Windows Forensics

Module 06

Windows Forensics

Windows forensics involves obtaining and analyzing digital information in Windows OSes for use as evidence in civil, criminal, or administrative cases

ICON KEY

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

Lab Scenario

A computer forensics examiner, Steve, was called to investigate the laptop of a 26-year-old man who was arrested. Steve started searching the contents of the laptop. He began his investigation on Windows® event logs and processes using various Windows forensic tools. He checked all the registries, event logs, and processes for evidence of any crimes. During the investigation, Steve found the paths for several images and videos pertaining to the nefarious activities performed by the suspect. He checked all the pictures and confirmed the existence of illicit activities on the laptop. Other evidence on the laptop proved that the man in custody was its primary user.

Lab Objectives

The goal of this lab is to explain the process of finding pieces of evidence from Windows OSes. Evidence in Windows OSes includes Windows memory analysis, volatile and non-volatile information, Windows event logs, Windows processes, search key values, and other data. Accomplishing this task will entail the following:

- Acquiring and investigating volatile memory contents of a Windows system
- Analyzing the Windows registry
- Examining forensic artifacts from web browsers
- Identifying and extracting forensic evidence from computers
- Investigating loaded processes on a live computer
- Analyzing event logs of a Windows machine
- Performing forensics investigation on a Windows machine

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics

Lab Environment

This lab requires:

- A computer running **Windows Server 2016** virtual machine
- A computer running **Ubuntu Forensics** virtual machine
- A web browser with internet connection
- Administrative privileges to install and run tools

Lab Duration

Time: 185 minutes

Overview of Windows Forensics

Windows forensics refers to investigation of cyber-crimes involving Windows machines. It involves gathering of evidence from a Windows machine so that the perpetrator(s) of a cyber-crime can be identified and prosecuted. Investigators performing forensics on Windows machines must have a thorough understanding of the various components of a Windows OS such as the file system, registries, system files, and event logs where they can find data of evidentiary value.

Lab Tasks

Recommended labs to assist you in Windows forensics:

- Acquiring Volatile Information from a Live Windows System
- Investigating Forensic Image of Windows RAM
- Examining Web Browser Artifacts
- Discovering and Extracting Forensic Data from Computers
- Extracting Information about Loaded Processes on a Computer
- Viewing, Monitoring, and Analyzing Events Occurred on a Windows Machine
- Performing Digital Forensic Investigation on a Computer
- Collecting and Parsing Forensic Artifacts on a Live Windows Machine

Lab Analysis

Analyze and document the results related to the lab exercise.

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

Lab

1

Acquiring Volatile Information from a Live Windows System

Acquiring volatile information from a live system involves obtaining information such as network information and process information about an OS.

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

Lab Scenario

For forensics investigation, investigators often need to gather data from a live Windows system to analyze details such as network information and process information by using different tools (command-line tools as well as GUI-based tools). Performing a thorough analysis will enable investigators to obtain vital evidence, which helps them solve various cases related to digital forensics.

As a forensics investigator, you should know how to collect volatile information from a live system.

Lab Objectives

The objective of this lab is to help you collect volatile information from a live Windows system.

Lab Environment

This lab requires:

- A computer running **Windows Server 2016** virtual machine
- A computer running **Windows 10** virtual machine
- Administrative privileges to execute commands
- A web browser with internet access
- **PsLoggedon** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Volatile Data Acquisition Tools\PsTools**

- **LogonSessions** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Volatile Data Acquisition Tools\LogonSessions**
- **NetworkOpenedFiles** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Volatile Data Acquisition Tools\NetworkOpenedFiles**

Note: You can download the latest version of **PsLoggedon** from the link https://docs.microsoft.com/en-us/sysinternals/downloads/psloggedon

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics

You can download the latest version of **LogonSessions** tool from the link https://docs.microsoft.com/en-us/sysinternals/downloads/logonsessions

You can download the latest version of **NetworkOpenedFiles** tool from the link https://www.nirsoft.net/utils/network_opened_files.html

If you are using the latest version of software for this lab, then the steps and screenshots demonstrated in the lab might differ.

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 15 minutes

Overview of the Lab

This lab familiarizes you with the procedures to collect volatile information from a host computer running on a Windows OS by using tools such as **PsTools**, **LogonSessions**, and **NetworkOpenedFiles.**

Lab Tasks

1. Login to the **Windows Server 2016** and **Windows 10** virtual machines.

TASK 1

Gather Information About Logged On Users

Note: Our task in this lab is to collect volatile information from a host machine. We will be using **Windows Server 2016** virtual machine as the host machine and **Windows 10** virtual machine as a locally connected machine. Therefore, throughout the course of this lab, ensure that the **Windows 10** virtual machine remains powered on.

2. In **Windows Server 2016** virtual machine, navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Volatile Data Acquisition Tools**.

3. Select the **PsTools** folder, hold the **Shift** key on the keyboard, right-click on the selected folder, and then select **Open command window here** from the context menu.

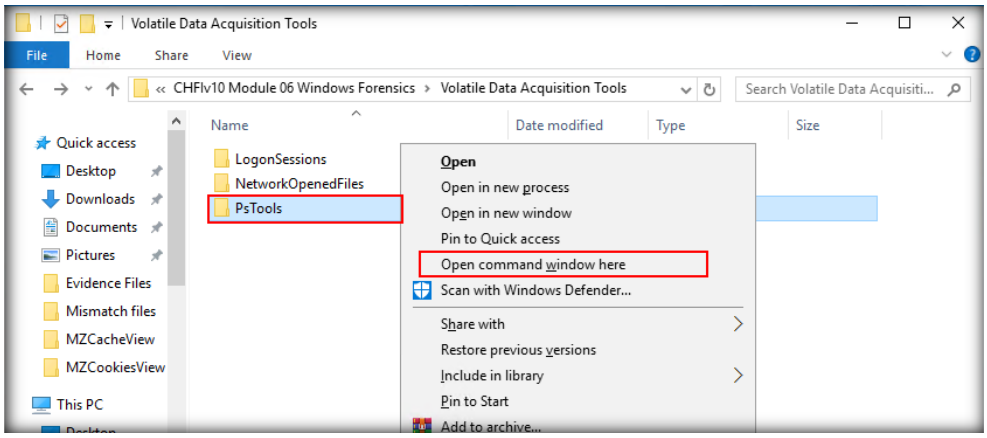


FIGURE 1.1: Select Open command window (Open PsTools)

4. The above operation will launch **Command Prompt** displaying the path of **PsTools**. Type **PsLoggedon64.exe** and press **Enter**.

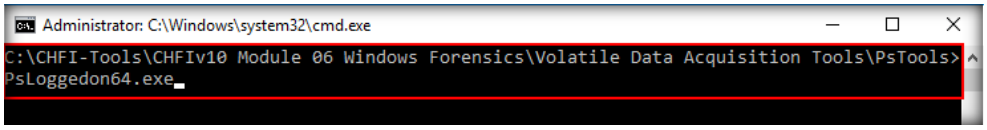


FIGURE 1.2: Type PsLoggedOn64.exe

5. The **PsLoggedon License Agreement** window will appear. Click **Agree** to continue.

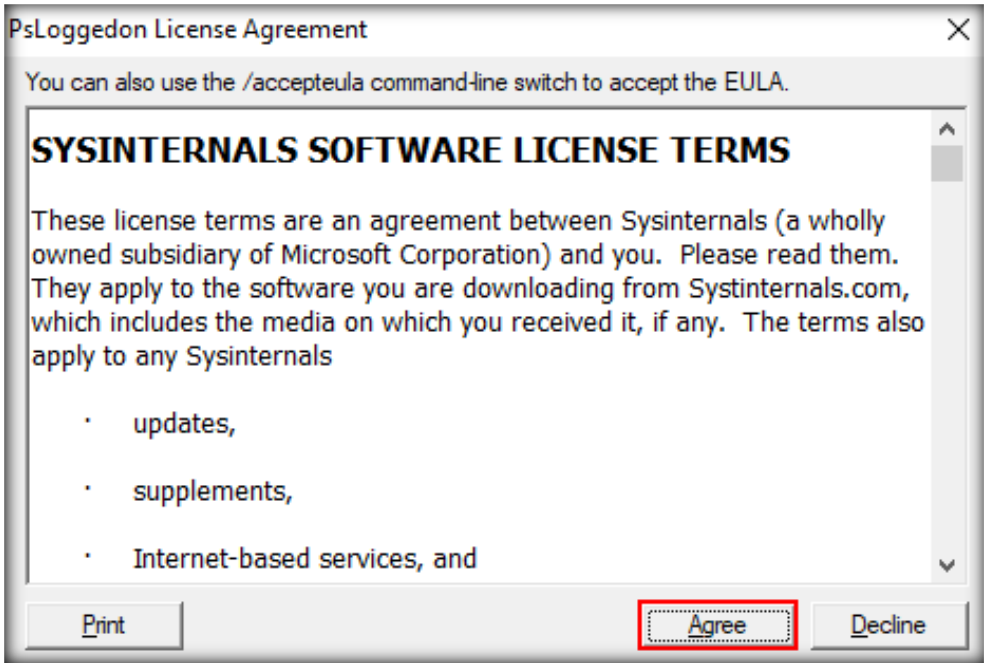


FIGURE 1.3: PSLoggedOn License Agreement window

6. The **Command Prompt** window will now display two results: **Users Logged on locally** (here, the local user is the **Administrator** of the local machine) and **Users Logged on via resource shares** (in this case, no user is logged on through resource shares; hence, the result displays **null** in the **Users logged on via resource shares** section).

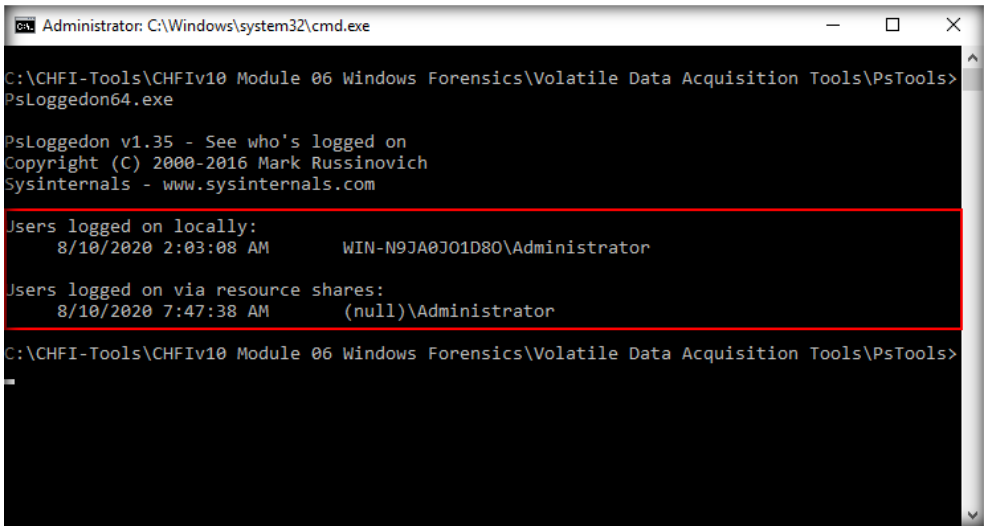


FIGURE 1.4: Results of PsLoggedOn64.exe

7. Now, close the **Command Prompt** window and launch it as an **Administrator** by right clicking the **Start** button (**Windows** icon button) and then clicking on **Command Prompt (Admin)**. We will now run the **net sessions** command, which will list the IP addresses of all the connected sessions on the host machine (here, **Windows Server 2016**), the command has listed the IP address of the **Windows 10** virtual machine as shown in the following screenshot:

Note: The **net sessions** command can be run only on **Windows Server 2016** virtual machine and the result might differ in your lab environment.

