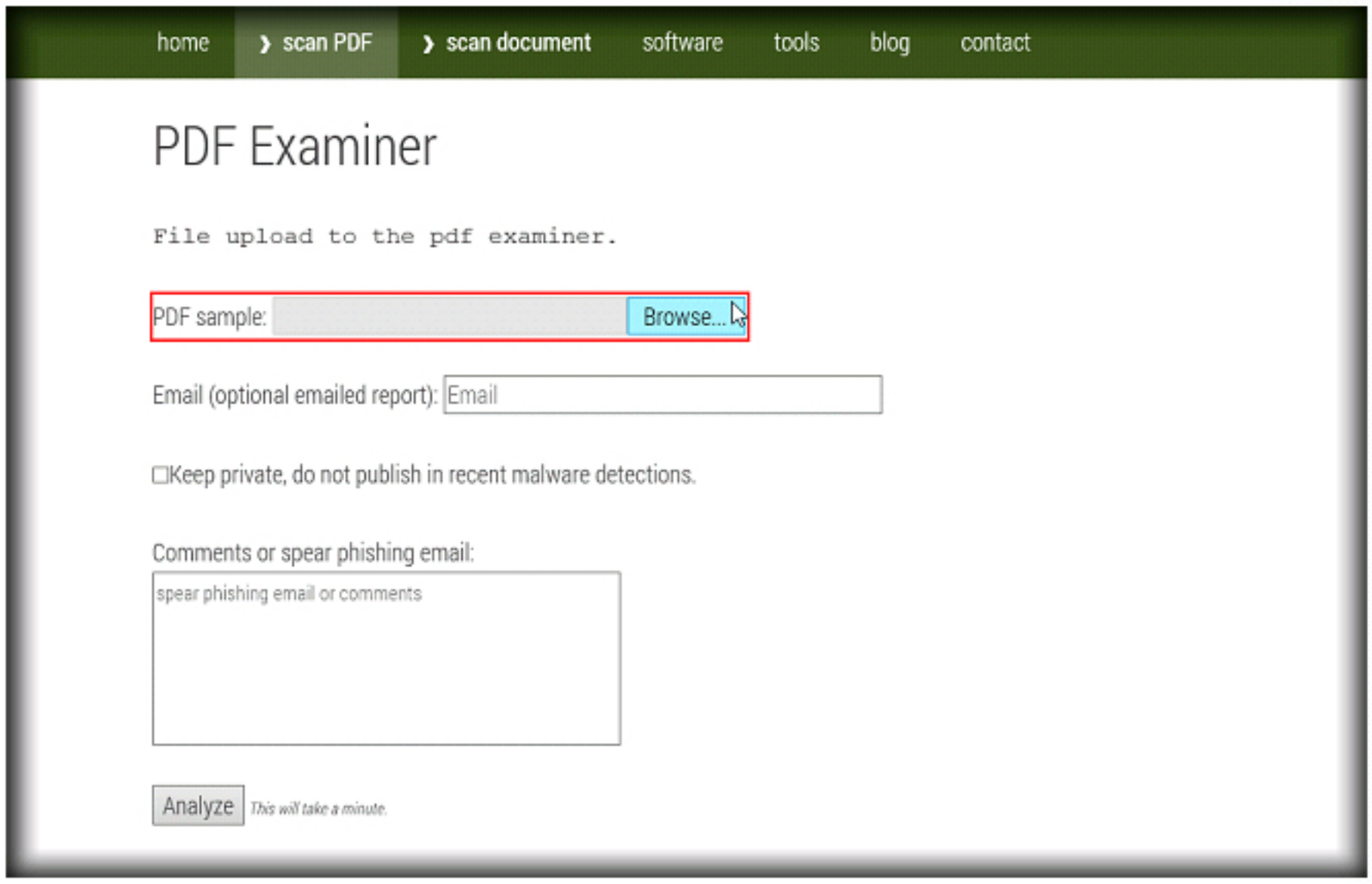


FIGURE 4.2: Scanning a PDF using Malware Tracker

4. Navigate to the file location, **C:\CHFI-Tools\Evidence Files**, select the **infected.pdf** file and click the **Open** button.

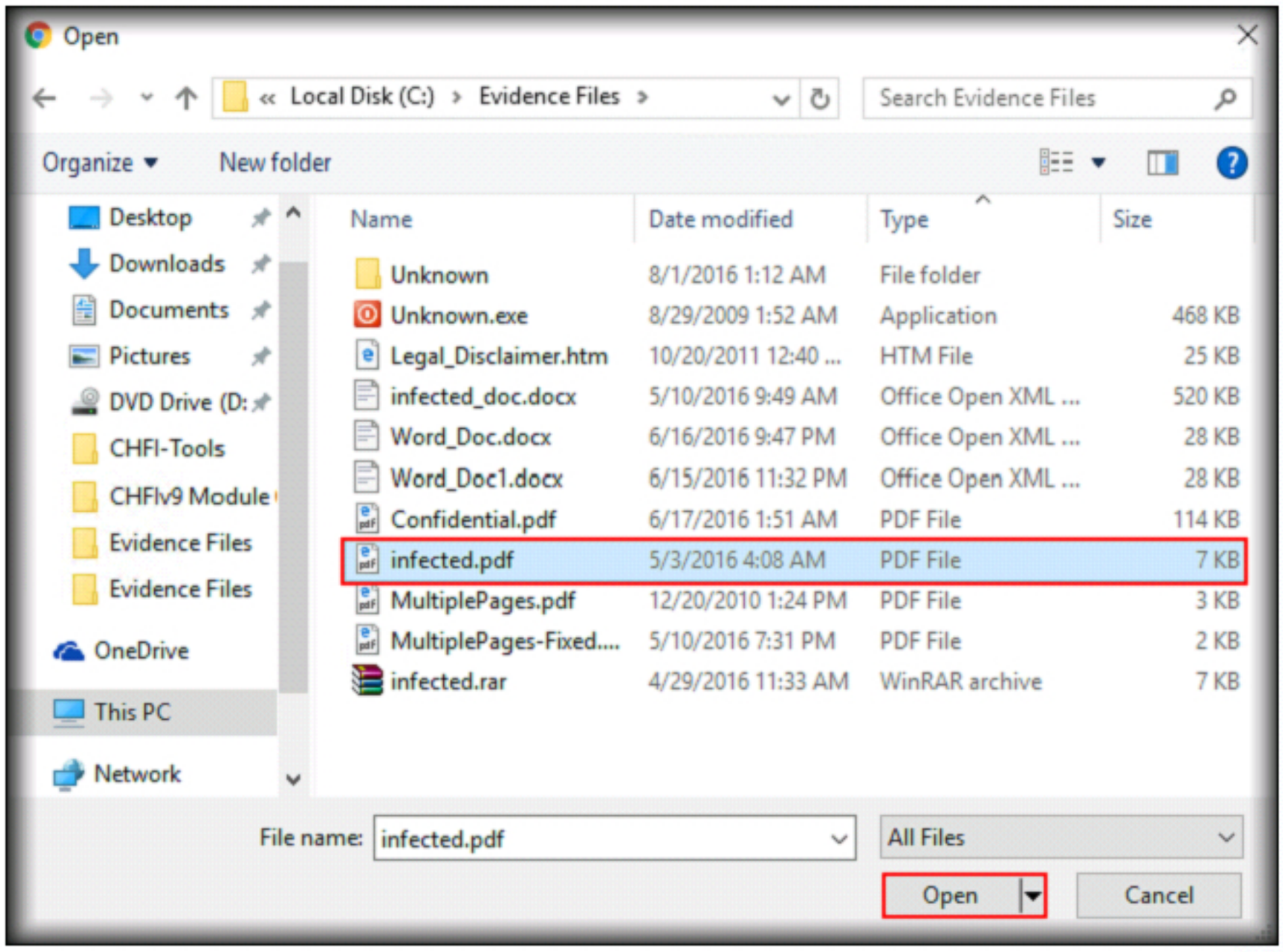


FIGURE 4.3: Navigating to the Evidence Files folder and selecting the suspicious Pdf file

5. The site displays a series of objects with different color specification. The **Object 6.0 @425** is red due to the presence of suspicious elements such as Javascript, unescape, substring and util.printf commands. These commands can perform functions such as modifying settings, gathering information, etc.

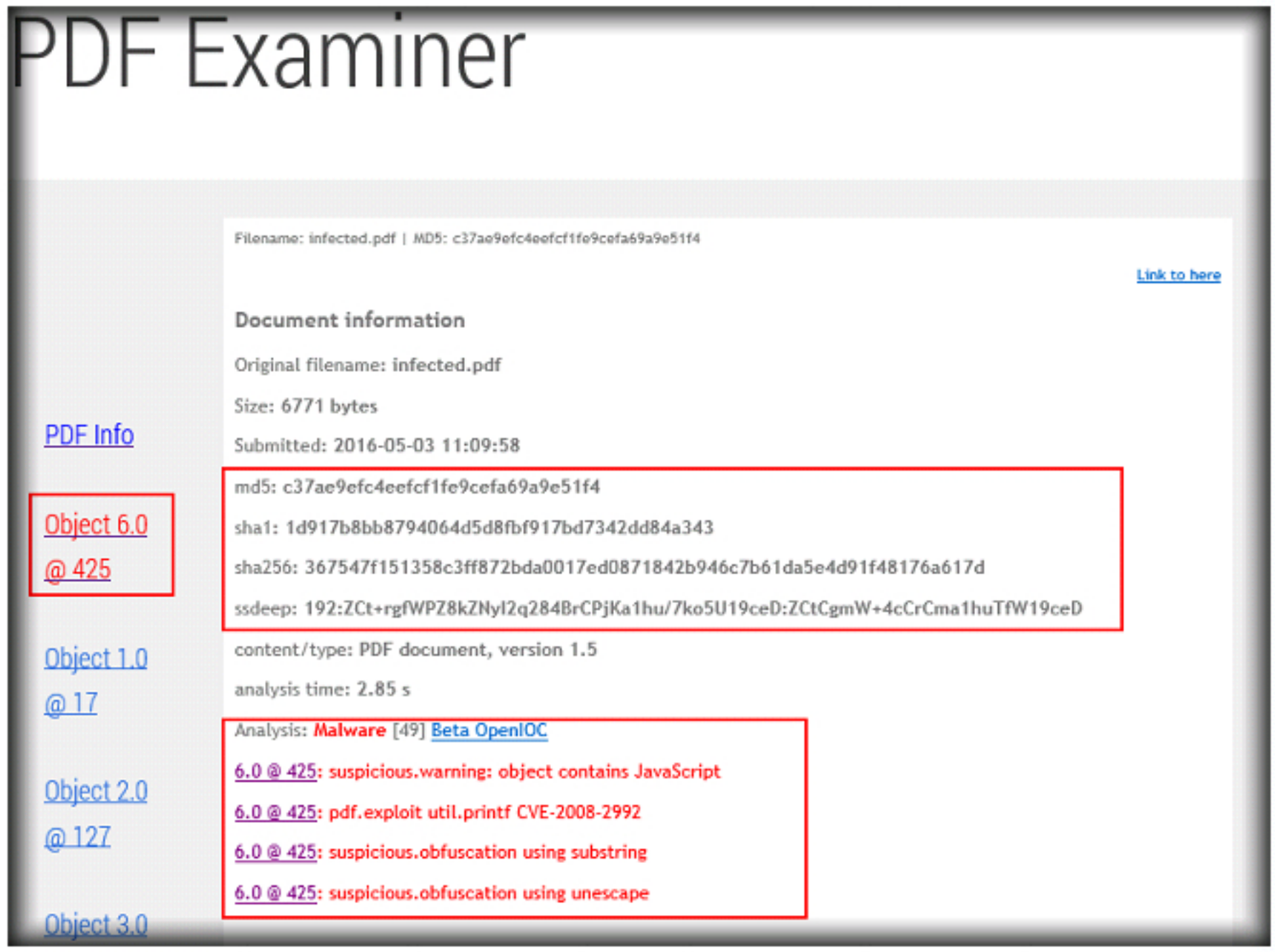


FIGURE 4.4: Examining objects having suspicious elements

6. Click the object name "**Object 6.0 @425**" from the left side column to view its details.

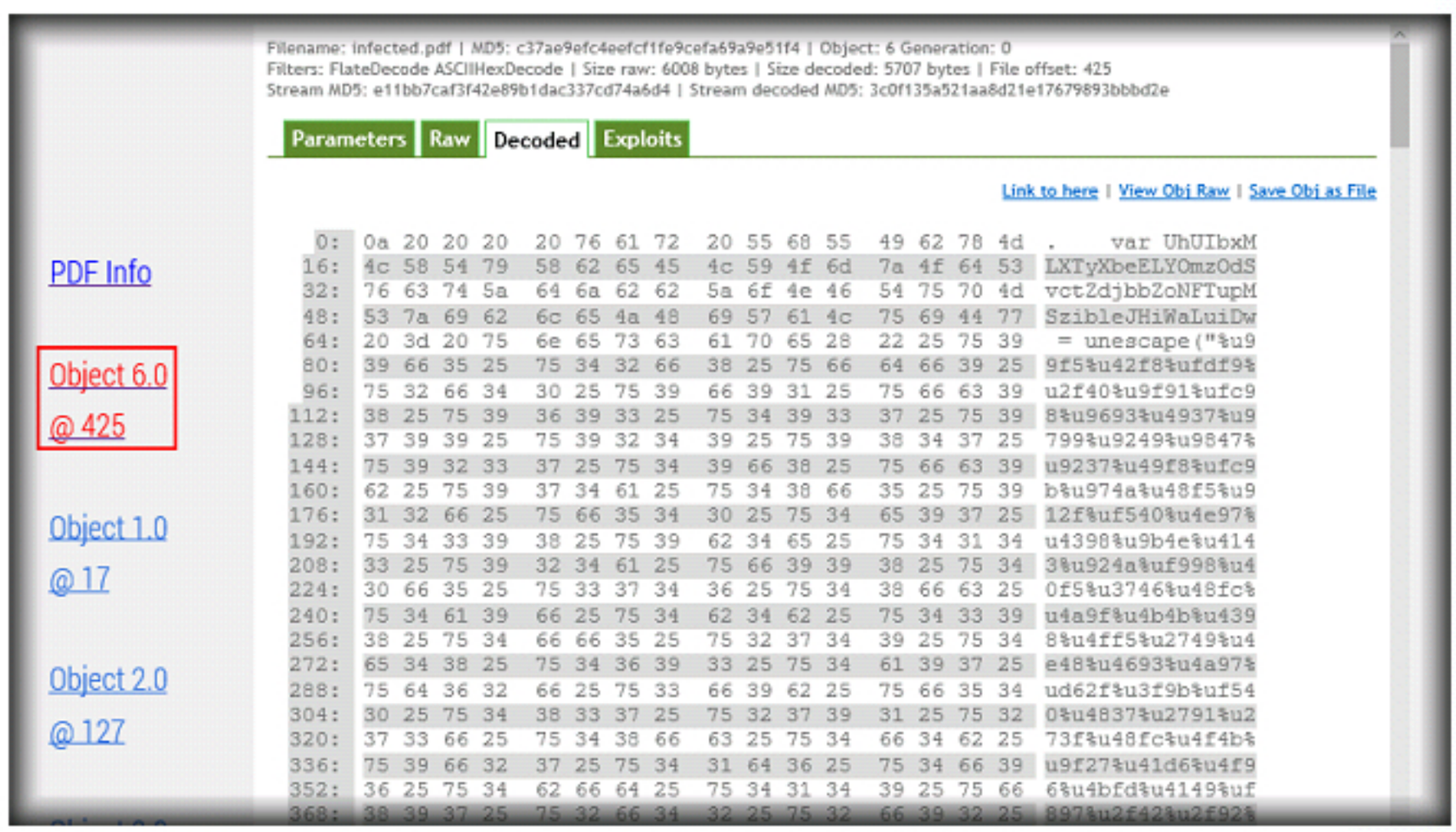


FIGURE 4.5: Viewing the complete details of the suspicious object

7. Click the **Exploits** tab to view the position of the suspicious strings. This will give a clear idea of content involved with these strings to perform malicious activities.

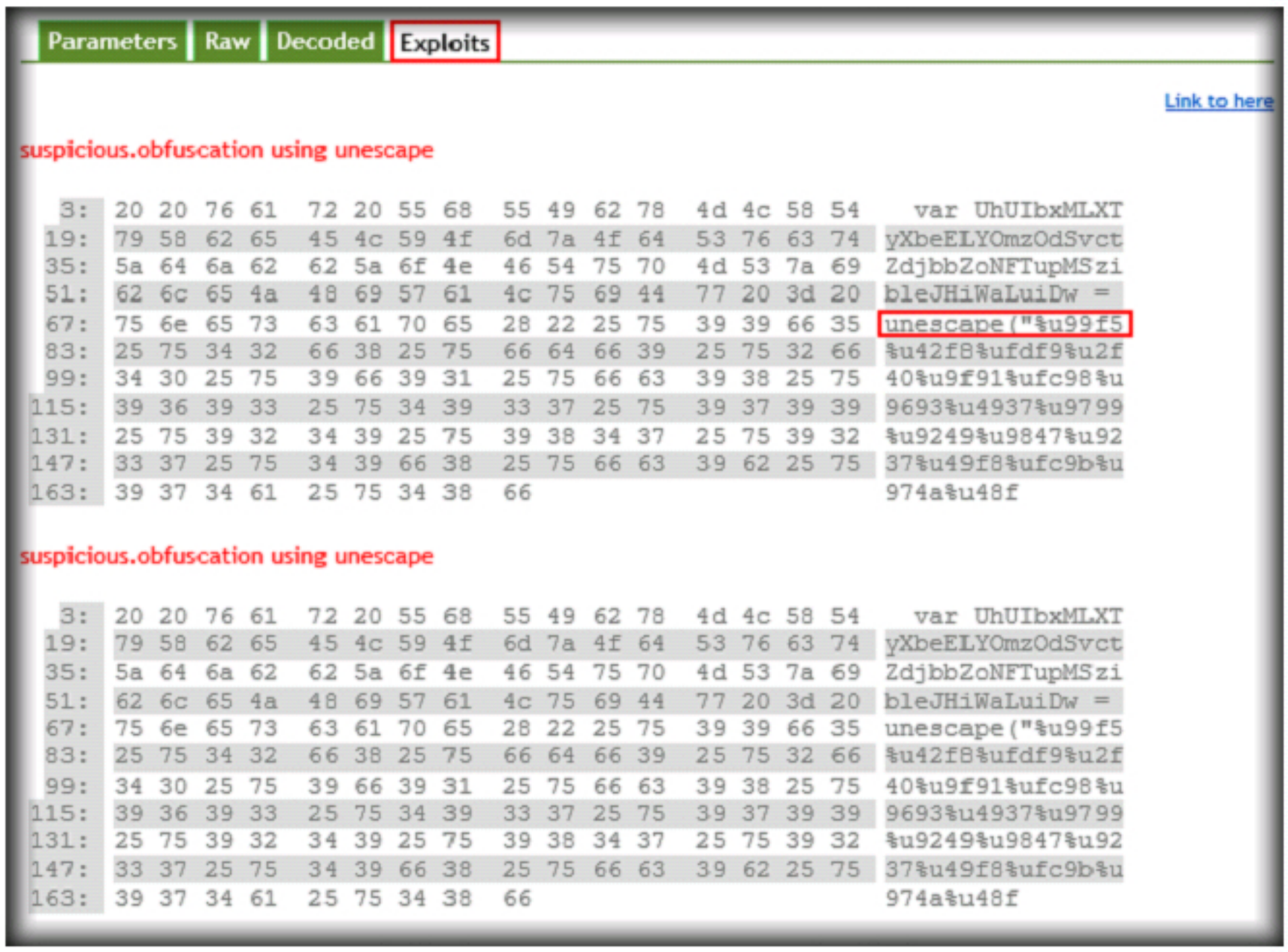


FIGURE 4.6: Viewing the position of suspicious strings using Exploits tab

Lab Analysis

Document all the findings you obtained during this lab.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☐ iLabs

Lab

5

Scan suspicious MS-office file for Malice

MS-Office files such as word documents and exel sheets often act as carriers in spam campaigns, as attackers try to transmit packed executable masqueraded as a document.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Attackers use MS-Office documents to deliver various malicious executables, to compromise system security and install an unknown program. Investigators need to be aware of such acts and be up to date with all the vulnerabilities existing with such document formats. The lab will help users find if the document stored in a compromised system has any suspicious strings and codes that could have helped attackers.

Lab Objectives

The objective of this lab is to help students analyze the MS-Office document files and extract the hidden malicious content using various online and offline tools.

Lab Environment

To perform this lab, you need:

- A computer running Windows 10 Virtual Machine.
- A web browser with Internet access.
- Administrative privileges to run the document analysis tool.

Lab Duration

Time: 30 Minutes

Lab Tasks

TASK 1

Analyze Malicious Documents

1. In this lab, your primary task is to upload the suspicious PDF file to online virus databases, such as www.virustotal.com and https://malwr.com, and check if these databases contain signatures of virus in the file which you are going to upload.

2. Login to **Windows 10** machine, launch a web browser and enter the URL https://www.virustotal.com. VirusTotal homepage appears, click the **Choose File** button to upload the suspicious file.

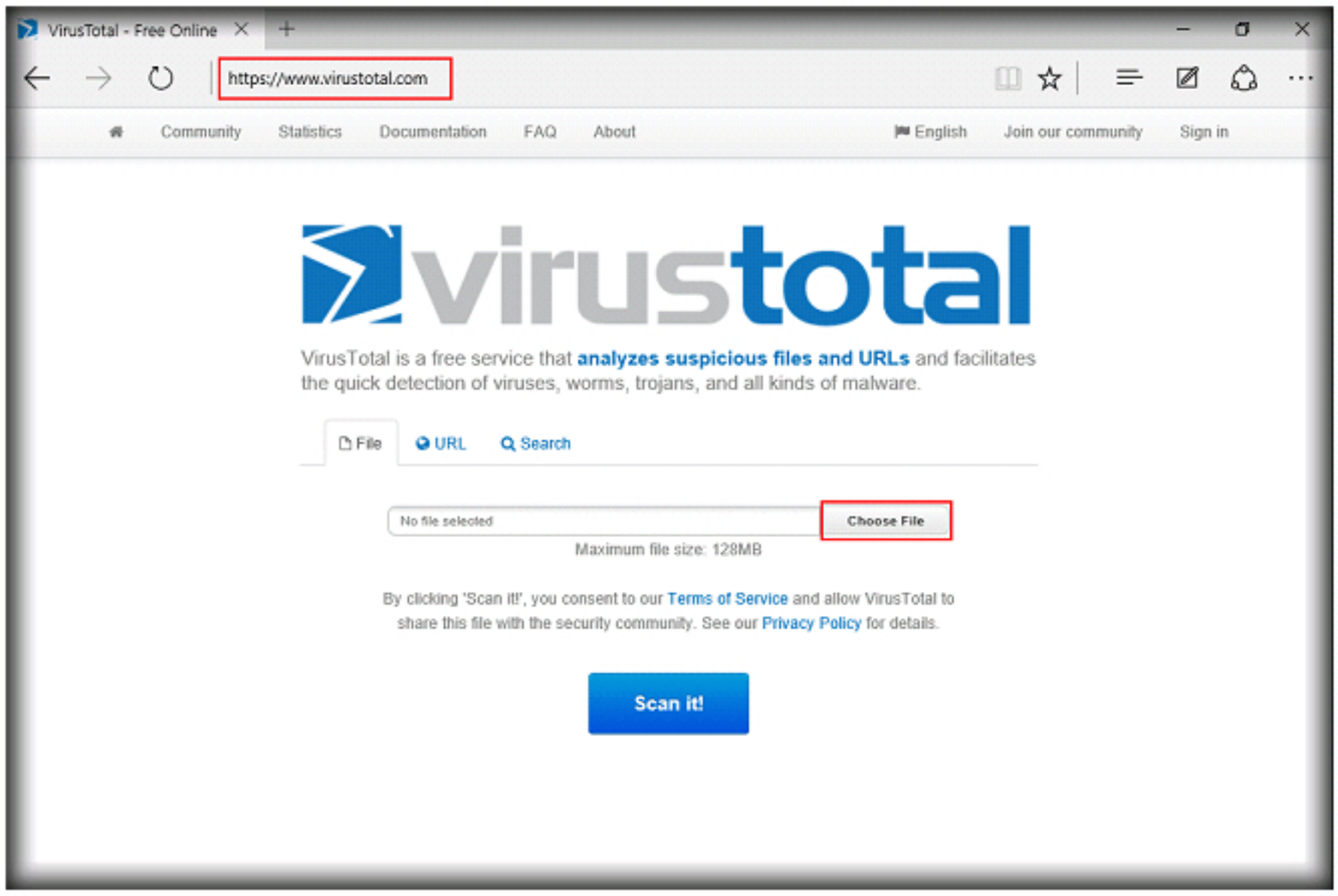


FIGURE 5.1: VirusTotal home page

3. **Open** window appears, Navigate to the file location: **Z:\Evidence Files**, select the file **infected_doc.docx** and click **Open**.

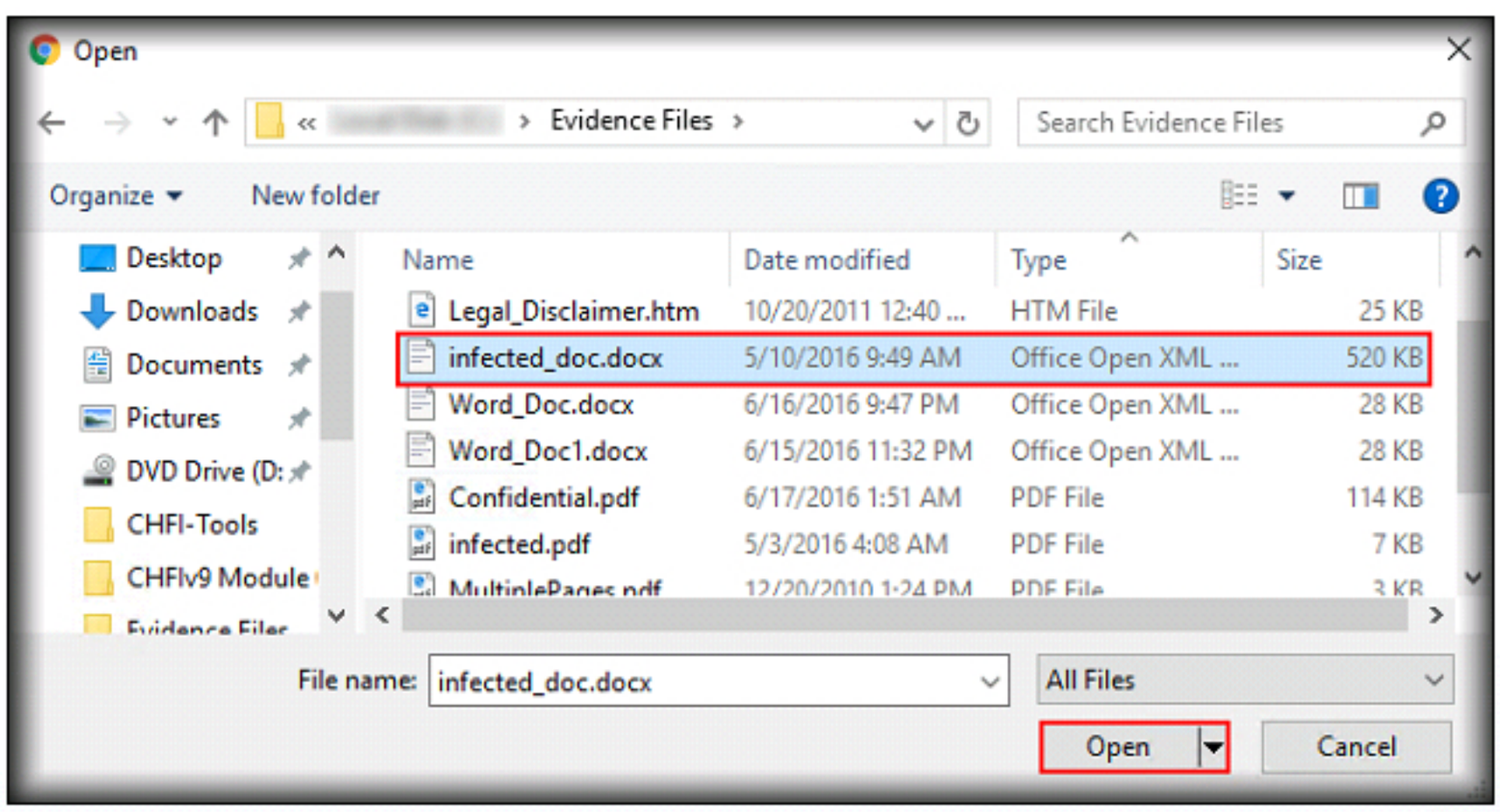


FIGURE 5.2: Navigating to the Evidence Files folder and selecting the suspicious file

4. Click **Scan it!**.

FIGURE 5.3: Scanning the suspicious file

5. The website will analyze the file and display the result. The website will cross-check databases of various antivirus programs for similar contents and reveal the result.
6. The result discloses the 56 available databases, 35 have confirmed the file to contain malice and also presented a list of result from different antivirus databases.
7. The result also displays the type of malware the file acts such as exploit, Trojan, Downloader, etc.

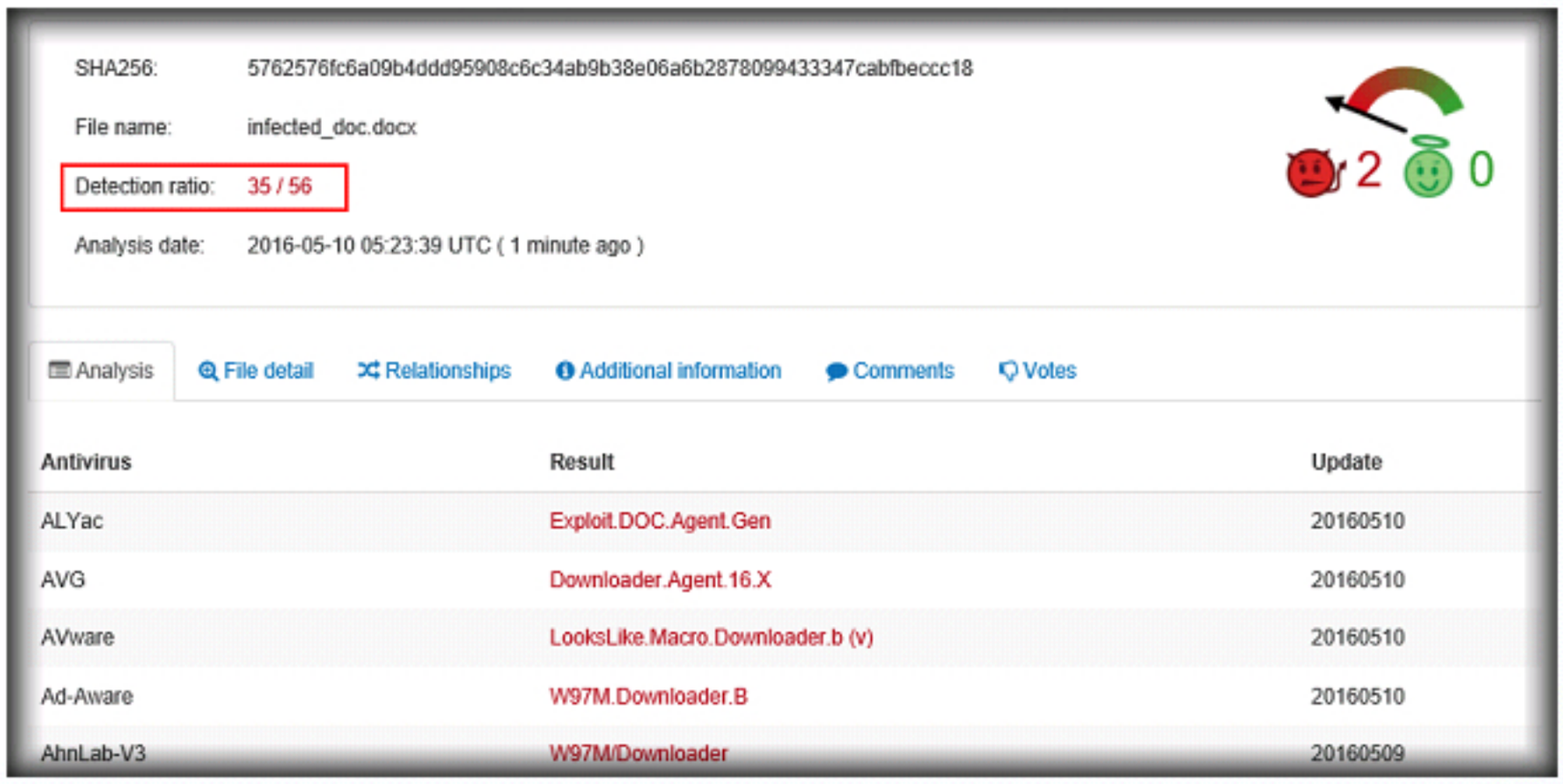


FIGURE 5.4: Viewing the results after completion of scanning

8. Click the **File detail** tab to check other information about the file. The commonly abused properties section reveals information about the content, actions, variables and commands in the file along with their impact on the system.
9. In the **Summary** section, the website shows the author, time of creation, file name, last print time, etc. Close observation reveals that the time of printing is in the year 2006, while the creation time is in 2009, which is contradictory. The observation divulges that the malicious content in the file is manipulating the date and time settings of a system.
10. The summary also reveals that the author had used the Spanish language during the creation of the file.