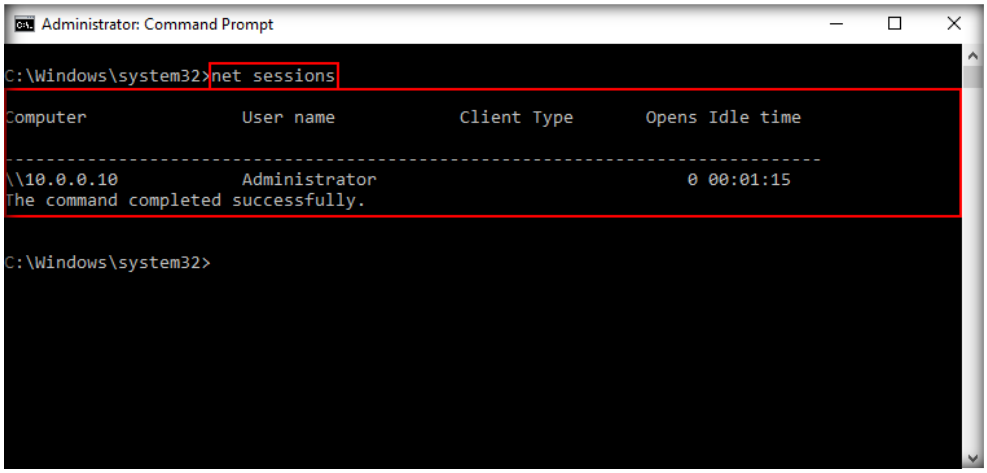


FIGURE 1.5: Execution of the net sessions command

8. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Volatile Data Acquisition Tools**, select the **LogonSessions** folder, hold the **Shift** key on the keyboard, right click the selected folder, and select **Open command window here** from the context menu.

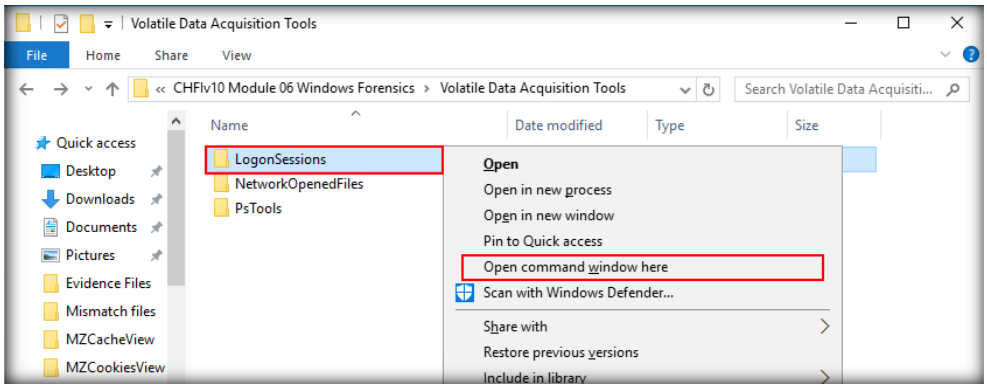


FIGURE 1.6: Select Open command window here

9. The above operation will launch **Command Prompt** with the path of **LogonSessions**. Type **LogonSessions64.exe** and press **Enter**.

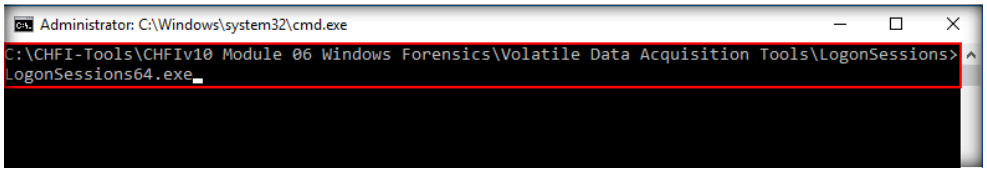


FIGURE 1.7: Type logonsessions64.exe

10. The **LogonSessions License Agreement** window will appear. Click **Agree** to continue.

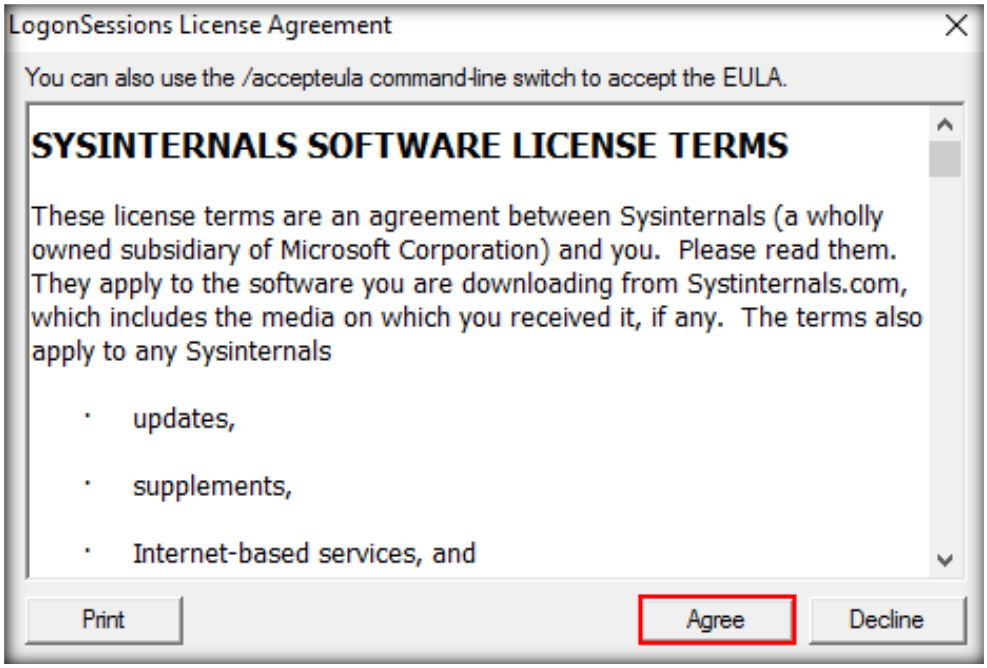


FIGURE 1.8: LogonSessions License Agreement window

11. The utility will now list the **currently active logon sessions**.

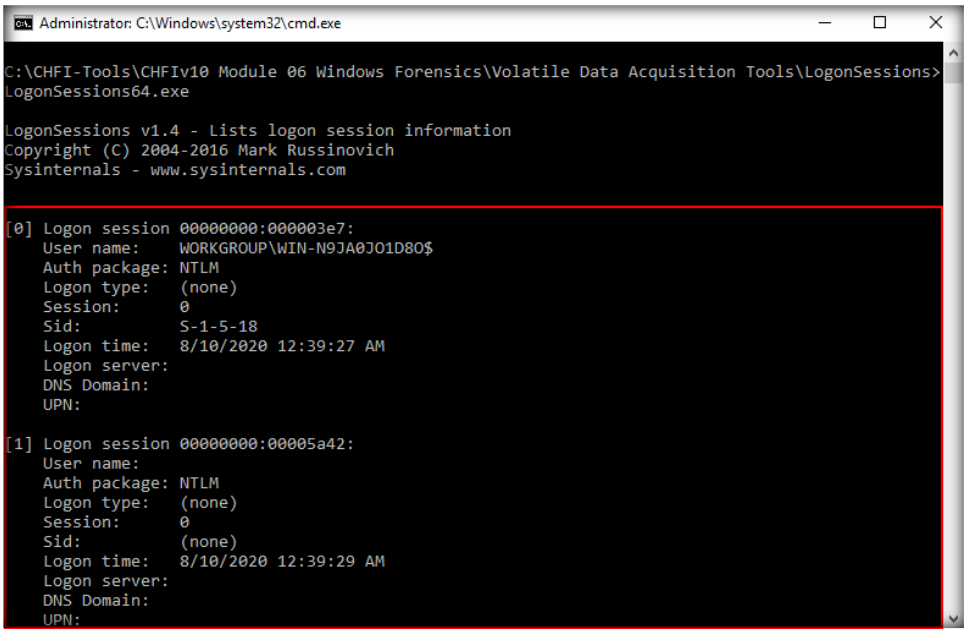


FIGURE 1.9: Result of LogonSessions64.exe

12. To list the **logon sessions**, their **types,** and the **processes** running under them, use the **-p** switch in combination with **LogonSessions64.exe**. All the running processes associated with each session will be displayed under the **UPN** section as shown in the screenshot:

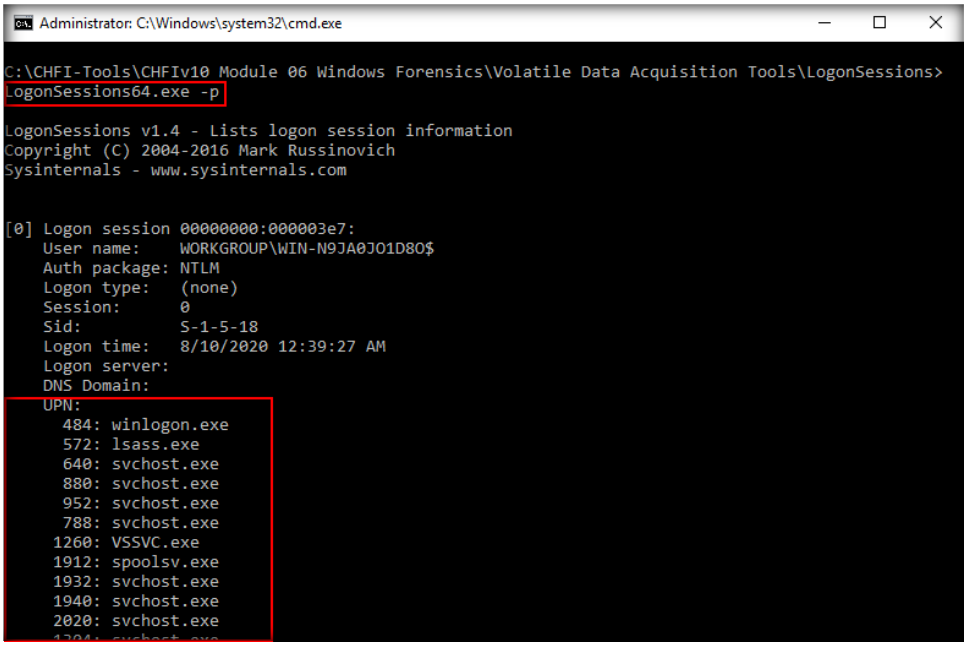


FIGURE 1.10: Result of LogonSessions with the -p switch

13. Close the **Command Prompt**.

TASK 2

Gather Information About Shared Files

14. Now, we will use the **net file** utility to retrieve information pertaining to all the open shared files with respect to **Windows Server 2016** virtual machine. Before you begin this task, switch to **Windows 10** virtual machine, and open **Z:**.
15. Switch back to **Windows Server 2016** virtual machine. To run the **net file** command, launch command prompt by right clicking the **Start** button and then clicking **Command Prompt (Admin)**. In the **Command Prompt** window, type **net file** and press **Enter**.