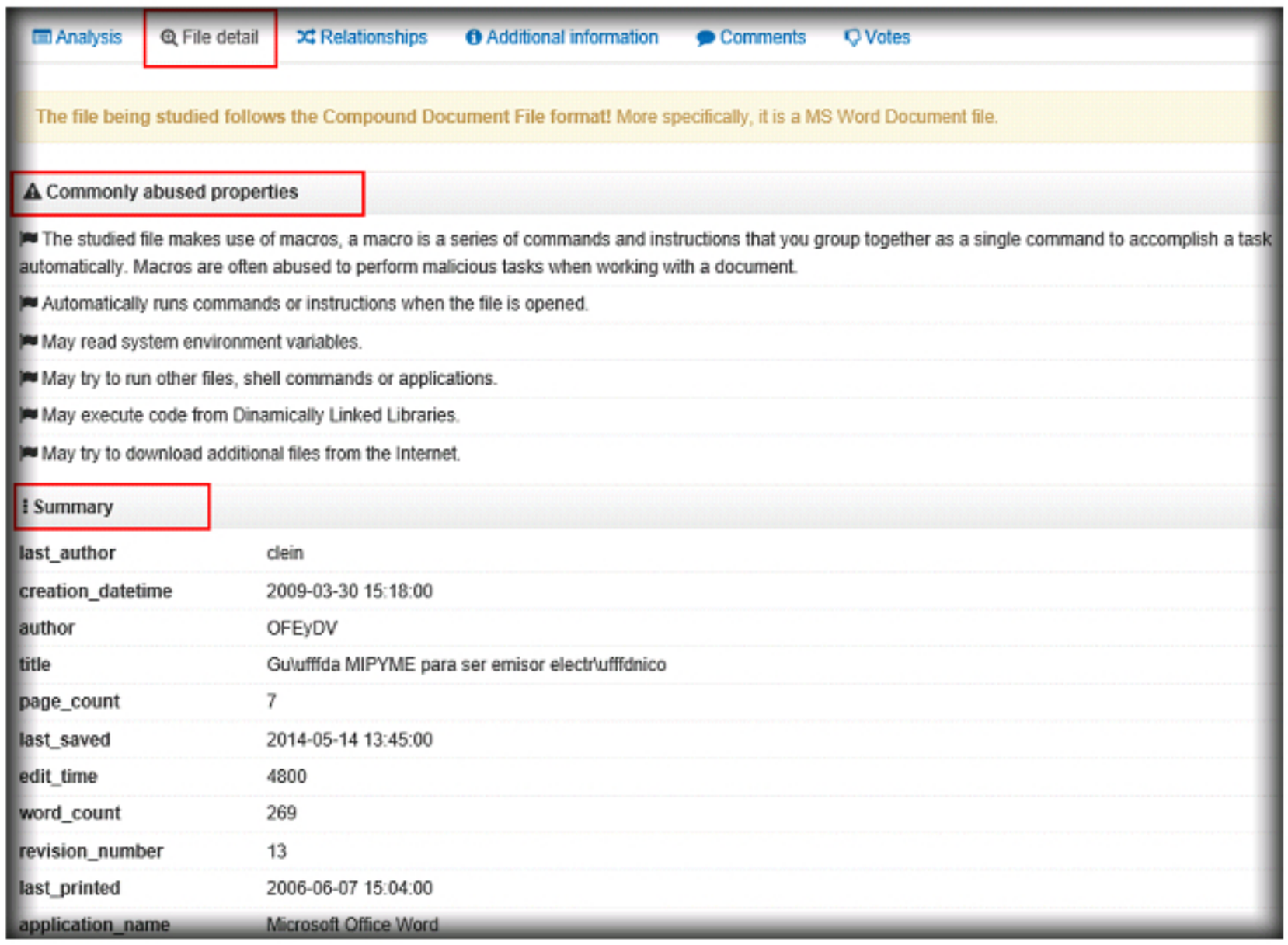


FIGURE 5.5: Examining the results from the File detail tab

11. Scroll down to the **OLE Streams** section, which displays file contents in rows. Click on a particular row to see further details such as type_literal, size, name and sid.

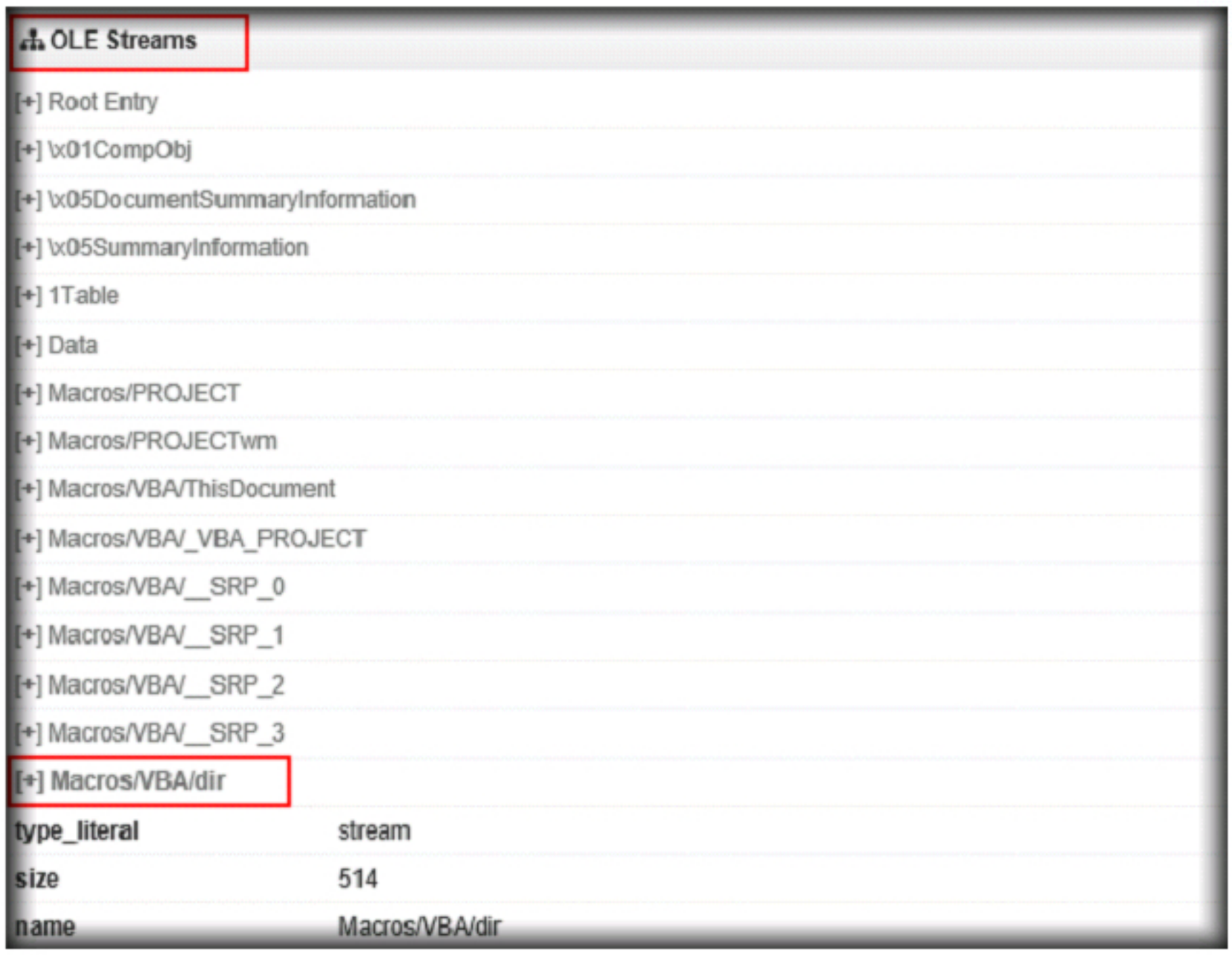


FIGURE 5.6: Examining the OLE Streams section

12. Scroll down to the **Macros and VBA code streams** section to gather information about the malicious content in the file. The section displays the tags that make it malicious.

13. Click the **ThisDocument.cls** link to see the coding available in the file. The coding includes Auto-open actions and hyperlinks that seem to fetch some executable files.

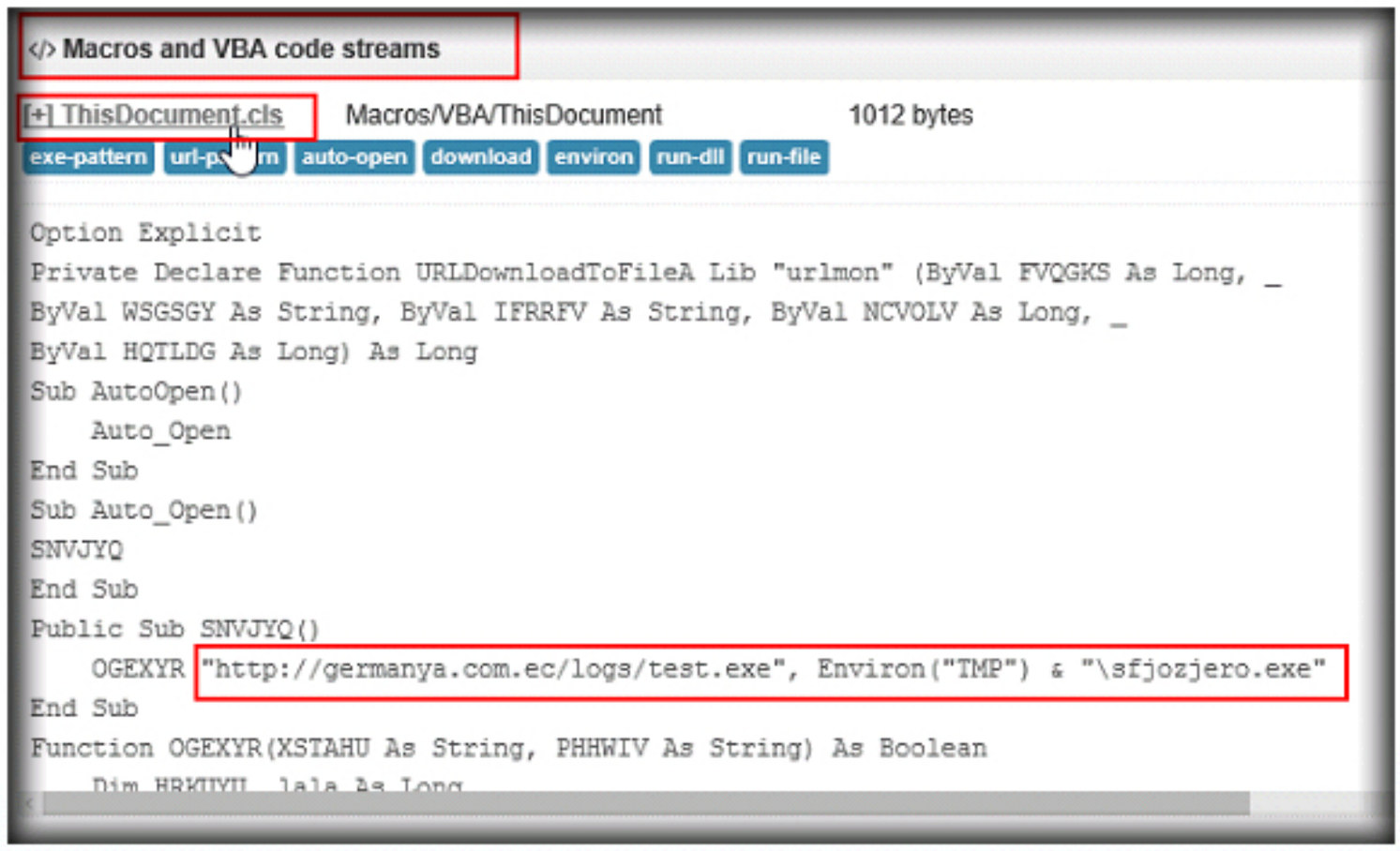


FIGURE 5.7: Examining the results from the Macros and VBA code streams section

14. Click the **Additional information** tab to view additional information of the file. The section displays different hash values of file such as MD5, SHA1, SHA256, etc. It also displays the tags of the contents that make file suspicious.
15. The VirusTotal metadata shows the first submission date and different file names which other have used to search the same file.