FIGURE 1.11: Execute the net file command on Command Prompt

16. The **net file** command will now display the **Path** of the shared folder from the local machine which is accessed by other machines (here, **Windows 10** virtual machine); it also displays the **Username** of the user account that is remotely accessing the shared folder along with the **ID** and **Locks** that are associated with the shared folder as shown in the screenshot:

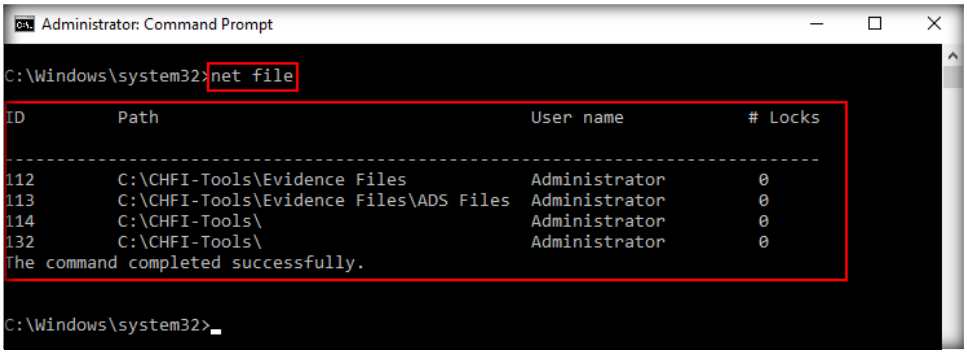


FIGURE 1.12: Result of the net file command

Note: The output of the command shown in the screenshot might vary in your lab environment.

17. For further analysis, you can use **NetworkOpenedFiles** to obtain more information regarding the shared **network folder/drive**, such as **IP Address**, **Host Name**, and **Permissions**.

Note: NetworkOpenedFiles is a utility for Windows that lists all the files that are currently open on the host machine (only those files that have been opened via remote login will be displayed). With the help of this tool, an investigator can identify suspicious files that are currently open on a host machine and have been opened via remote login.

18. Before running **NetworkOpenedFiles**, switch to **Windows 10** virtual machine and navigate to **Z:\Evidence Files** to find the **Infected.pdf** file. Open the file **Infected.pdf**.

19. Now, to run **NetworkOpenedFiles**, switch back to **Windows Server 2016** virtual machine, navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Volatile Data Acquisition Tools\NetworkOpenedFiles**, and double-click **NetworkOpenedFiles.exe**

20. The **NetworkOpenedFiles** window appears, displaying the shared folders and files that have been accessed, i.e., **CHFI-Tools**, **Evidence Files**, and **Infected.pdf**, along with information such as the **IP Address** of the computer on which they have been remotely opened, the **User** (name of the user account) accessing the folders and file(s), and the **Host Name** of the computer from which the listed folders and file(s) have been opened (in this case, the **Host Name** section would reflect **Windows 10** virtual machine's name). Double-click **C:\CHFI-Tools\Evidence Files\Infected.pdf**.

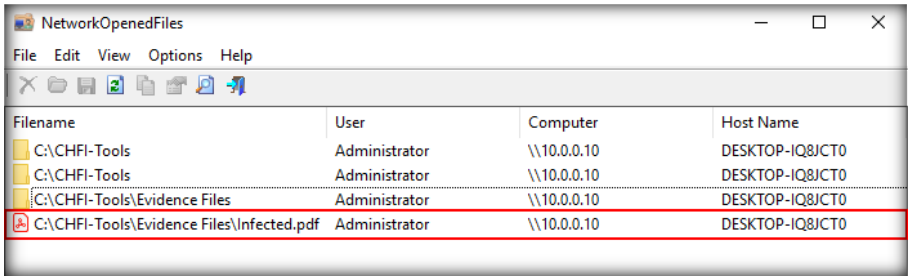


FIGURE 1.13: NetworkOpenedFiles window

Note: The tool displays the folders **CHFI-Tools** and **Evidence Files** and the file **Infected.pdf** because they have been opened remotely from another computer. The opened files and folders displayed in your lab might vary from the ones shown in the screenshot.

21. A **Properties** window will now open, displaying all the details pertaining to the file **Infected.pdf**.

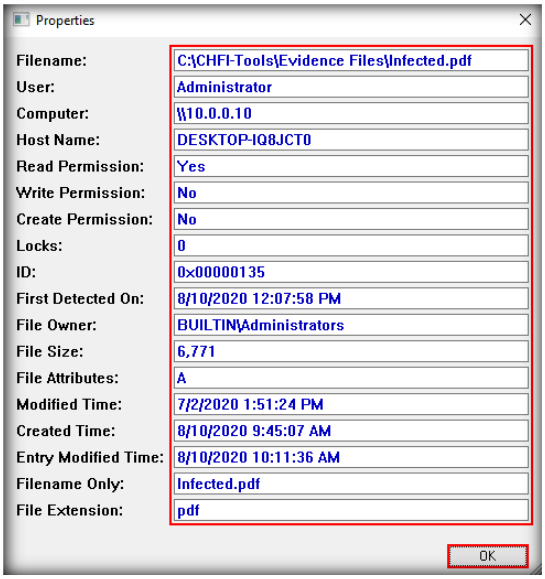


FIGURE 1.14: Properties window for Infected.pdf

Note: For the demonstrative purpose of this lab, the suspicious file has been represented as **Infected.pdf**. In the real time, the suspicious file may bear any name.

22. In this manner, you can also determine other files/folders that are currently open on the host machine.

Lab Analysis

Document the complete results of this lab.

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

Lab

2

Investigating Forensic Image of Windows RAM

Analyzing the RAM dump of a system helps investigators retrieve valuable evidence pertaining to a case of cyber-crime.

ICON KEY

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

Lab Scenario

An investment-banking firm discovered that confidential information pertaining to its trade secrets has been compromised. The firm consulted its expert forensic investigator to determine the source and cause of this data breach. The firm uses Windows systems to store all its data. The investigator needs to find out if there are any malicious processes running on the firm's restricted systems that store confidential data. To do so, the investigator must examine the RAM contents of the Windows machines to identify any malicious behavior.

To be an expert forensic investigator, you must know how to detect malicious processes running on a system by using the appropriate tools.

Lab Objectives

The objective of this lab is to analyze the RAM dump of a Windows machine by using the Redline utility and Volatility framework.

Lab Environment

This lab requires:

- A computer running **Windows Server 2016** virtual machine
- Administrative privileges to execute commands
- A web browser with internet access
- **Redline** installer located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Memory Forensics Tools\Redline**

- **Volatility** tool for Windows located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Memory Forensics Tools\Volatility**

Note: You can download the latest Windows version of **Redline** tool from the link https://www.fireeye.com/services/freeware/redline.html

You can download the latest Windows version of **Volatility** tool from the link https://www.volatilityfoundation.org/releases

If you are using the latest version of software for this lab, then the steps and screenshots demonstrated in the lab might differ.

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 30 minutes

Overview of the Lab

This lab familiarizes you with the examination of Windows memory dumps using the Redline utility and Volatility framework. The examination involves investigating and identifying attacks or malicious behavior that occurred on the target machine.

Lab Tasks

1. Login to the **Windows Server 2016** virtual machine.
2. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Memory Forensics Tools\Redline**, double-click **Redline-2.0.msi**, and follow the wizard-driven installation steps to install the application.

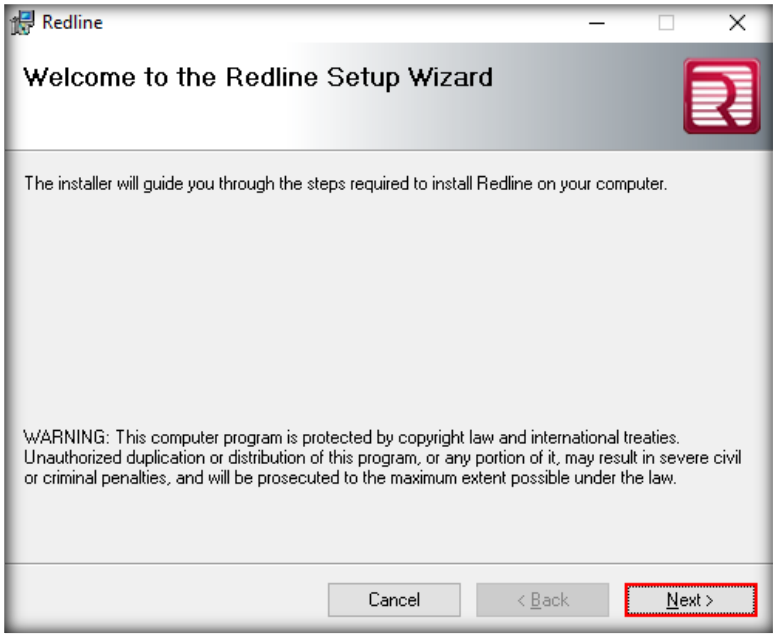


Figure 2.1: Redline Utility Setup Wizard

3. After completing the installation, click on the **Windows** icon at the lower-left corner of the screen and then on the **Redline** icon to launch the application.

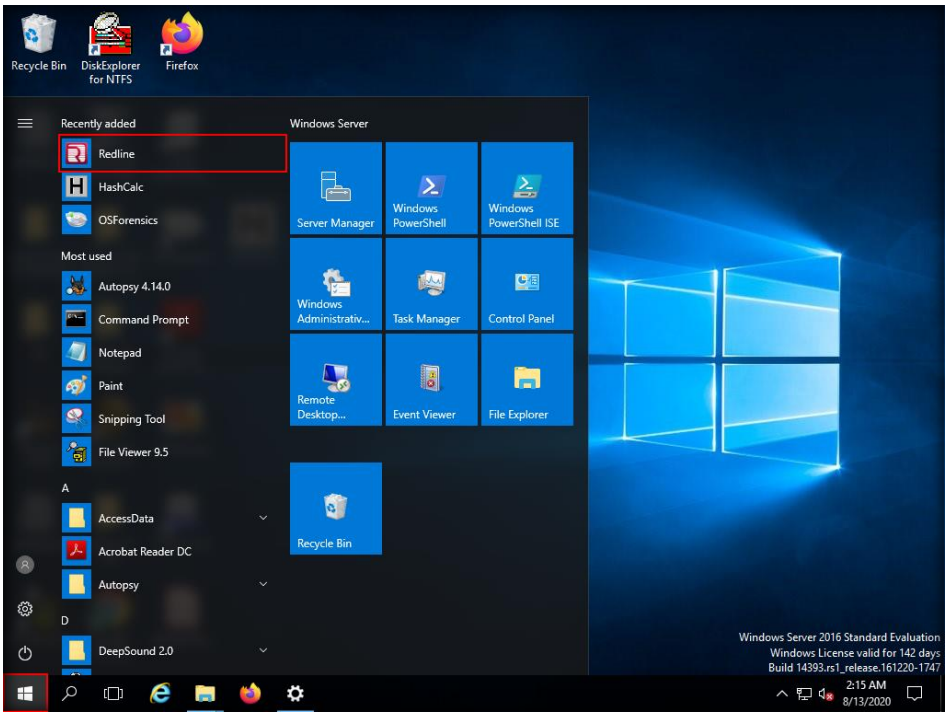


Figure 2.2: Redline utility icon in Windows Start Menu

TASK 1

Perform Malware Analysis Using Redline

4. The main window of the **Redline** tool appears. In this lab, we are going to examine a memory image. So, hover the mouse cursor over the **Analyze Data** section and click **From a Saved Memory File**.