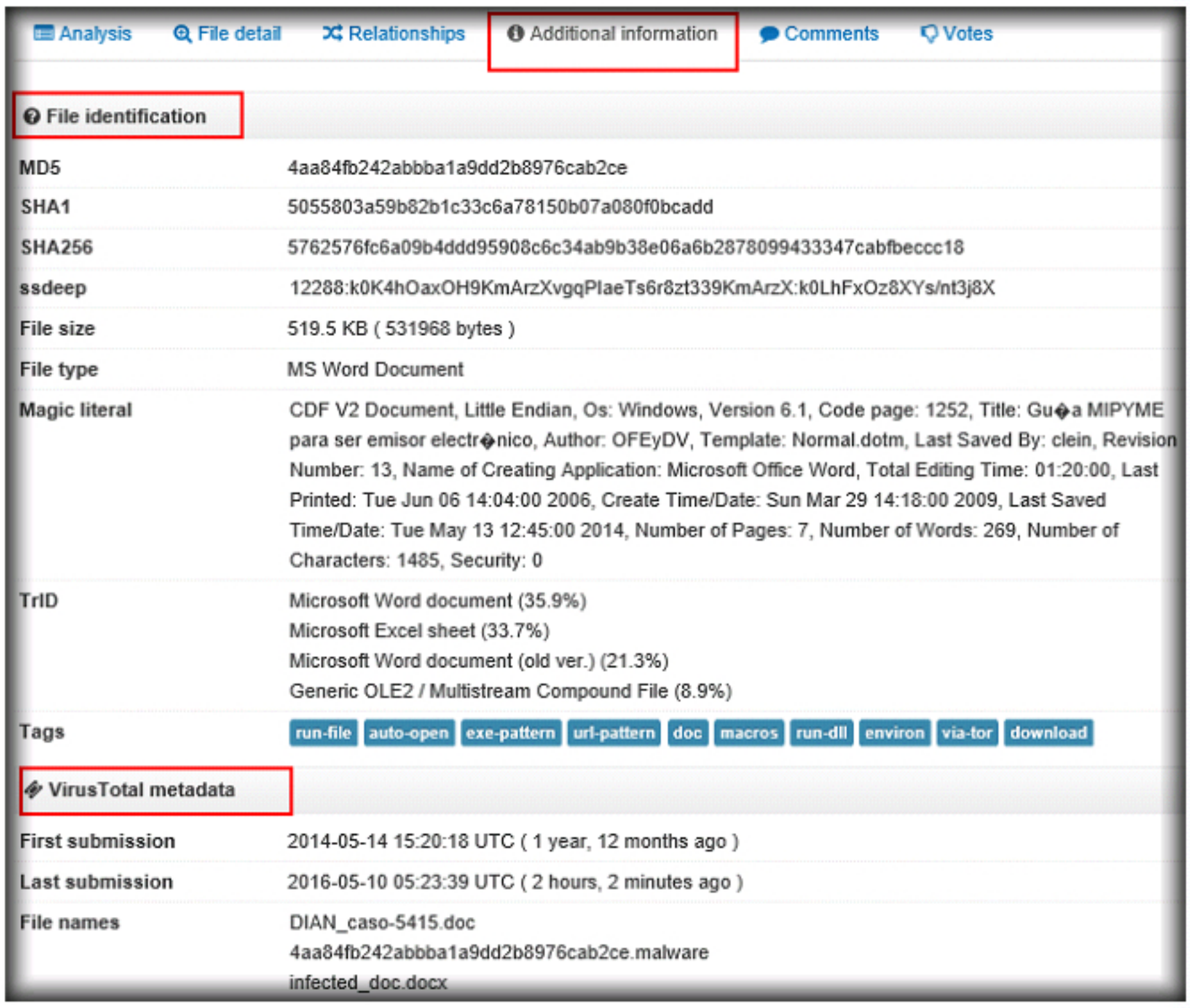


FIGURE 5.8: Viewing the results from the Additional information tab

16. For further analysis use the website **https://malwr.com**. Type the URL https://malwr.com in the address bar and press **Enter**. The malwr webpage appears, click **Submit**.

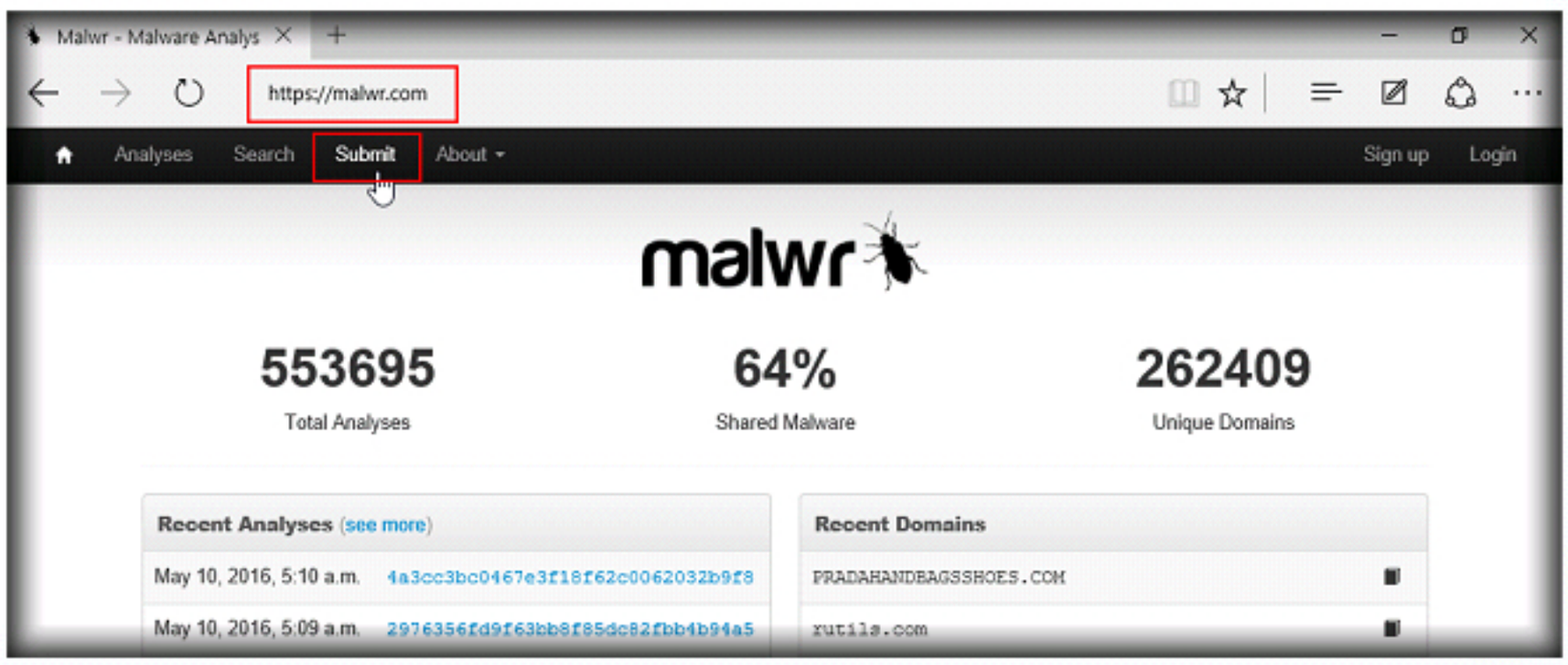


FIGURE 5.9: malwr home page

17. Click **Select file** to upload the suspicious file.

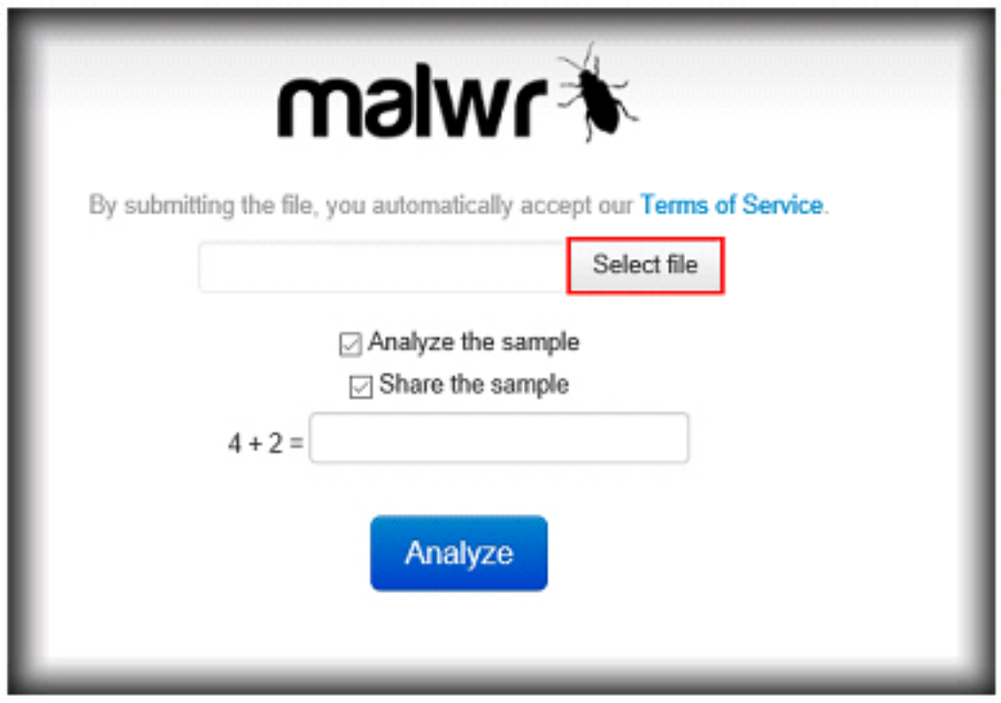


FIGURE 5.10: Screenshot showing the Select file button

18. Navigate to the file location: **Z:\Evidence Files**, select the file infected_doc.docx and click **Open**.

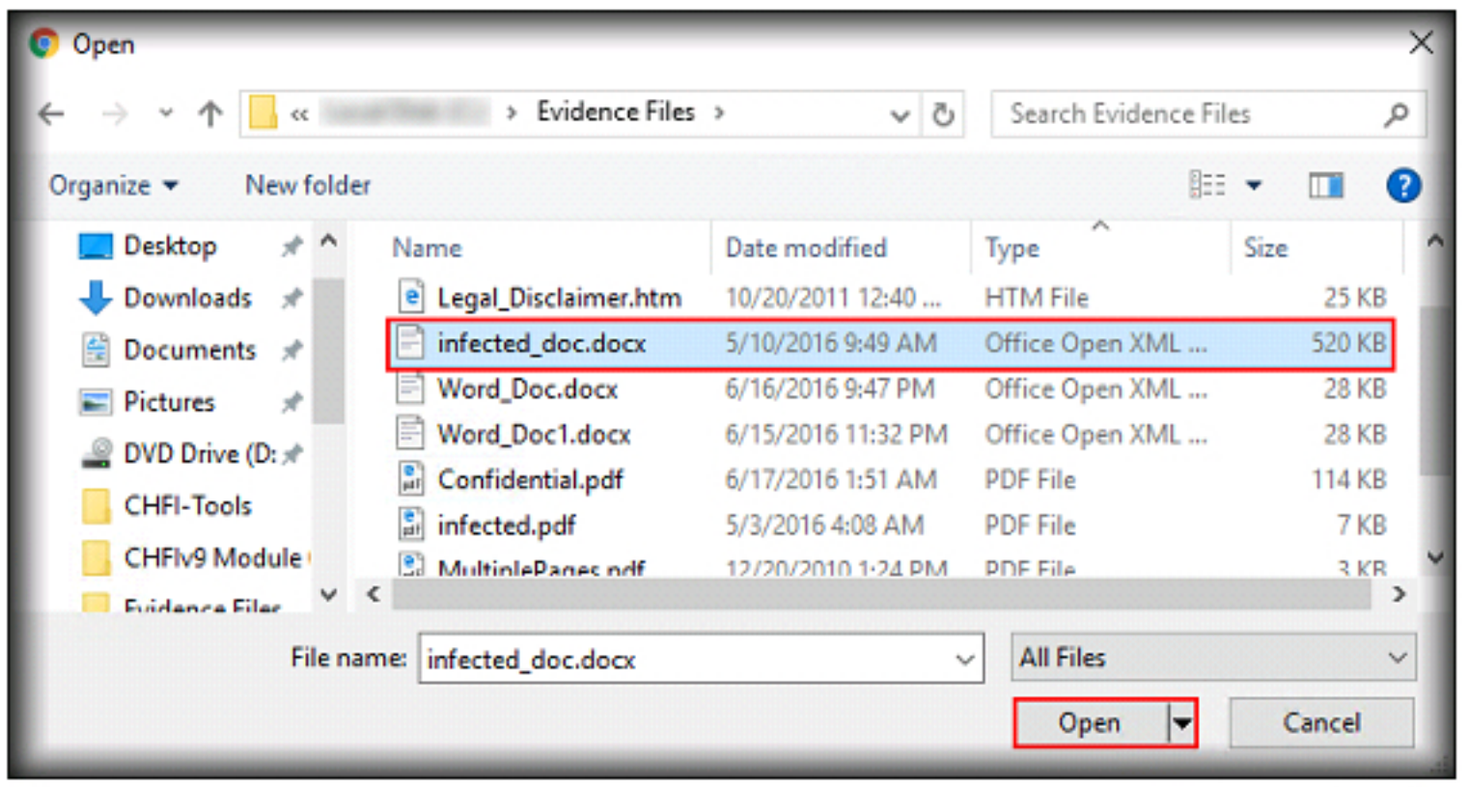


FIGURE 5.11: Navigating to the Evidence Files folder and selecting the suspicious file

19. Answer the question and click **Analyze**.

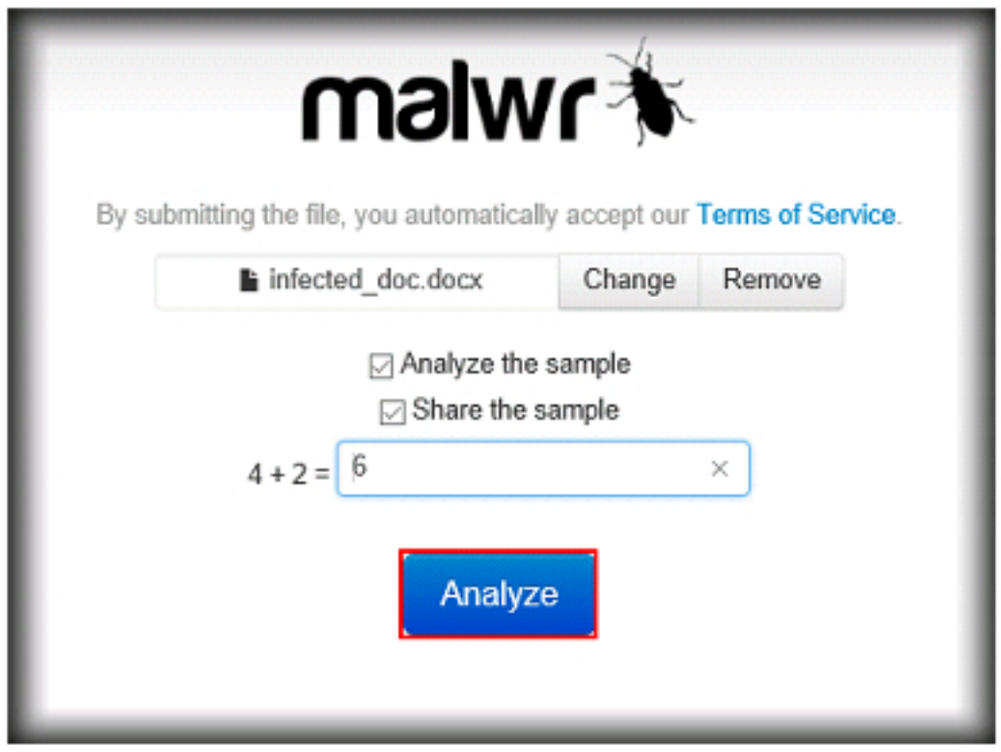


FIGURE 5.12: Performing malware analysis of the suspicious file

Note: In case, the website prompts a window with information on previous analysis of a similar file, click the Reanalyze button.

20. The website displays the **Quick Overview** section containing the size of the file, different hash values, etc.

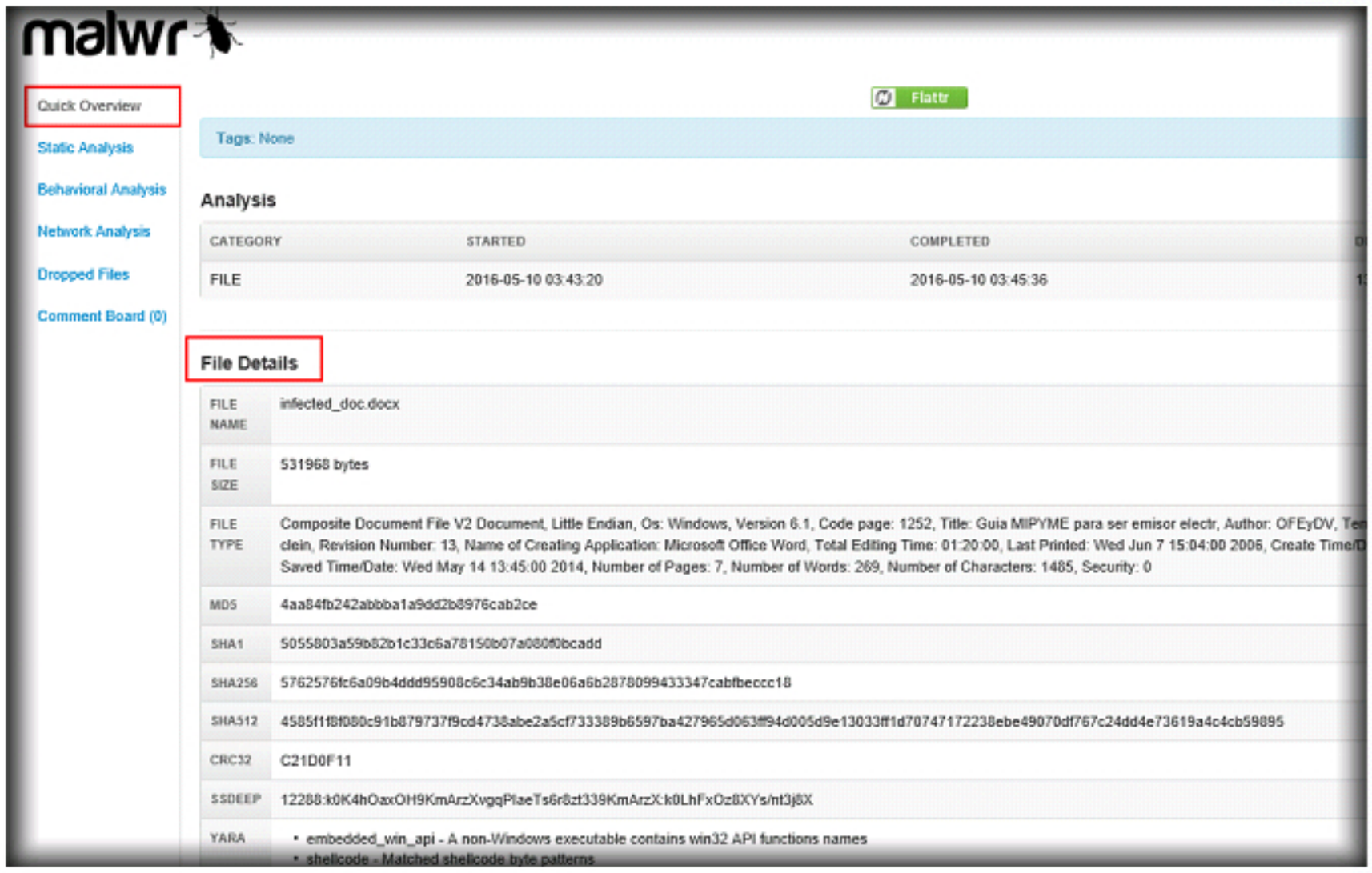


FIGURE 5.13: Viewing the results from the Quick Overview section

21. Click the **Static Analysis** in the left pane and click **Strings** to look at the strings in the file.

FIGURE 5.14: Viewing the results from the Static Analysis section

22. Go through the strings to find suspicious elements and hyperlinks the file tries to connect with. The links in list try to access images or executables from unknown sources. Most of the strings are in Spanish, inferring that the malicious content was either encrypted to divert the attention or the document is of Spanish origin.

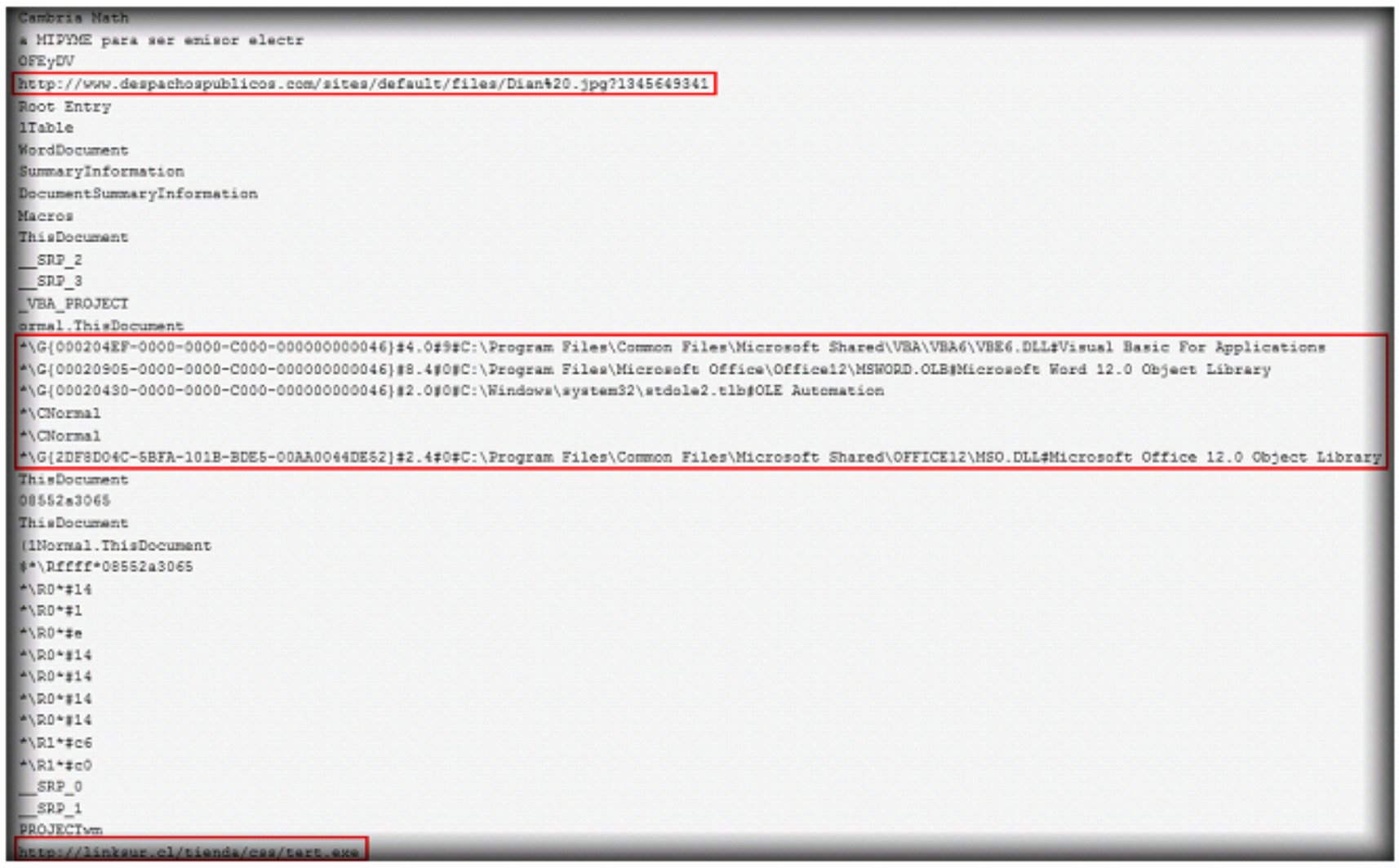


FIGURE 5.15: Examining the strings to find any suspicious elements

23. Click the **Behavioral Analysis** button in the left pane to look at the behavior of the file when executed. The section will reveal all the processes it started including network, filesystem, registry, services and synchronization.

24. The site helps in better understanding by providing different color bands for different types of processes. The site also provides a PID to the malware executable present.

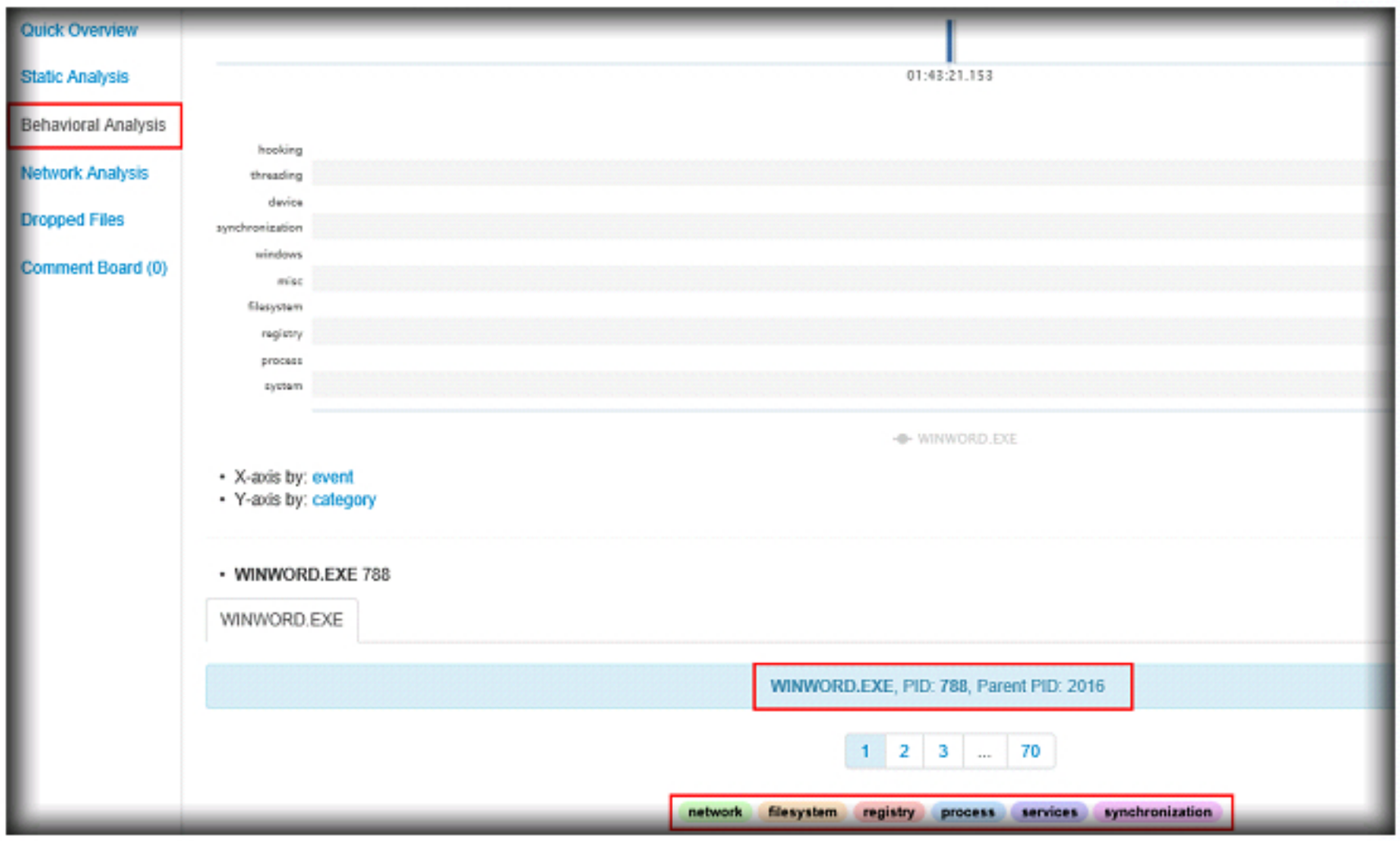


FIGURE 5.16: Viewing the results from the Behavioral Analysis section

25. Scroll down to see a table of all the processes along with API, arguments, status, return and repeated status. This table will help investigators in analyzing all processes the file would initiate and assess their impact on the system and its network.

TIME	API	ARGUMENTS	STATUS	RETURN	REPEATED
2016-05-09 18:43:21,153	LdrGetDllHandle	ModuleHandle: 0x7c800000 FileName: KERNEL32.DLL	success	0x00000000	
2016-05-09 18:43:21,153	LdrGetProcedureAddress	Ordinal: 0 FunctionName: DecodePointer FunctionAddress: 0x7c913405 ModuleHandle: 0x7c800000	success	0x00000000	
2016-05-09 18:43:21,153	LdrGetDllHandle	ModuleHandle: 0x7c800000 FileName: KERNEL32.DLL	success	0x00000000	
2016-05-09 18:43:21,153	LdrGetProcedureAddress	Ordinal: 0 FunctionName: DecodePointer FunctionAddress: 0x7c913405 ModuleHandle: 0x7c800000	success	0x00000000	
2016-05-09 18:43:21,153	LdrGetDllHandle	ModuleHandle: 0x7c800000 FileName: KERNEL32.DLL	success	0x00000000	
2016-05-09 18:43:21,163	LdrGetProcedureAddress	Ordinal: 0 FunctionName: EncodePointer FunctionAddress: 0x7c9133df ModuleHandle: 0x7c800000	success	0x00000000	
2016-05-09 18:43:21,163	LdrGetProcedureAddress	Ordinal: 0 FunctionName: DecodePointer	success	0x00000000	

FIGURE 5.17: Analyzing the process the file would initiate

26. Click on the other sections to analyze them. The file has not produced any network differences. Click the **Dropped Files** section to view the dropped files.

27. The website reveals that the file will drop two files onto the system and also provides their details, such as size, type and hash values.

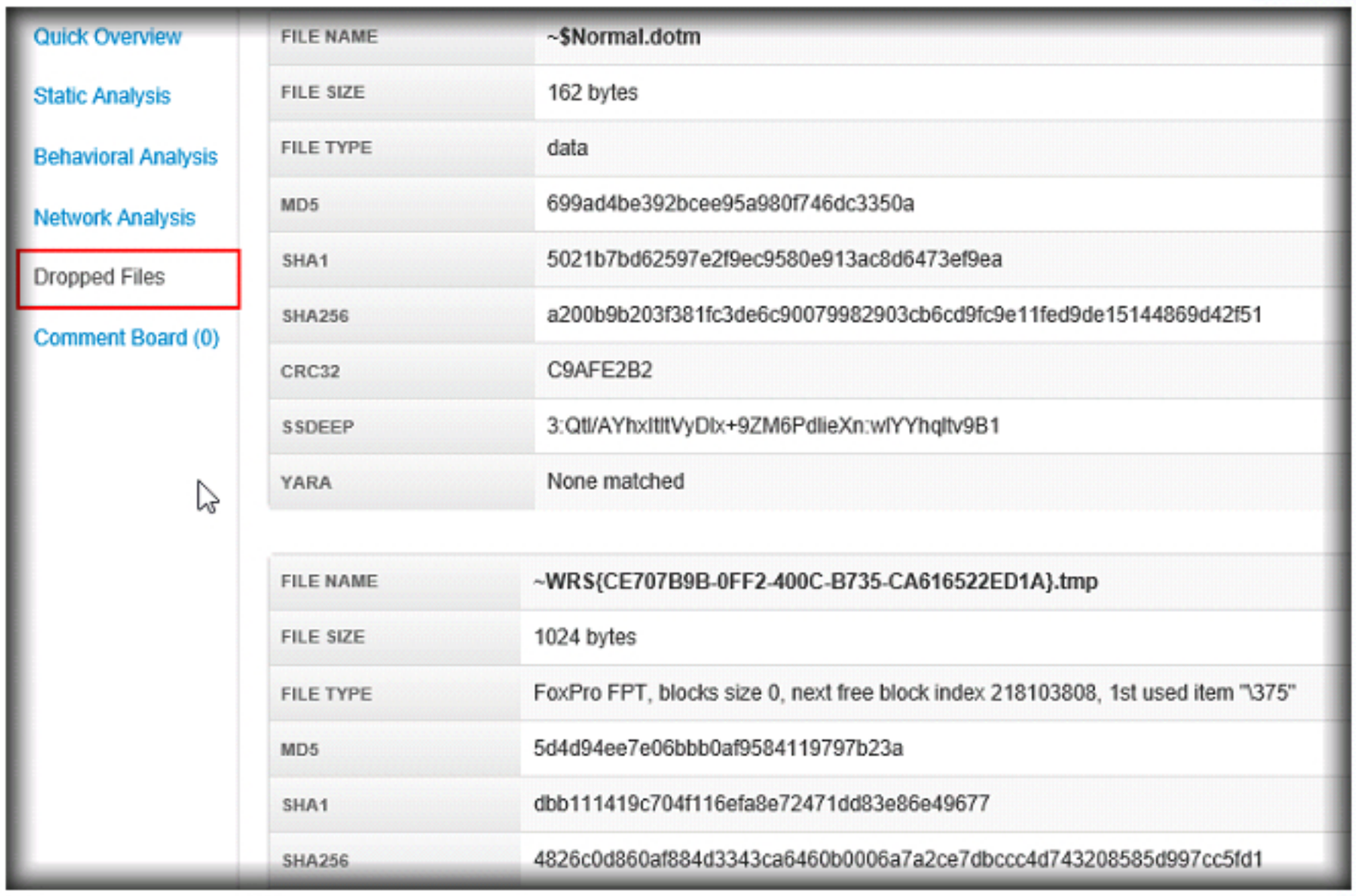


FIGURE 5.18: Viewing the results from the Dropped Files section

28. Though it has been proved that the file contains malicious content using online tools, investigators also need to check what type of coding has made it so and what its impact on the system.