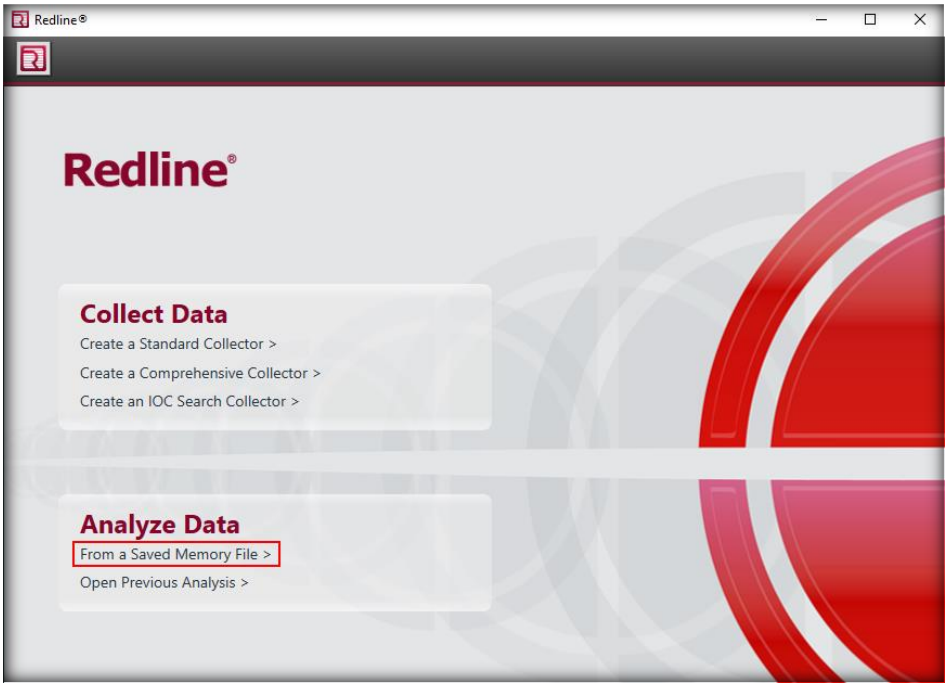


Figure 2.3: Click on From a Saved Memory File

5. The **Start your Analysis Session** window appears. Here, you need to specify the location of the memory image you wish to examine. So, click on the **Browse** button.

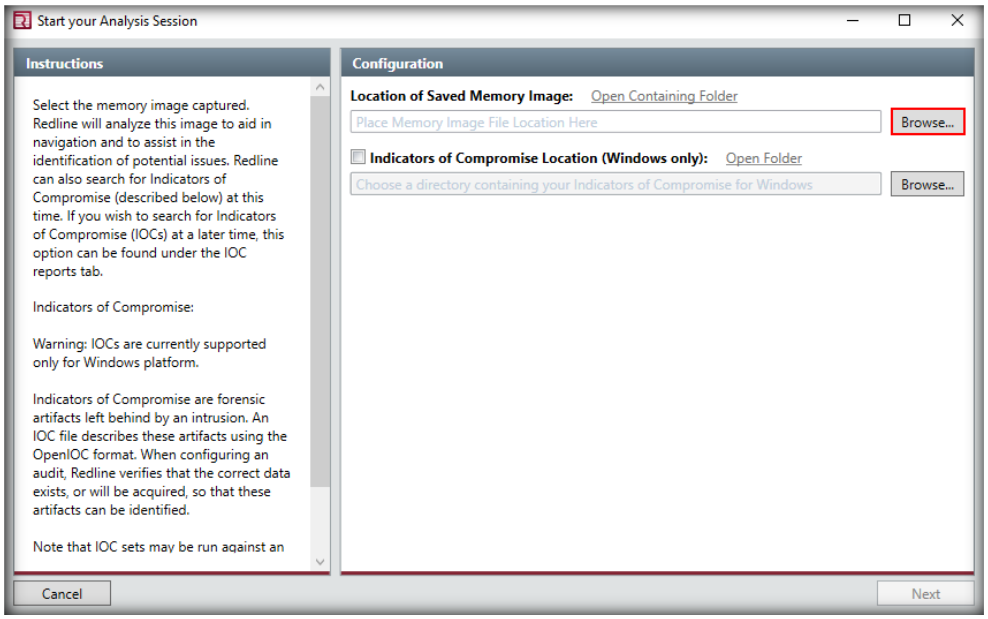


Figure 2.4: Select the Browse option to specify the image file location

6. The **Open** window appears. Navigate to **C:\CHFI-Tools\Evidence Files\Forensic Images**, choose **All Files** from the file-type dropdown list, select **Windows_RAM.mem**, and click **Open**.

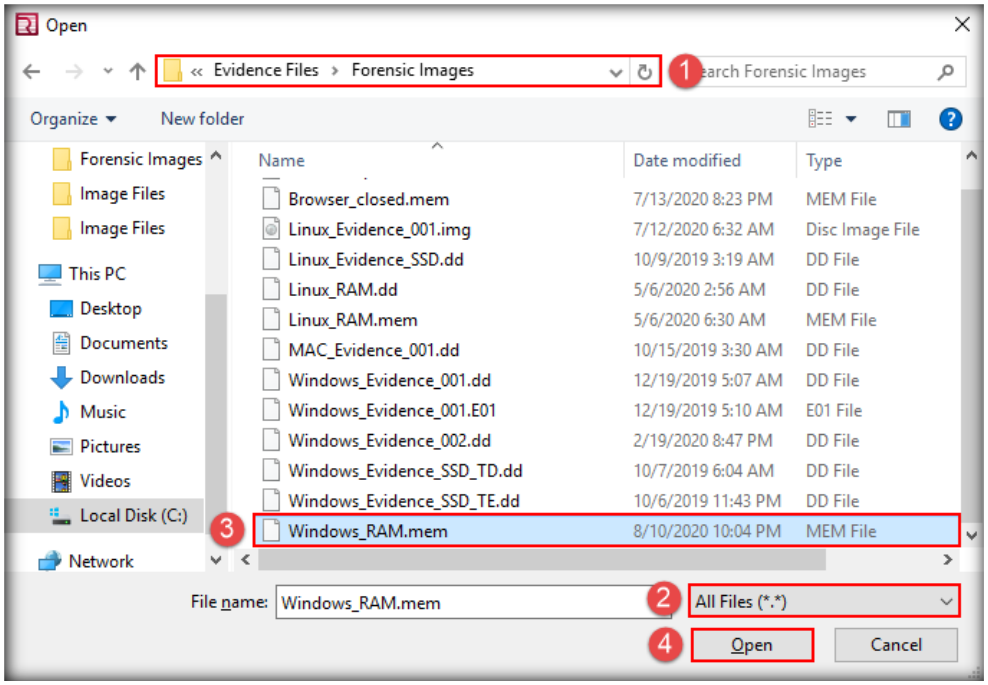


Figure 2.5: Select the memory image file

7. The file is now set for analysis. Click **Next**.

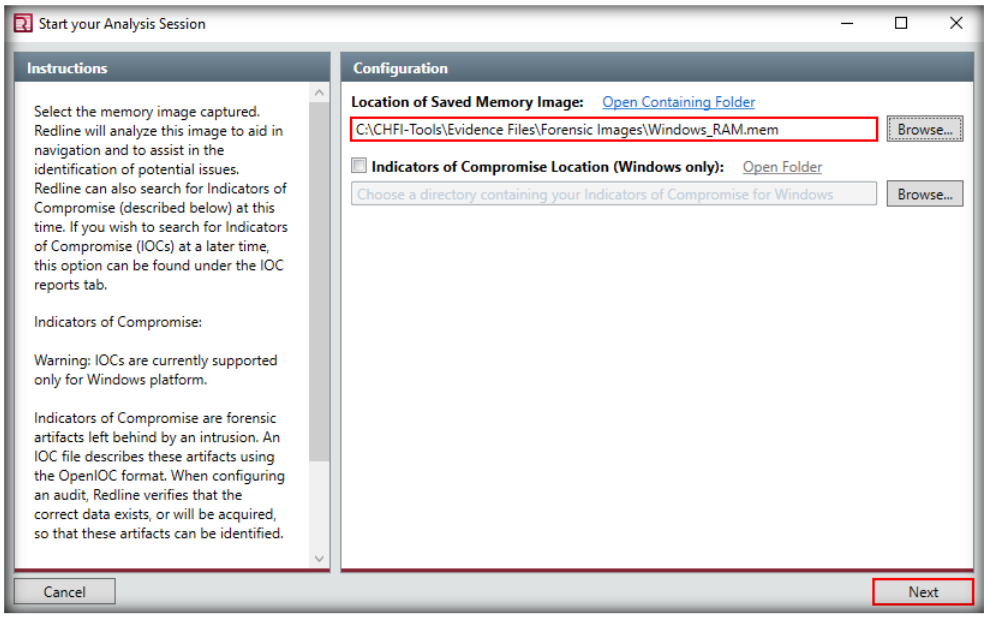


Figure 2.6: Memory image file selected

8. Now, you must specify a name to the analysis session and the location to which the session should be saved. In this lab, we are setting the name for the analysis session as **Windows Ram Analysis** and the location to the default, i.e., **C:\Users\Administrator\Documents**.

9. Click **OK** to proceed further.

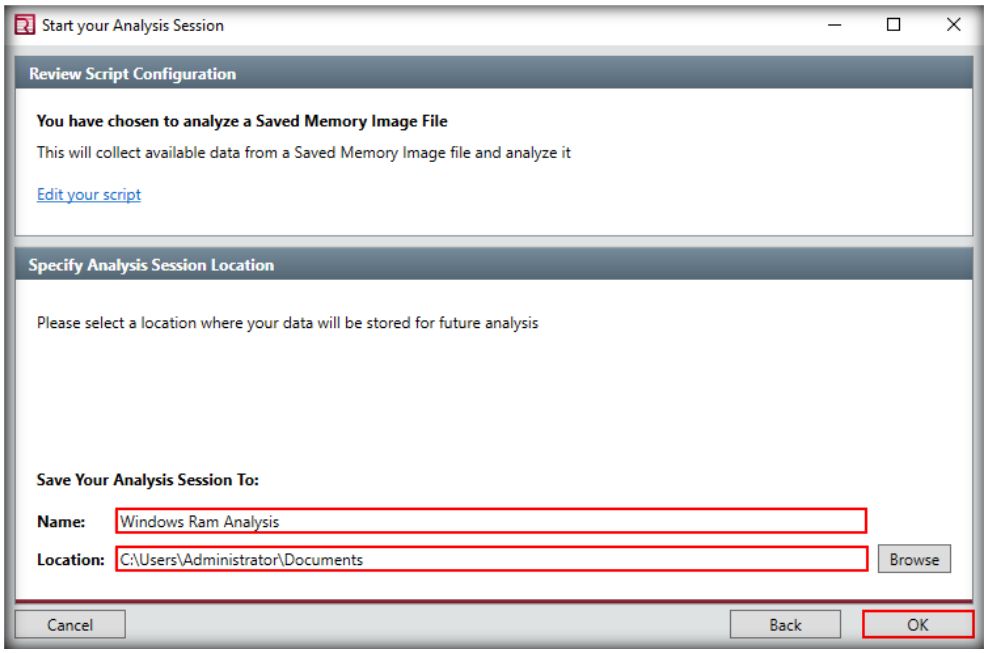


Figure 2.7: Specify Analysis Session Location section

10. **Redline** begins to analyze the image file and takes around **60 minutes** to complete the analysis.