29. To launch **Offvis** tool, navigate to the **Z:\CHFIv9 Module 11 Malware Forensics\Malicious Document Analysis Tools\OffVis** location and double click the **OffVis.exe** file.

Note: The tool requires .Net Framework 3.5 to install and function properly.

30. OffVis GUI appears, Click **File** from the menu bar and select **Open Data File...** option from the drop down list.

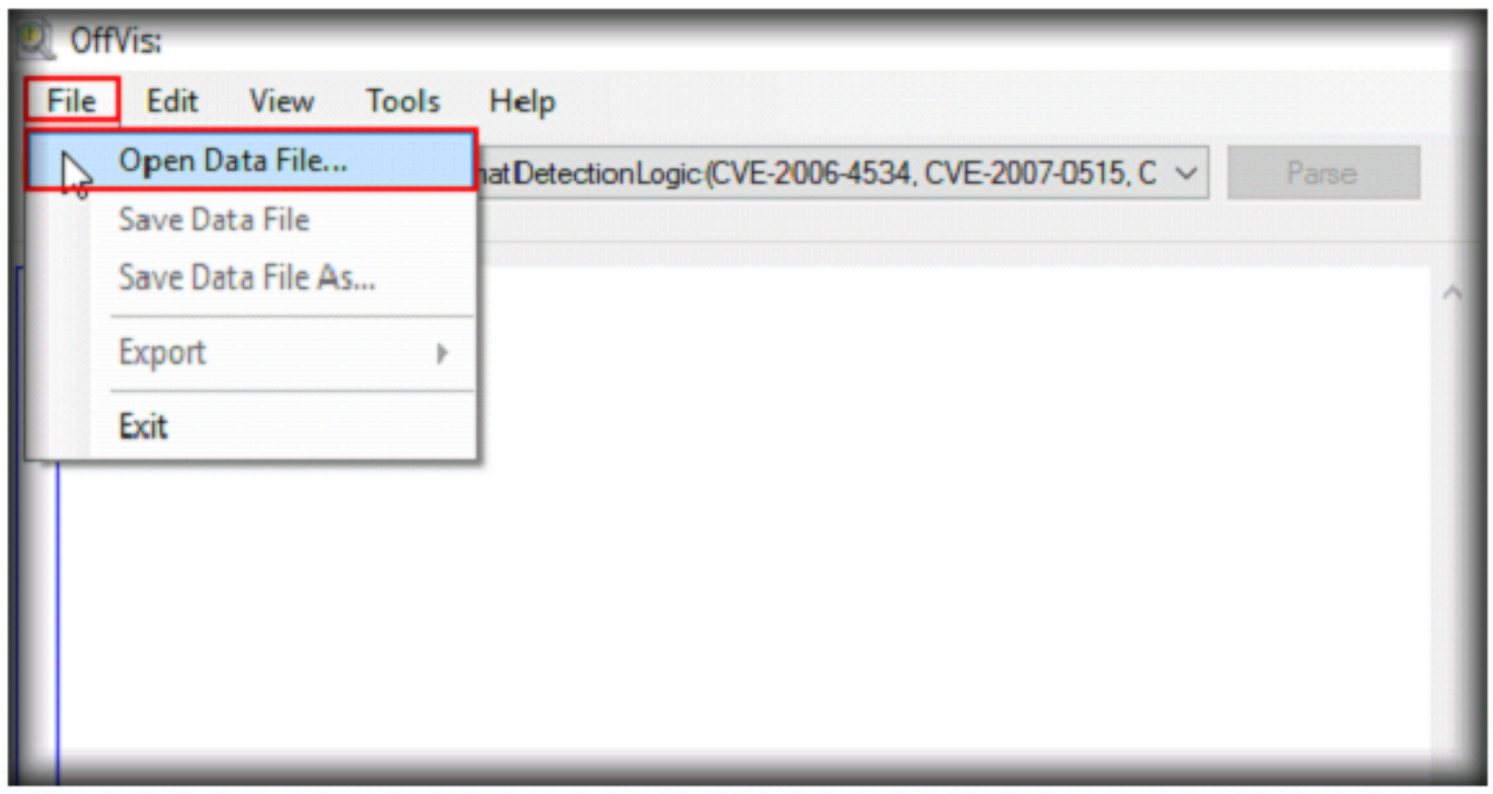


FIGURE 5.19: Browsing the suspicious file in OffVis tool

31. **Select A Data File** window appears; Navigate to the location of the suspicious file, i.e., **Z:\Evidence Files** select the file **infected_doc.docx** and click **Open**.

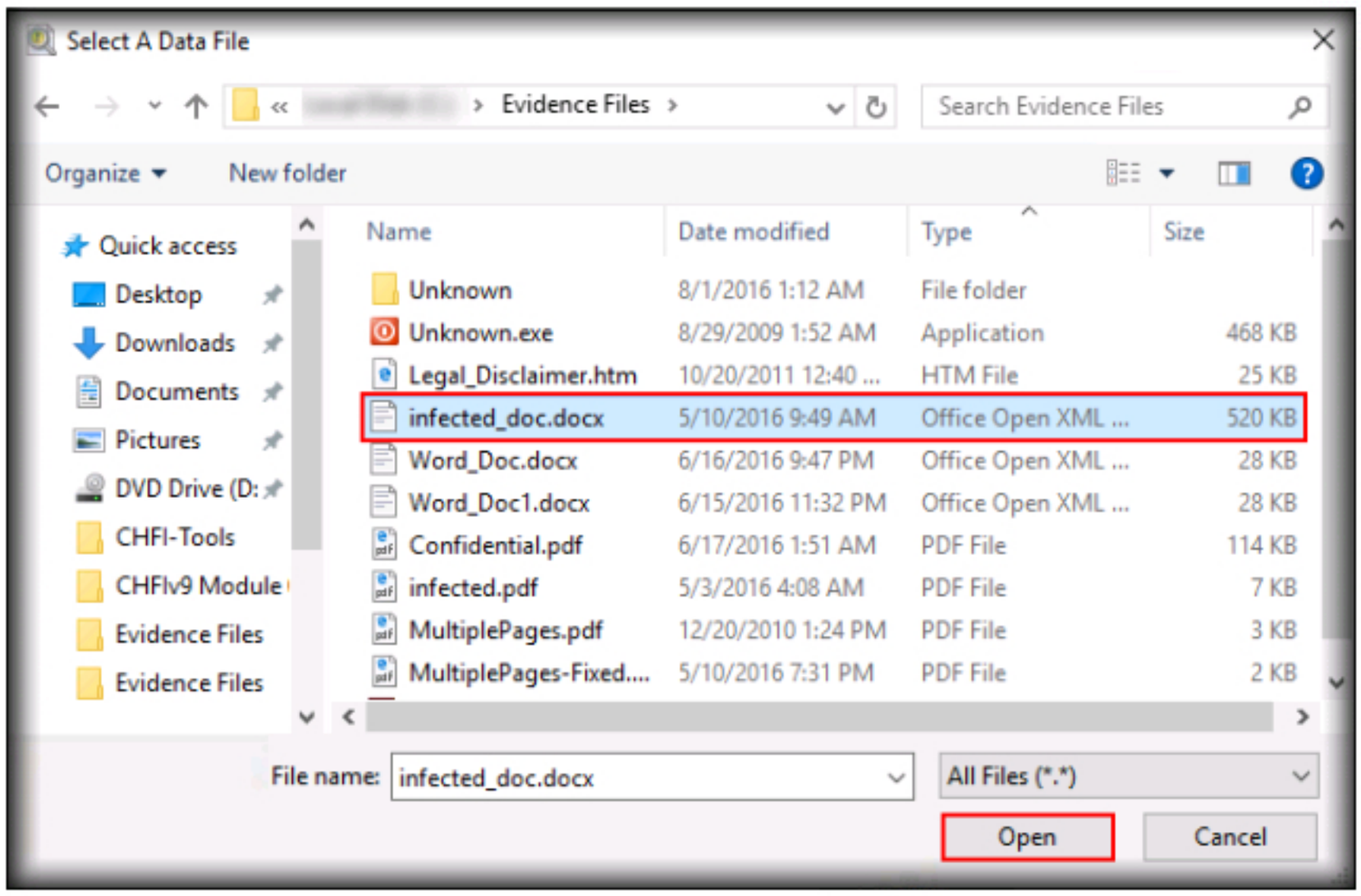


FIGURE 5.20: Navigating to the Evidence Files folder and selecting the suspicious file

32. After uploading the file select the format to parse it in. Click the **Parser** drop-down list located under the toolbar and select **WordBinaryFormatDetectionLogic** function.

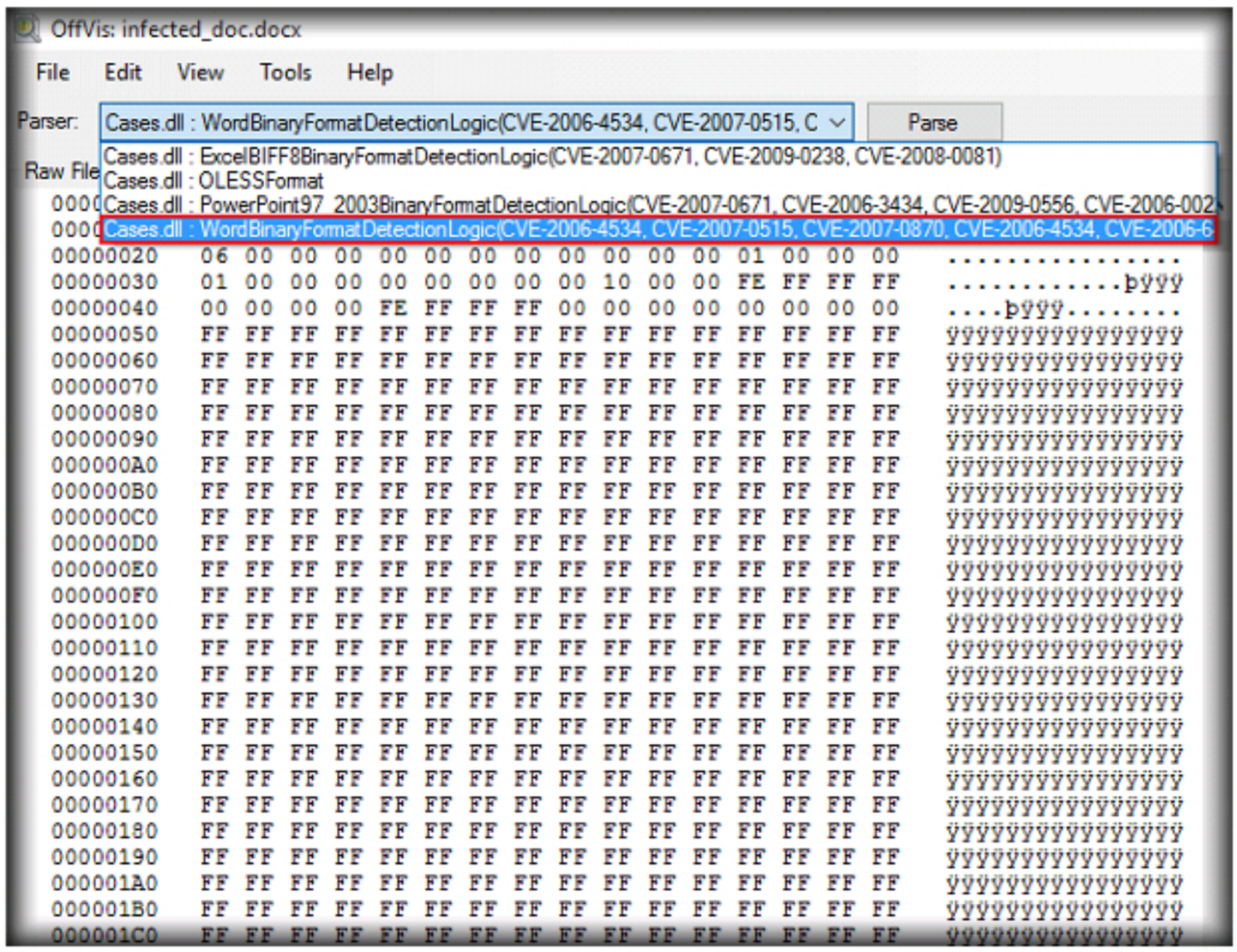


FIGURE 5.21: Selecting the required function for parsing the file

33. Click on the **Parse** button.

FIGURE 5.22: Initiating the Parser function

34. The tool will display hex data of the file along with addresses of the data representing the string or element location in the file.

35. Scroll down to look at the recognizable strings in the tool. The tool displays the code: "**INCLUDEPICTURE "C:\\Users\\clein\\Desktop\\a.png" * MERGEFORMATINET**", which seems like a command for adding a picture named a.png to the system desktop.

36. Click on the string, and the tool will display its location in the word file on the right side in the parsing results column.

37. Further analysis reveals that the file tries to access the **www.dian.gov.co** website. The other parts of the string appear to be a different language. Scroll down to find more strings that help to trace malice in the file.

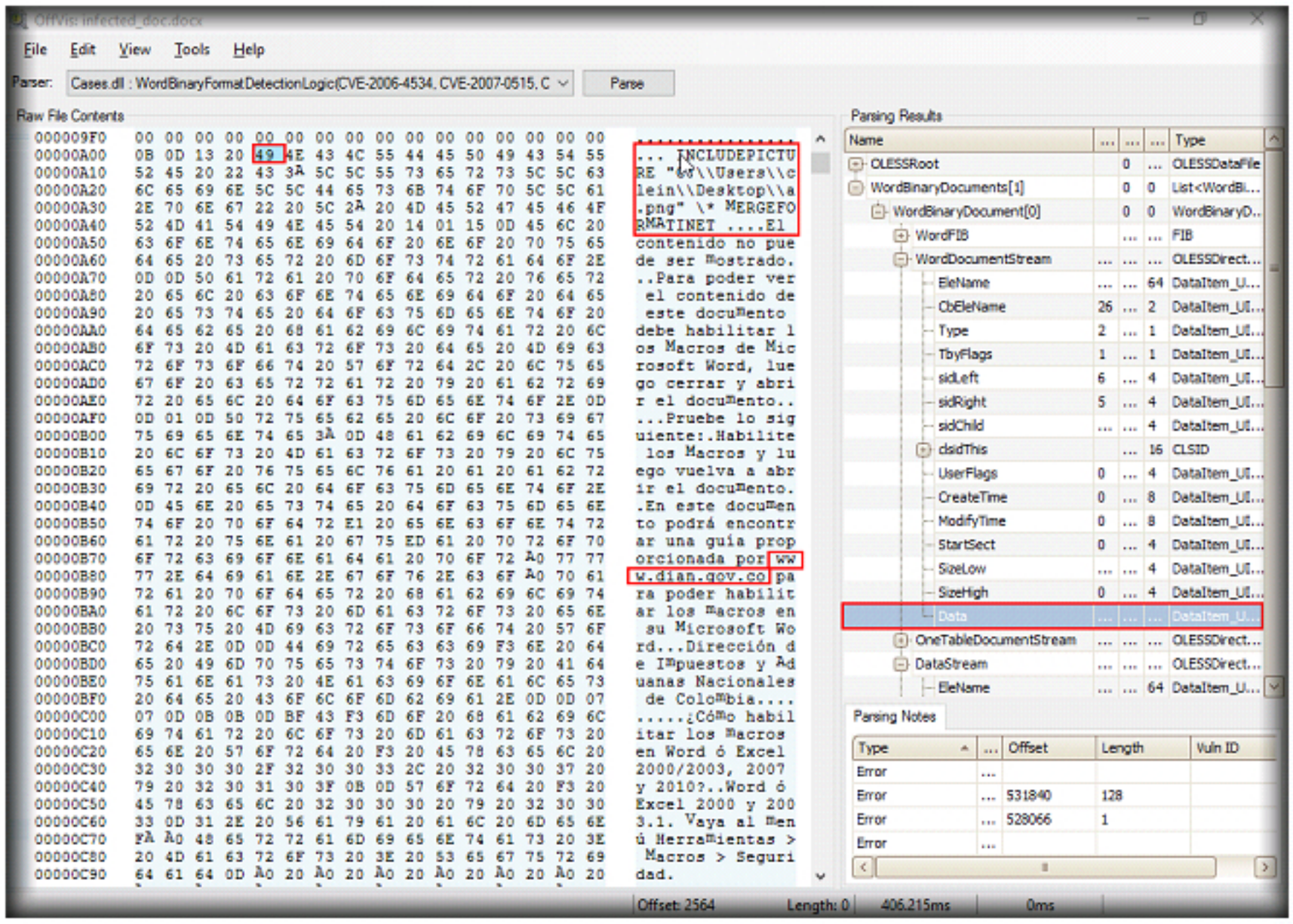


FIGURE 5.23: Examining the internal details of the suspect file

38. Close observation reveals that the file tries to connect with the faraway browser through embedded URL links such as:

http://ns.adobe.com/xap/1.0/

http://www.w3.org/1999/02/22-rdf-syntax-ns#

http://purl.org/dc/elements/1.1/"

http://ns.adobe.com/xap/1.0/

http://purl.org/dc/elements/1.1

http://www.w3.org/1999/02/22-rdf-syntax-ns#

39. The file also contains SQL manipulation scripts along with these links that could help the attacker bypass various security features.

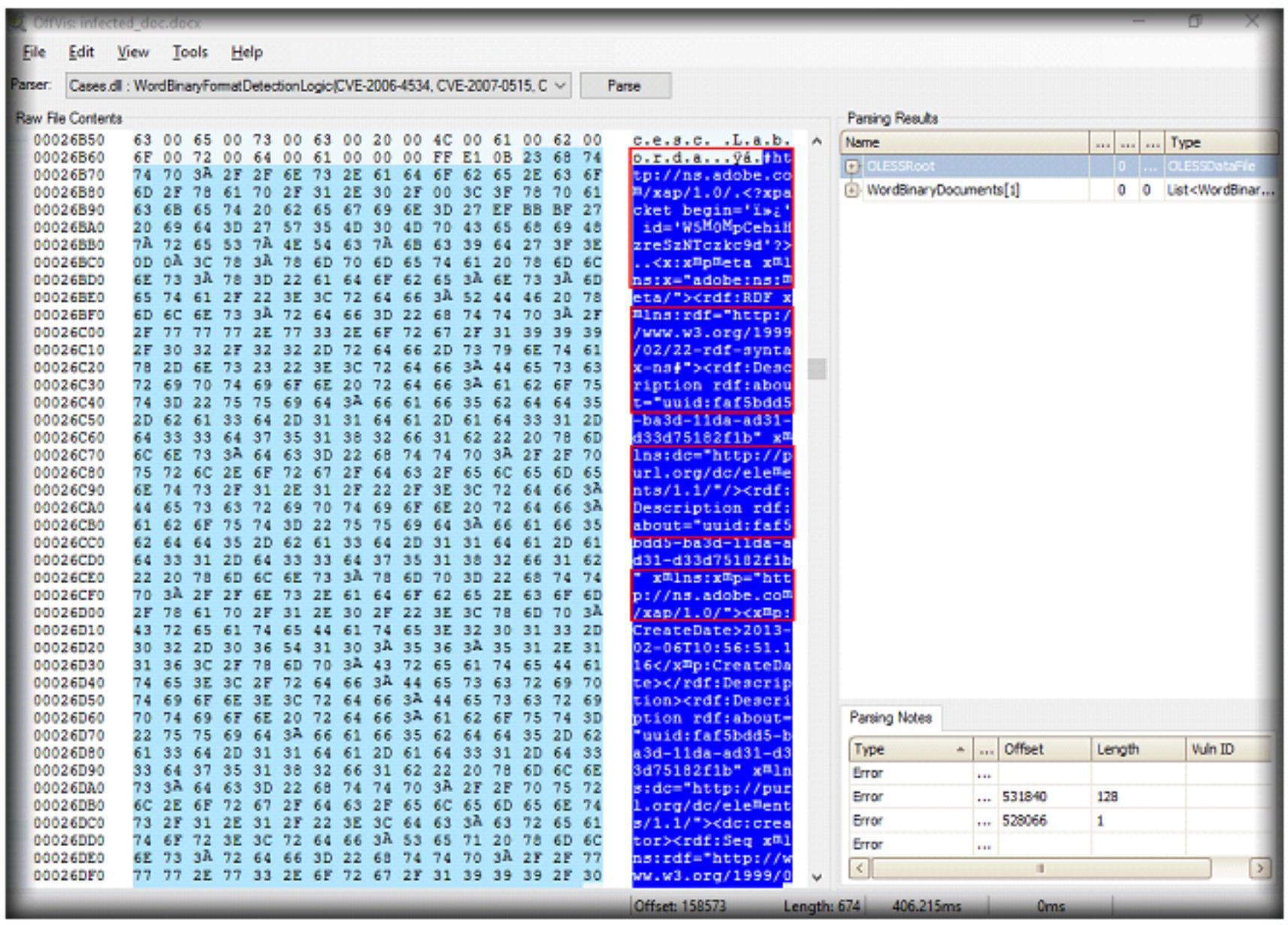


FIGURE 5.24: Performing analysis on the suspicious elements of the suspect file

40. The table of hex value shows that the coding also includes a program to manipulate internal files of the corrupted system. The code tries to add new VBA structures to the system DLLs and also share them with connected systems in the form of word files.