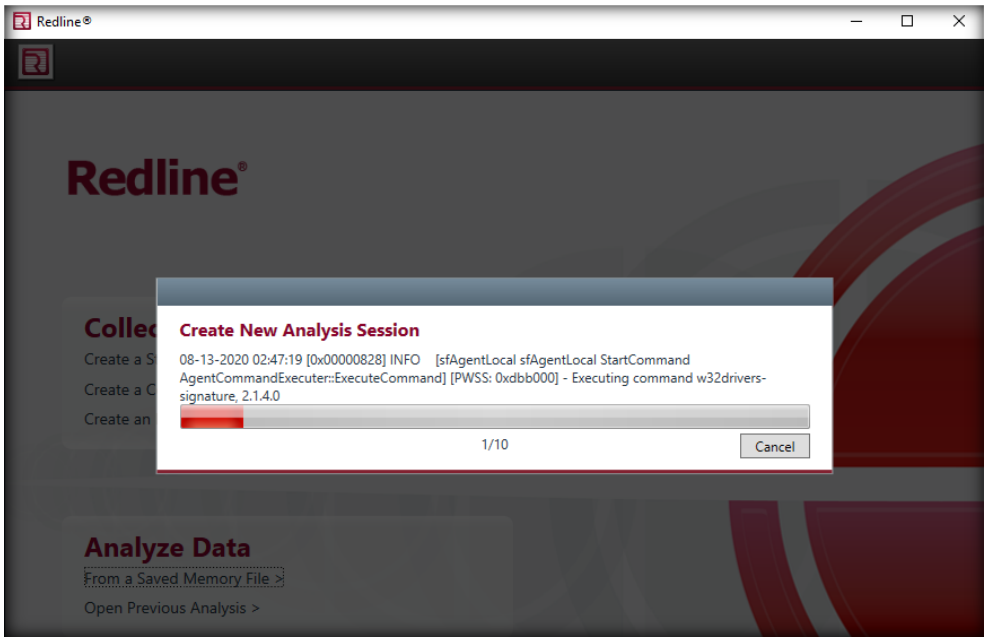


Fig 2.8: Analysis Session status window

11. After creating the analysis session, the following window appears. Double-click **Processes** in the left pane to view all the processes recorded on the image.

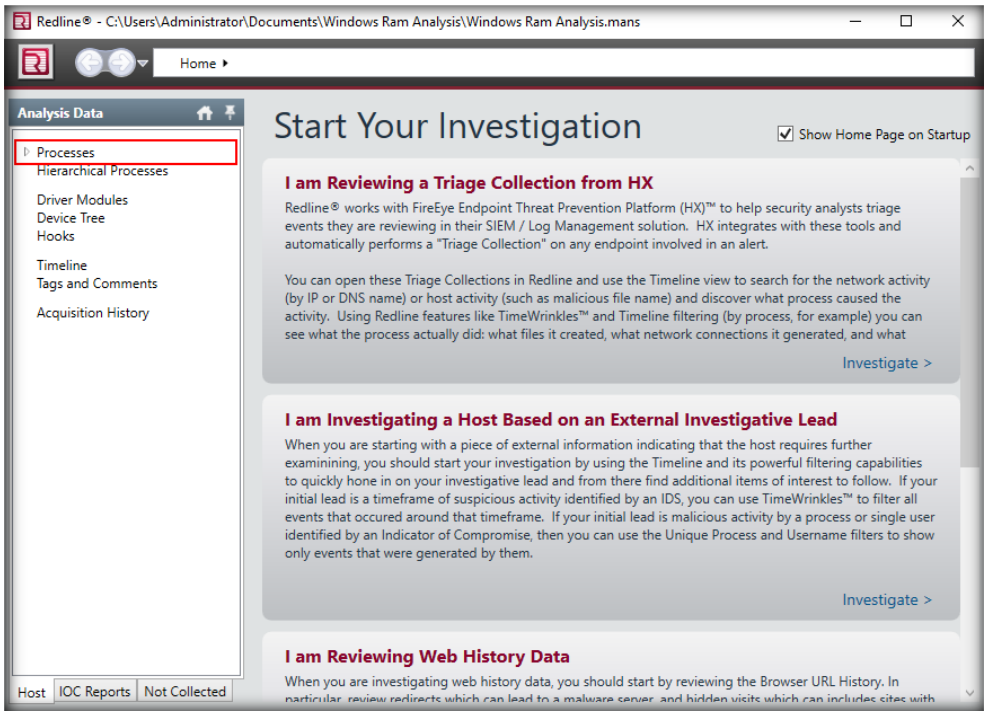


Figure 2.9: Redline tool

12. **Redline** displays all the processes recorded in the RAM as shown in the following screenshot:

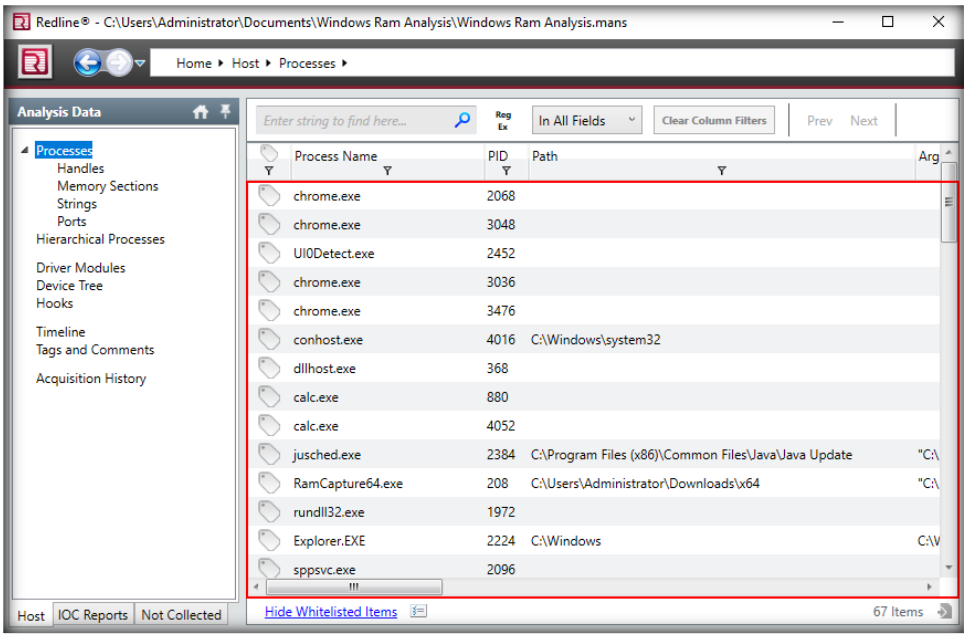


Figure 2.10: Displaying processes recorded in the RAM image file

13. Examining the processes can help investigators in determining if there are any suspicious processes running on the machine. However, if they fail to identify any such processes, they may examine the network connections on the evidence to identify any unusual activities.

14. Click **Ports** in the left pane. The **Ports** section appears in the right pane, displaying all the network connections and processes associated with them. Unpin the **Filters** section for clear visibility of the **Ports** section.

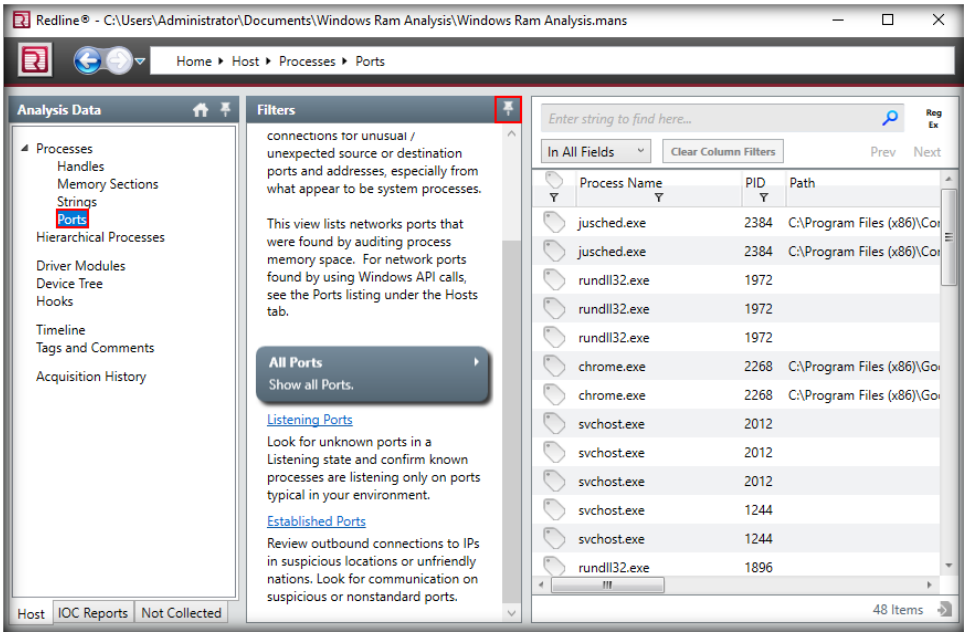


Figure 2.11: Select Ports

15. Examine the Ports section to find any anomalies in the network. You will observe that a process named **rundll32.exe** is connected to the remote IP **172.20.20.21** on port **4444** and the connection is **closed**.

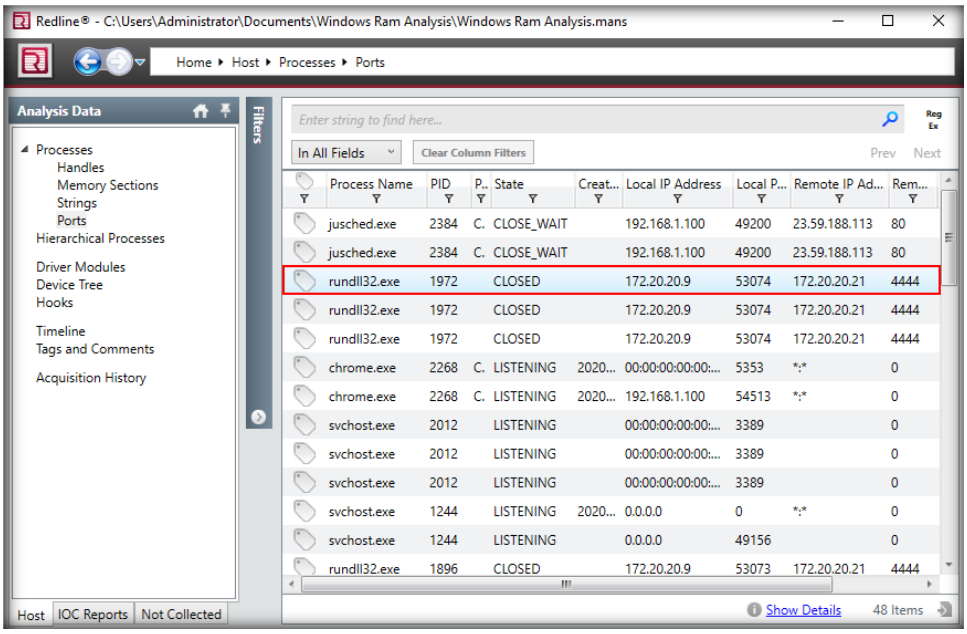


Figure 2.12: Detection of a malicious process

16. Scroll down the section to find any other anomalous connections. While scrolling down, you will observe that the same process **rundll32.exe** exists with PID **1896**, which is connected to IP **172.20.20.21** on port **4444**, but the connection state is **ESTABLISHED** and active, which is suspicious.

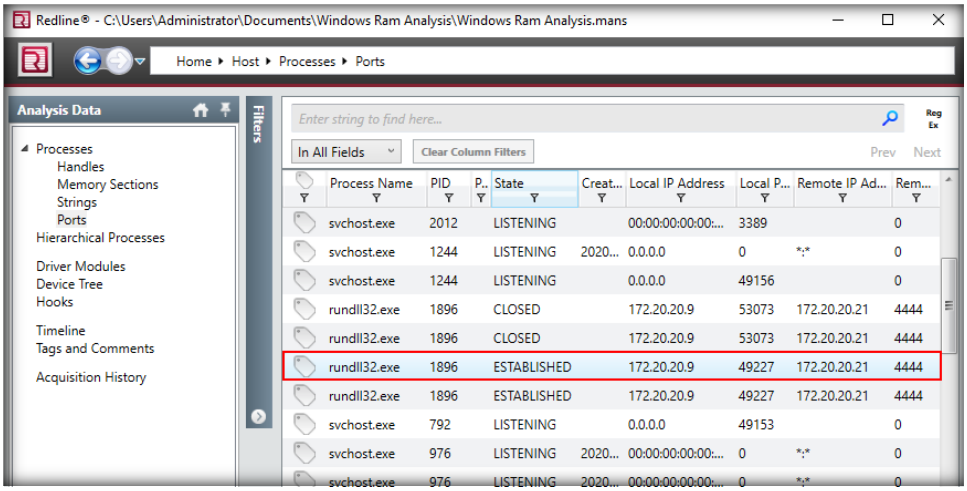


Figure 2.13: Malicious process rundll32.exe

17. For further analysis, we will go to the **Hierarchical Processes** section, where the parent and child processes are listed. Click **Hierarchical Processes** in the left pane.

18. The **Hierarchical Processes** section appears in the right pane, displaying the processes in a hierarchy, as shown in the following screenshot:

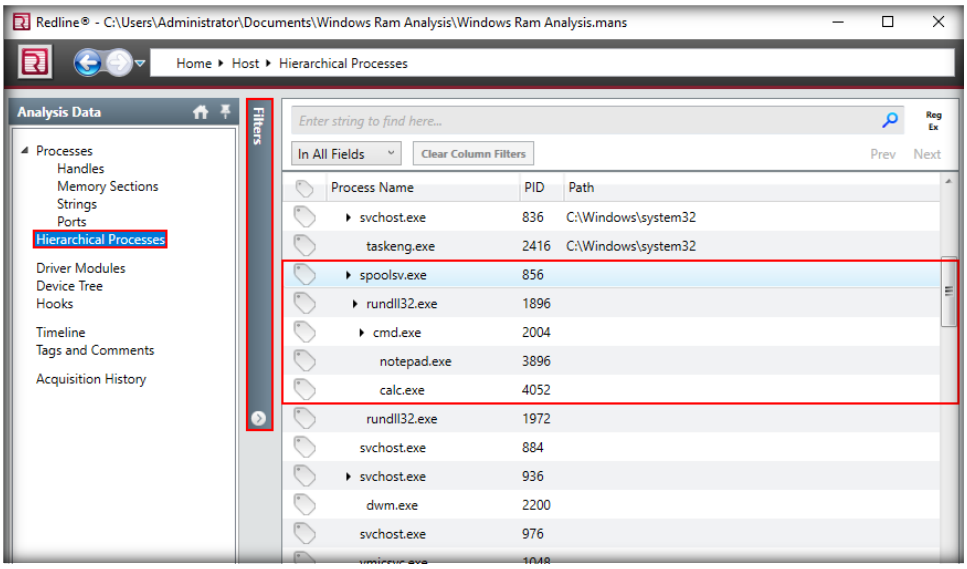


Figure 2.14: Hierarchical Processes tab

19. The screenshot above shows that the **rundll32.exe** process, which we identified in the earlier steps, has a parent process **spoolsv.exe** and child process **cmd.exe**. From the child process **cmd.exe**, two processes have been executed: **notepad.exe** and **calc.exe**.

20. The above finding indicates that the **spoolsv.exe** process was compromised, leading to the creation of child processes **rundll32.exe** and **cmd.exe**. The **notepad.exe** and **calc.exe** processes were later executed from **cmd.exe**, which corresponds to **Command Prompt**.

21. To view the timeline of the events recorded on the RAM, click **Timeline** in the left pane. This displays the timeline of activities, as shown in the following screenshot:

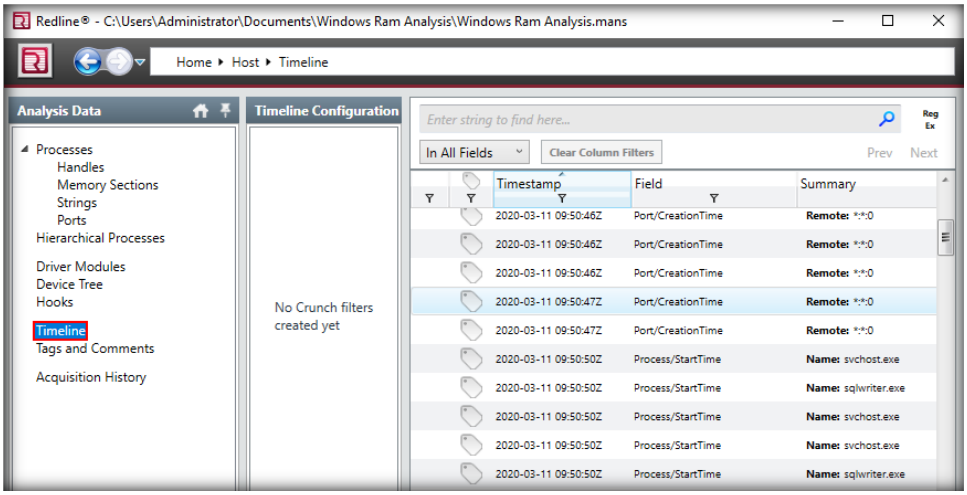


Figure 2.15: Select Timeline to view recorded events on RAM

TASK 2

Perform RAM Analysis Using Volatility Framework

22. Close the application.
23. Now, we will analyze the image using the **Volatility** framework.
24. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Memory Forensics Tools**, select **Volatility** folder, hold the **Shift** key, and right-click the selected folder.
25. Select **Open command window here** from the context menu.

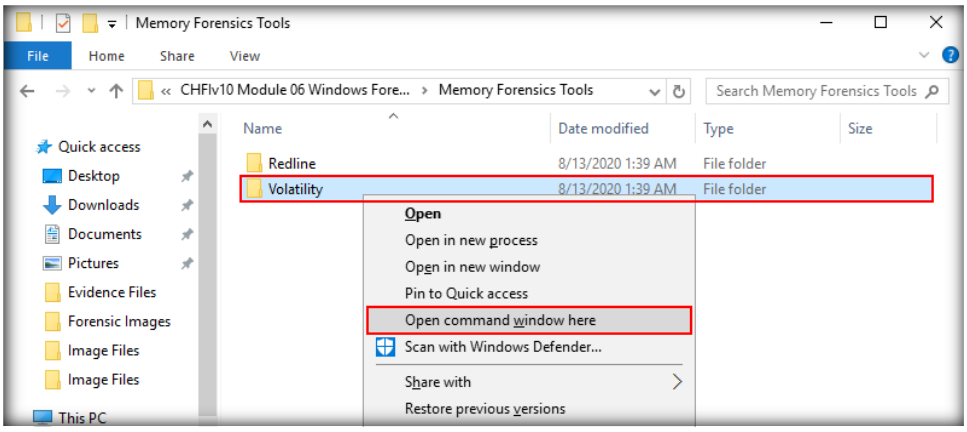


Figure 2.16: Open Volatility framework using Command Prompt

26. **Command Prompt** appears with the path set to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Memory Forensics Tools\Volatility**. In this lab, we will be analyzing **Windows_RAM.mem** located at **C:\CHFI-Tools\Evidence Files\Forensic Images**.
27. Our first task is to fetch information about the image. Volatility's **imageinfo** plugin helps in retrieving this information. So, type **volatility_2.6_win64_standalone.exe -f "C:\CHFI-Tools\Evidence Files\Forensic Images\Windows_RAM.mem" imageinfo** in **Command Prompt** and press **Enter**.
28. Volatility analyses the image file and displays its information as shown in the following screenshot:

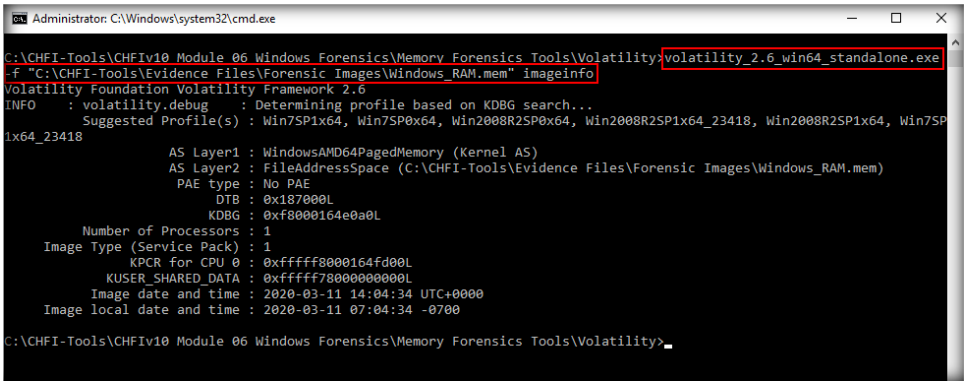


Figure 2.17: Analyzing an image file for profile suggestions

29. Alternatively, you can use the **kdbgscan** plugin to scan the kernel debugger and list the suggested profiles. To use the plugin, type

volatility_2.6_win64_standalone.exe -f "C:\CHFI-Tools\Evidence Files\Forensic Images\Windows_RAM.mem" kdbgscan in **Command Prompt** and press **Enter**. Volatility analyzes the kdbg header and lists the possible profiles, as shown in the screenshot:

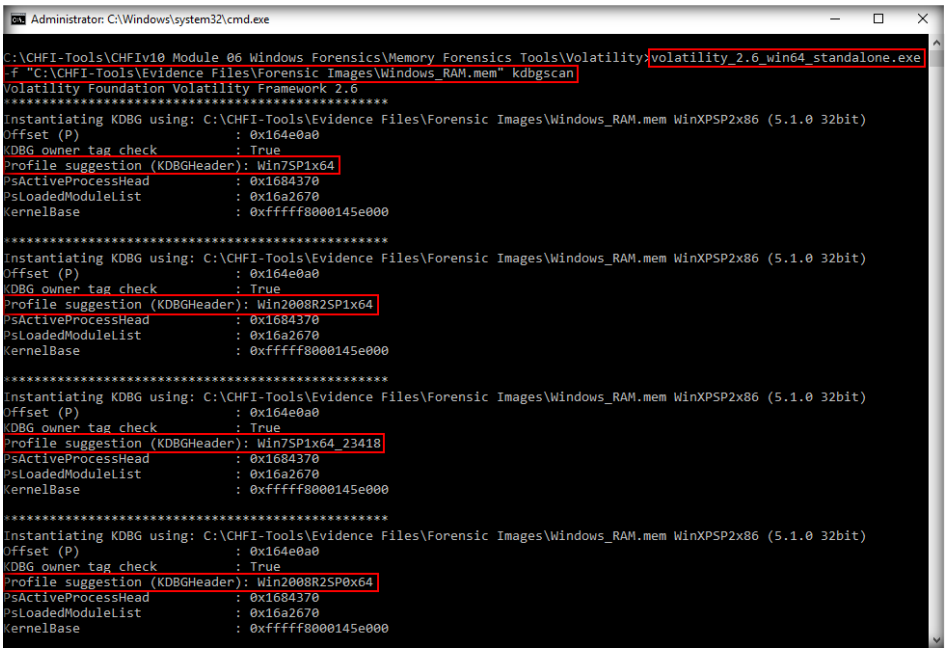


Figure 2.18: Identifying the profile of the system

30. Volatility displays a list of suggested profiles. We must choose a profile from the list that could be a close match to the profile on RAM. In this lab, we are going to choose **Win2008R2SP0x64**.

31. Since we found a suitable profile, we will now view the network connections associated with the machine that are recorded in the image. To view network connections, type the command **volatility_2.6_win64_standalone.exe -f "C:\CHFI-Tools\Evidence Files\Forensic Images\Windows_RAM.mem" --profile=Win2008R2SP0x64 netscan** in **Command Prompt** and press **Enter**.

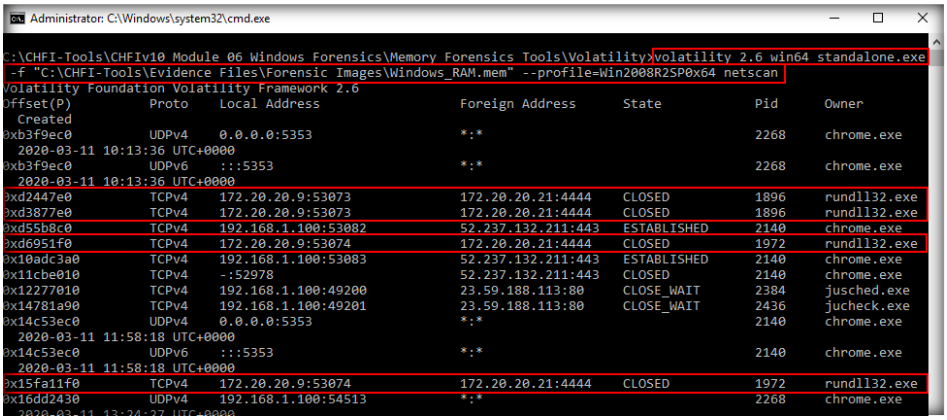


Figure 2.19: Viewing network connections

32. The screenshot above shows that a process named **rundll32.exe** has a network connection established with the suspect machine, but the connection has been closed. The PIDs for the process are 1972 and 1896. We will now see if there is any active connection with the same process.

33. Scroll down the **Command Prompt** window to detect any active connection with **rundll32.exe**. We can observe an active connection by **rundll32.exe** with PID 1896, as shown in the following screenshot:

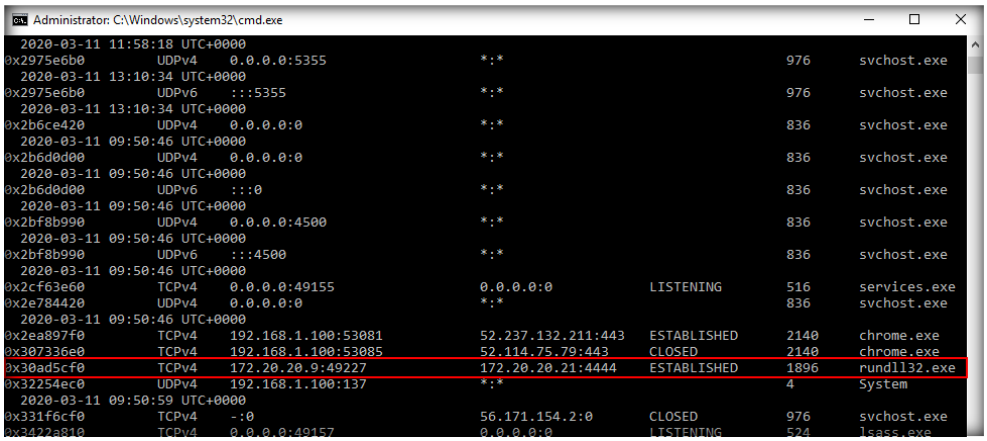


Figure 2.20: Active connection identified

34. From the above screenshot, it can be inferred that the **rundll32.exe** process has been initiated multiple times. While the connection pertaining to the PID **1972** is closed, the connection with PID **1896** is active. Based on this information, we will conduct further investigation.

35. You can observe the same process in the process list by issuing the following command: **volatility_2.6_win64_standalone.exe -f "C:\CHFI-Tools\Evidence Files\Forensic Images\Windows_RAM.mem" --profile=Win2008R2SP0x64 pslist**.

36. Volatility lists all the processes as shown in the following screenshot:

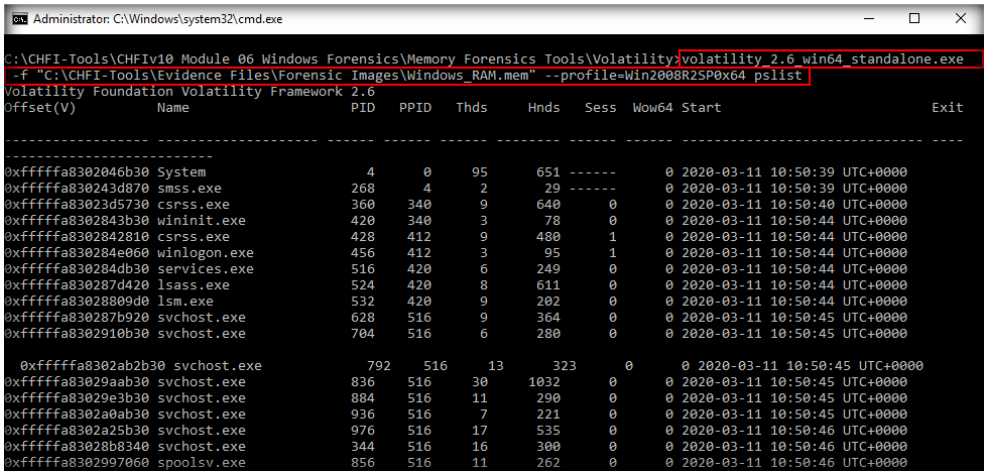


Figure 2.21: View running processes

37. Scroll down the list to find the **rundll32.exe** process, which was observed in the network connections listed earlier.

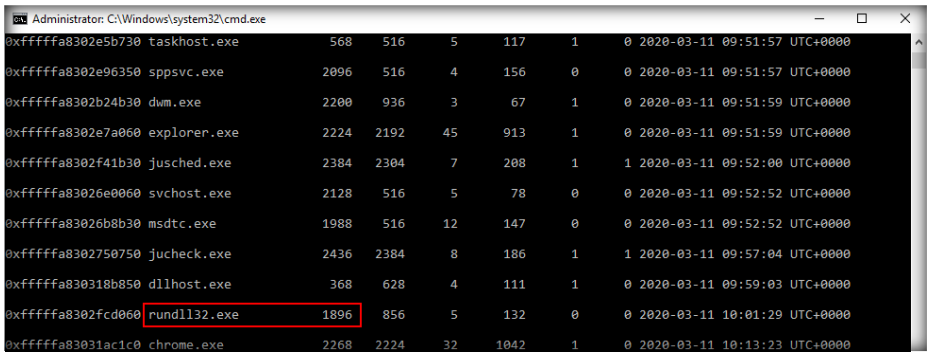


Figure 2.22: rundll32.exe process in pslist

38. Based on the process, you can now list the files and registries used by it. To view the files used by the process, type **volatility_2.6_win64_standalone.exe -f "C:\CHFI-Tools\Evidence Files\Forensic Images\Windows_RAM.mem" --profile=Win2008R2SP0x64 handles -p 1896 -t file** and press **Enter**.

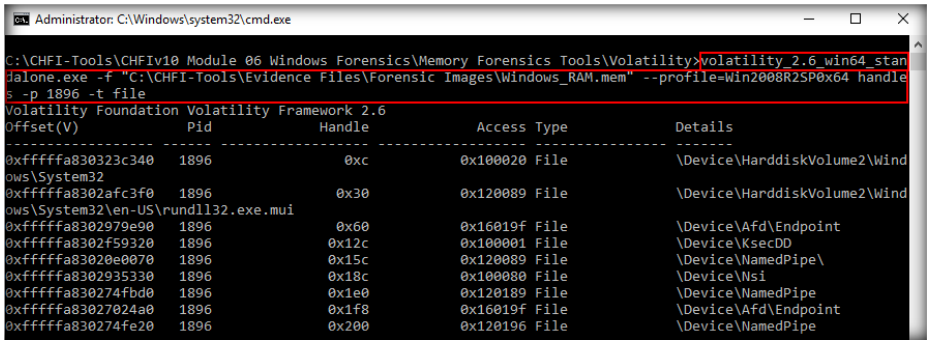


Figure 2.23: View files opened by a process

39. To view the registries associated with the process, type **volatility_2.6_win64_standalone.exe -f "C:\CHFI-Tools\Evidence Files\Forensic Images\Windows_RAM.mem" --profile=Win2008R2SP0x64 handles -p 1896 -t key** and press **Enter**.

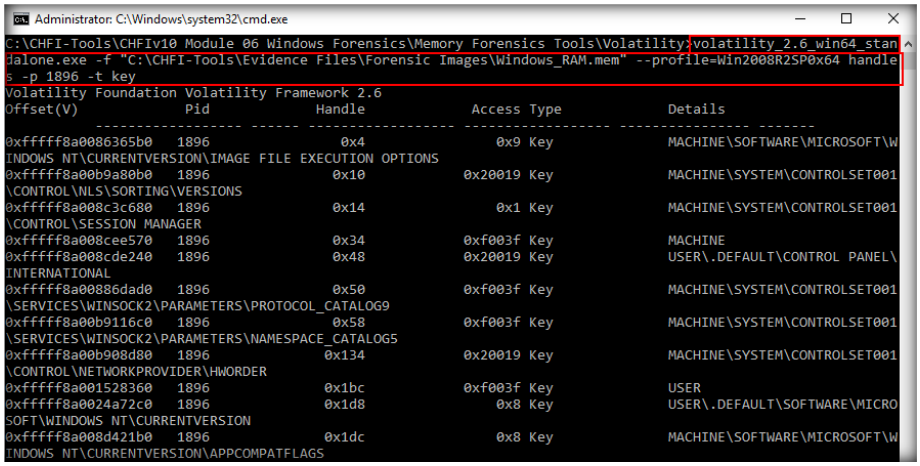


Figure 2.24: View registries opened by a process

40. To list the process tree to which the **rundll32.exe** process belongs, enter **volatility_2.6_win64_standalone.exe -f "C:\CHFI-Tools\Evidence Files\Forensic Images\Windows_RAM.mem" --profile=Win2008R2SP0x64 pstree**.

41. This lists the process tree pertaining to **rundll32.exe**, as shown in the following screenshot:

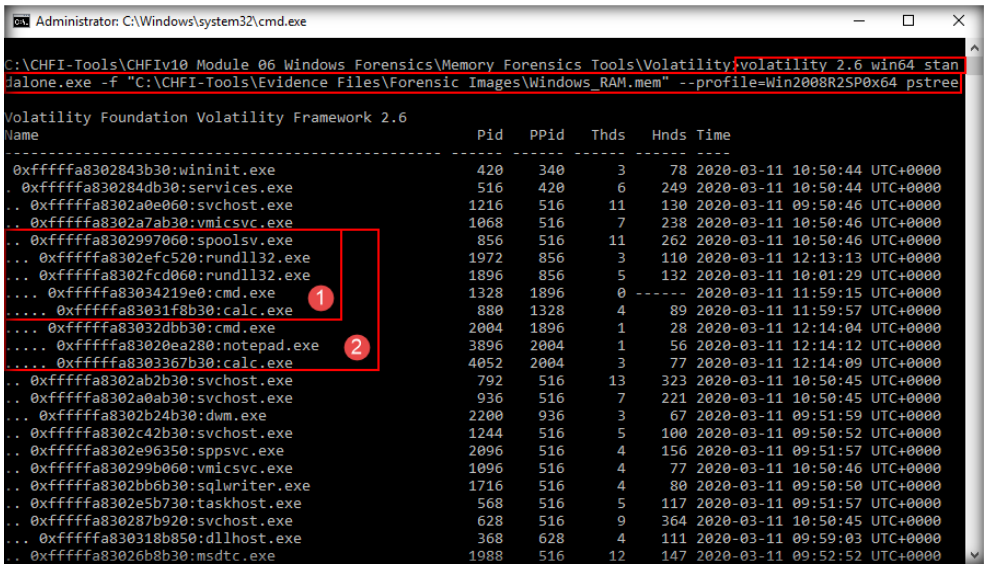


Figure 2.25: Child processes of rundll32.exe

42. In the above screenshot, **1** denotes the first process flow and **2** denotes the second.

43. In 1, we can observe the process flow as **spoolsv.exe → rundll32.exe → rundll32.exe → cmd.exe → calc.exe**. The first **rundll32.exe** process (PID: **1972**) is closed, while the second **rundll32.exe** process (PID: **1896**) begins later, with the following process flow: **spoolsv.exe → rundll32.exe → cmd.exe →** (**notepad.exe** and **calc.exe** both originated from **cmd.exe**).

44. This indicates that the **rundll32.exe** process originated from the **spoolsv.exe** process and, in turn, created a child process named cmd.exe, from which calculator and notepad were initiated.

45. In this manner, we can use Volatility to examine RAM images. Apart from the plugins demonstrated in the above steps, you can use plugins such as malfind, arp, ifconfig, and lsof to gather additional information.

TASK 3

Examine RAM Dump Using Strings

46. RAM images can also be examined using the **strings** command-line utility in Ubuntu. This utility helps you obtain information such as files/directories located on the machine and URLs and IPs stored on the disk, which can help an investigator in identifying valuable information.

47. Login to the **Ubuntu Forensics** virtual machine.

48. Click on the **Files** icon in the **Launcher** panel to launch the **File Manager**.

FIGURE 2.26: Launch File Manager

49. The **File Manager** window appears, pointing to the **Home** directory. Click on the bookmarked **chfi-tools on 10.0.0.16** directory.

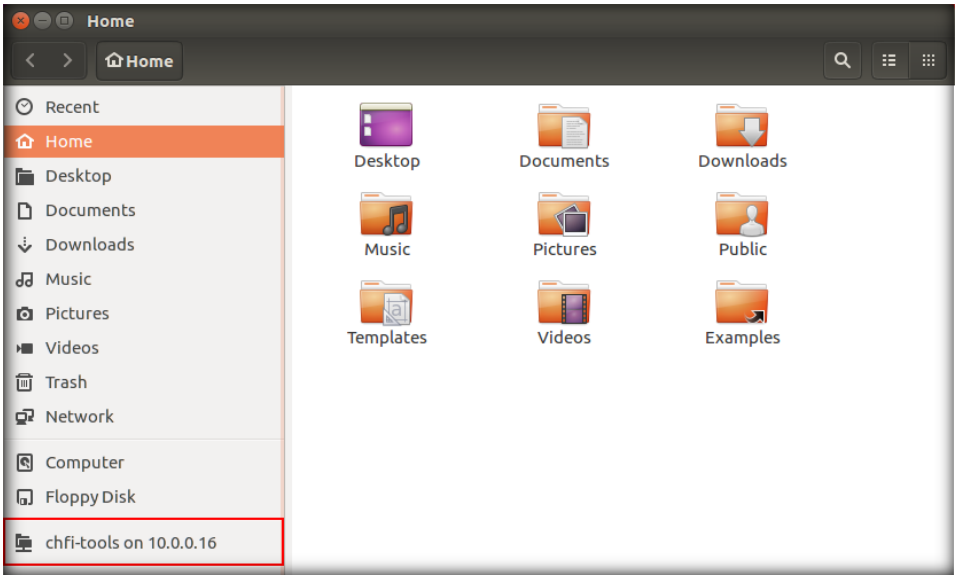


FIGURE 2.27: Enter CHFI-Tools directory

50. The **CHFI-Tools** directory appears in the window, as shown in the following screenshot:

FIGURE 2.28: View CHFI-Tools

51. Now, go to **Evidence Files → Forensic Images**. You will be able to view the **Windows_RAM.mem** file in the images. **Copy** this file.

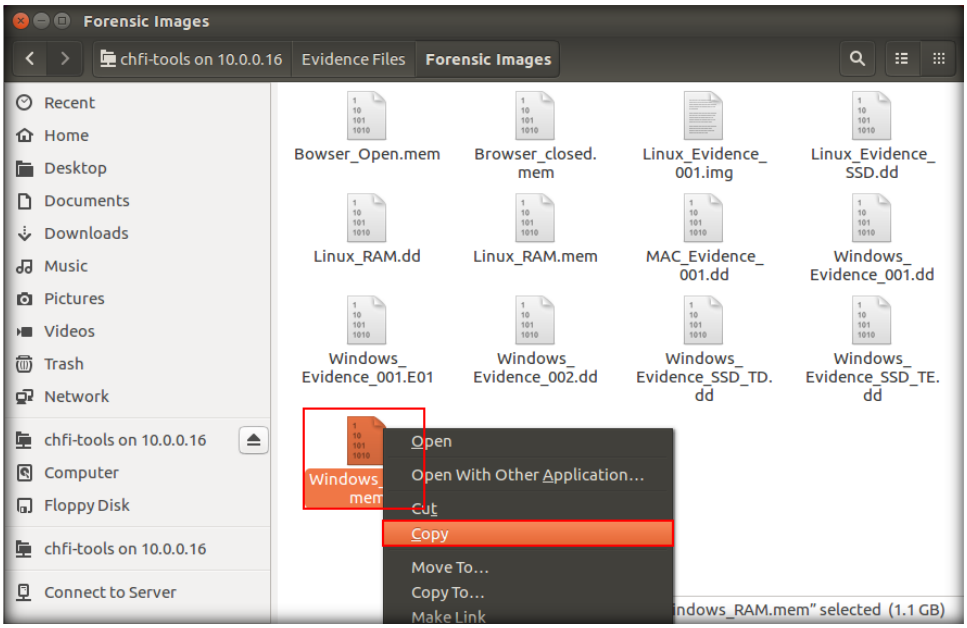


Figure 2.29: Copying the RAM image file

52. Navigate to the **Home** directory and **paste** the file.

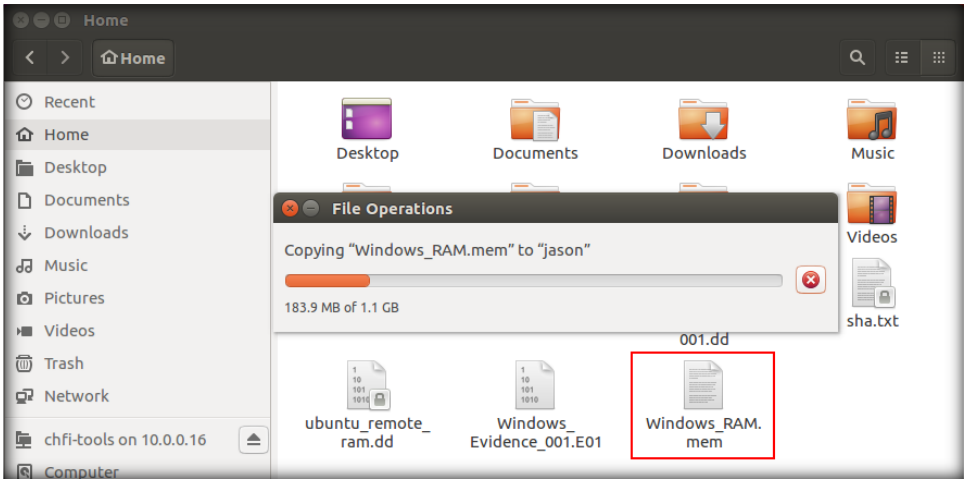


Figure 2.30: Image file copy status

53. Now, click on the **Terminal** icon from the launcher panel to launch the command-line terminal.

FIGURE 2.31: Click the Terminal icon

54. The command-line terminal launches. You require root privileges in the terminal to install an application.
55. So, type **sudo su** and press **Enter**. You will be prompted to enter the password. Type **toor** as the password and press **Enter**.
56. Now, you will enter the **root** terminal, as shown in the following screenshot:

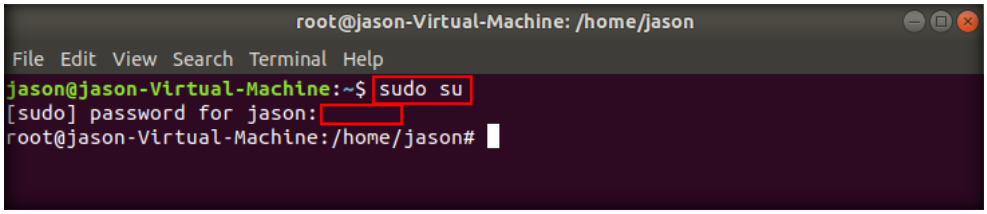


FIGURE 2.32: Run as super user

57. To list all the **files** and **directories**, type **strings Windows_RAM.mem | grep -