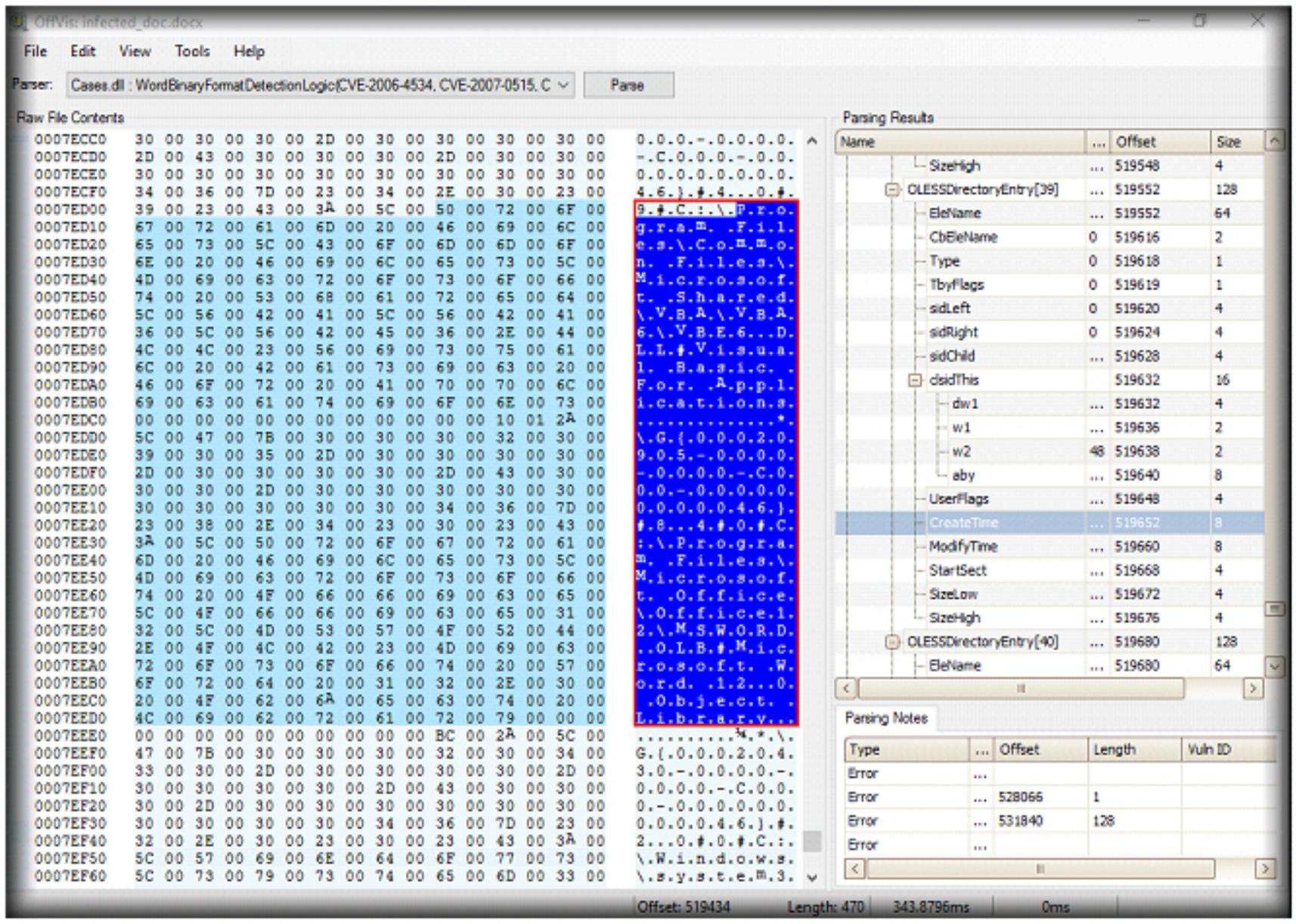


FIGURE 5.25: Performing analysis on the suspicious elements of the suspect file

41. The file also consists of coding that tries to modify the system32 files and other important directories.

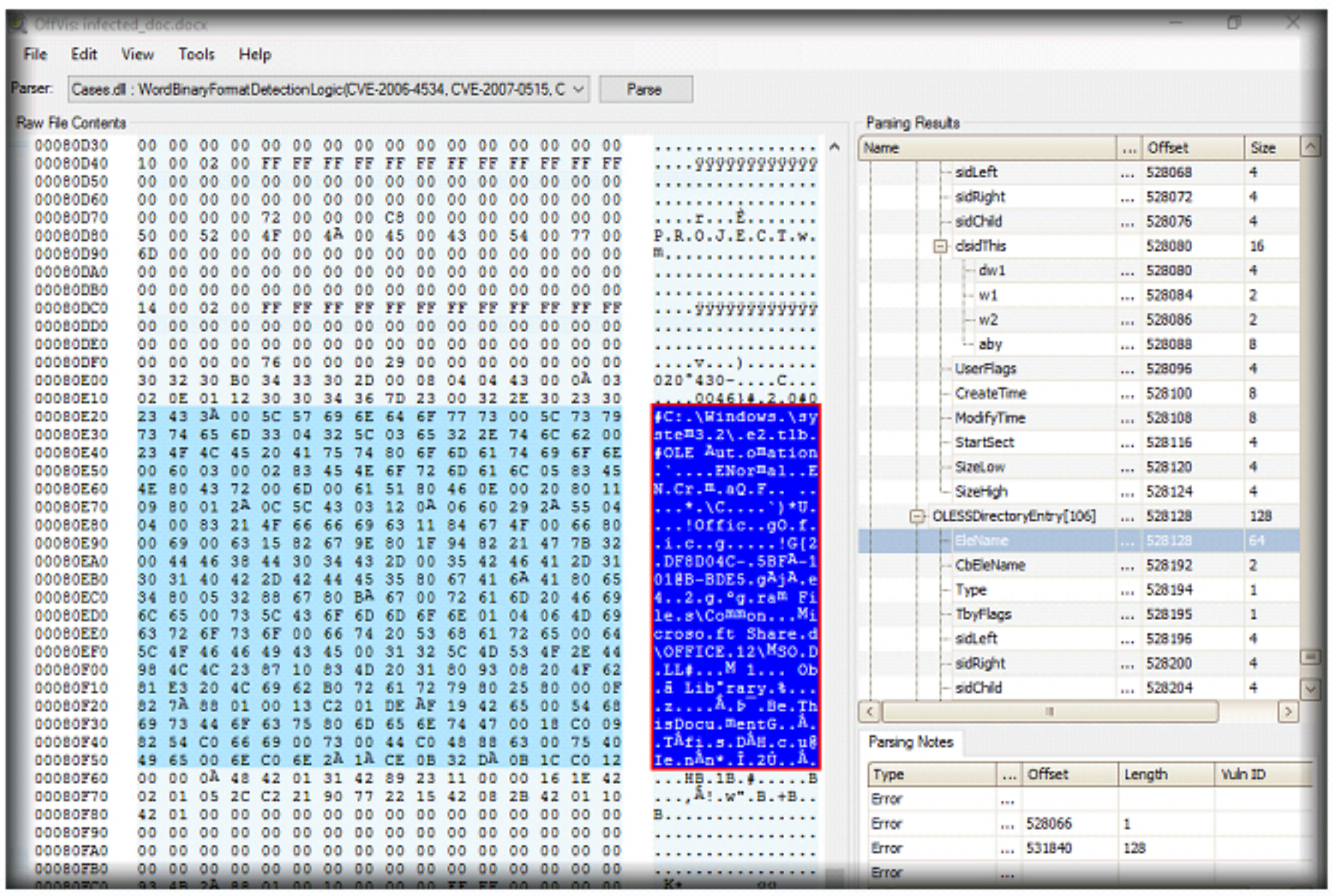


FIGURE 5.26: Performing analysis on the suspicious elements of the suspect file

42. The file consists codes to manipulate system program files such as:

    C:\Program Files\Common Files\Microsoft Shared\VBA\VBA6\VBE6.DLL

    C:\Program Files\Microsoft Office\Office12\MSWORD.OLB

    C:\Windows\system32\stdole2.tlb

    C:\Program Files\Common Files\Microsoft Shared\OFFICE12\MSO.DLL

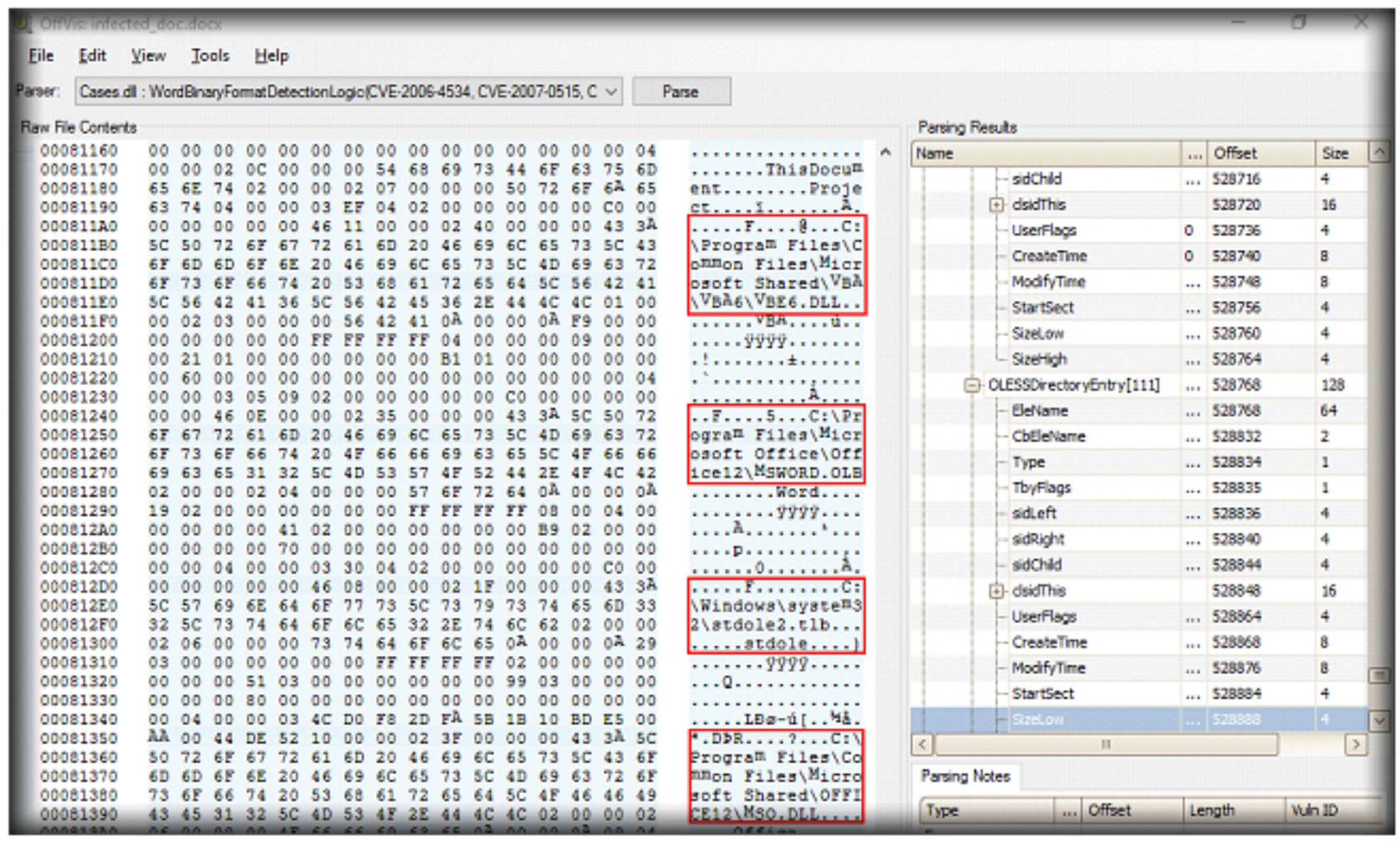


FIGURE 5.27: Performing analysis on the suspicious elements of the suspect file

43. Other commands and codes available in the file include RLDownloadToFileA, urlmon, URLDownloadToFileA, AutoOpen, Auto_Open, SNVJYQ, OGEXYR, Workbook_Open, etc.

44. The word document tries to access external links, such as http://bit.ly/1gudjPa and http://linksur.cl/tienda/css/tert.exe, to download unknown file \lkjljlljk and an executable named \sfjozjeri.exe.

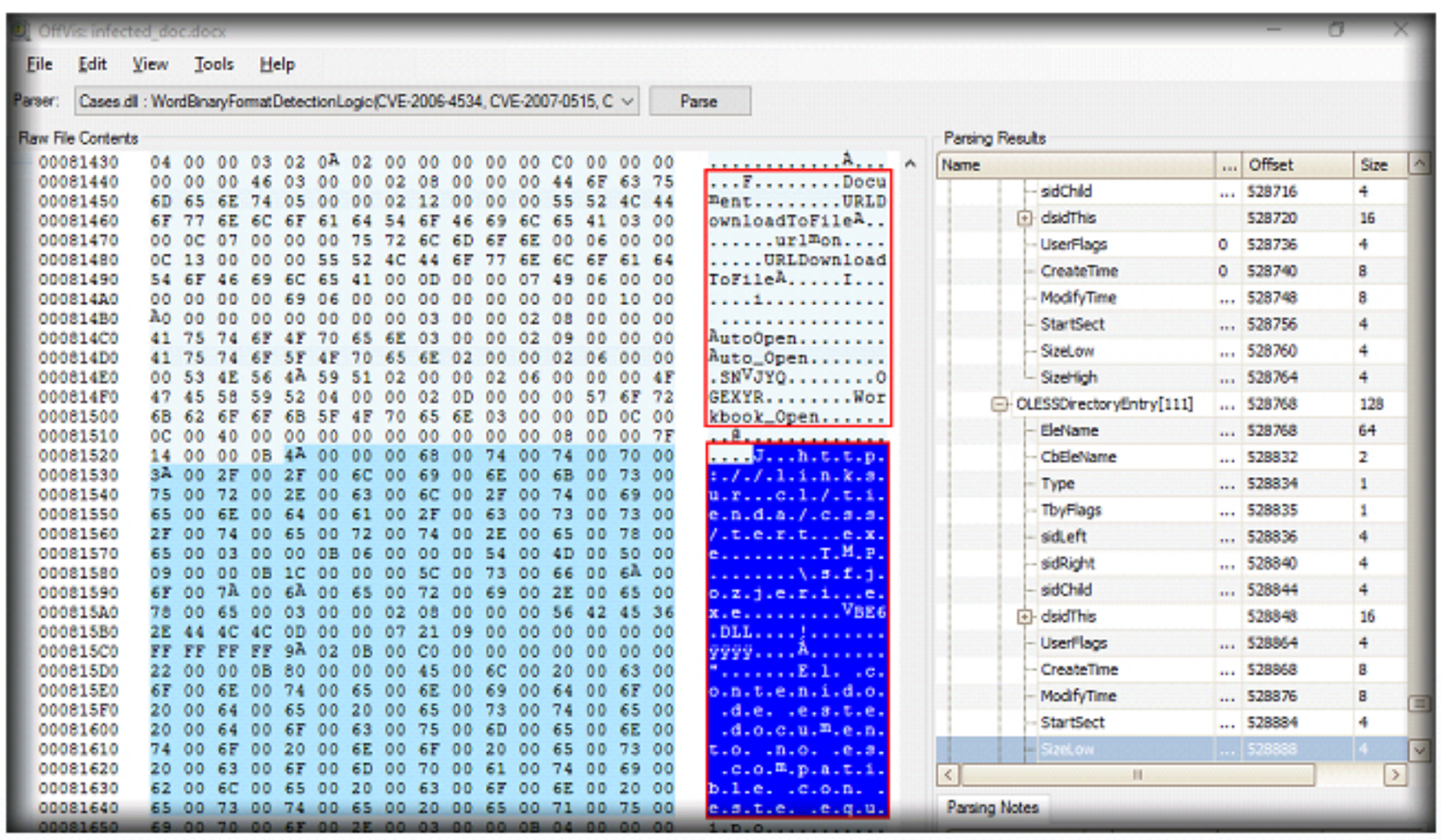


FIGURE 5.28: Performing analysis on the suspicious elements of the suspect file

45. The word document also includes encrypted hyperlinks that help to access external executables, download them onto the system and run them to perform malicious activities.
46. The file tries to access the link "http://germanya.com.ec/logs/test.exe" to download test.exe and sfjozjero.exe.
47. http://germanya.com.ec/logs/counter.php link is present that seems to be a PHP website that gathers log information of the compromised system.

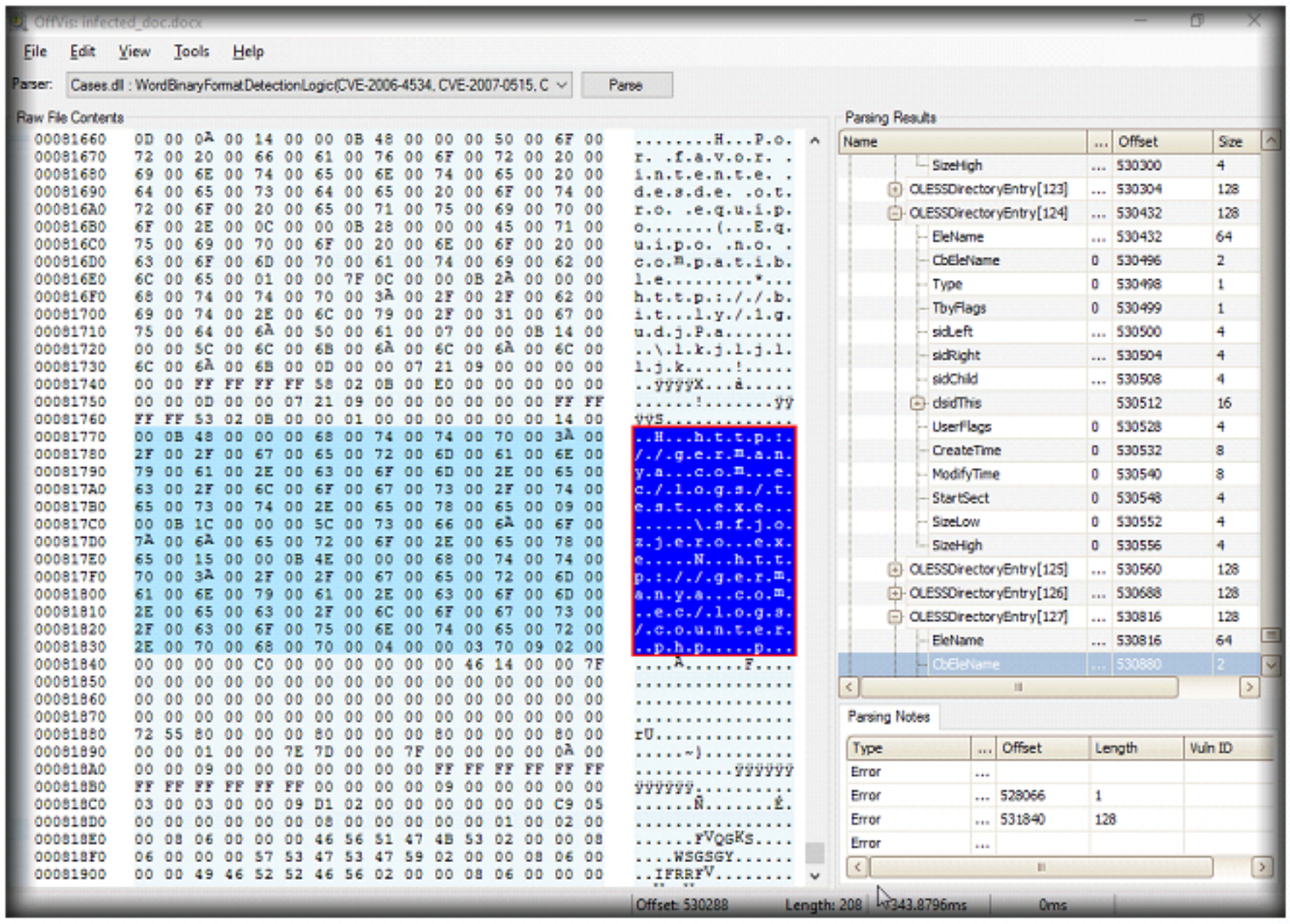


FIGURE 5.29: Performing analysis on the suspicious elements of the suspect file

48. The file also launches various encrypted commands to control the flow and other variables. Some of the commands include:
    - ***\G{000204EF-0000-0000-C000-000000000046}#4.0#9#C:\Program Files\Common Files\Microsoft Shared\VBA\VBA6\VBE6.DLL#Visual Basic For Applications**
    - ***\G{00020905-0000-0000-C000-000000000046}#8.4#0#C:\Program Files\Microsoft Office\Office12\MSWORD.OLB#Microsoft Word 12.0 Object Library**
    - ***\G{00020430-0000-0000-C000-000000000046}#2.0#0#C:\Windows\system32\stdole2.tlb#OLE Automation**
    - ***\CNormal**
    - ***\CNormal**
    - ***\G{2DF8D04C-5BFA-101B-BDE5-00AA0044DE52}#2.4#0#C:\Program Files\Common Files\Microsoft Shared\OFFICE12\MSO.DLL#Microsoft Office 12.0 Object Library**

49. Investigators can also click on errors the tool displays at the bottom right. The second error points towards the discrepancies in the end of the word file.

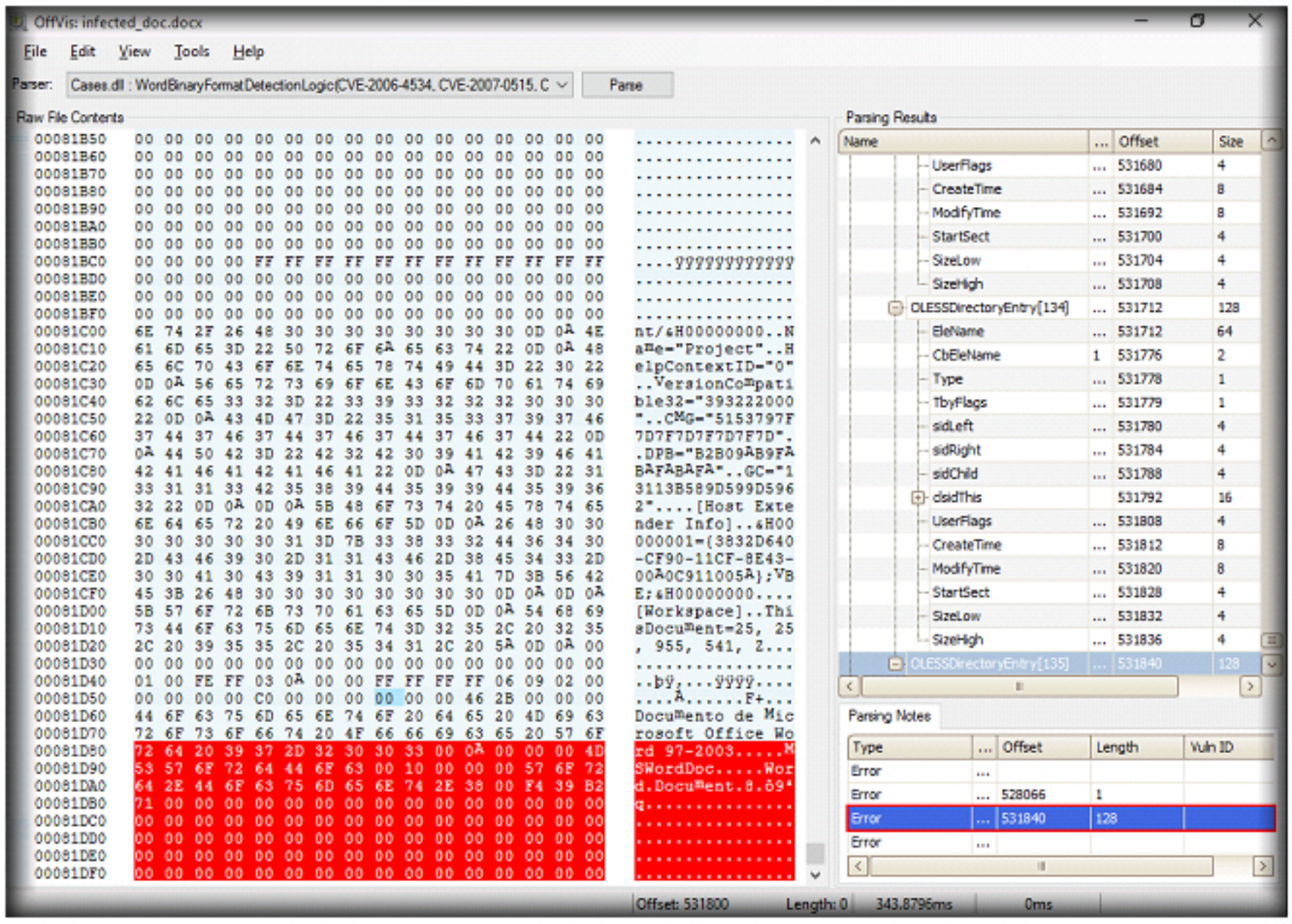


FIGURE 5.30: Examining the errors displayed by the tool

Lab Analysis

Document all the findings you obtained during this lab.

ASK YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs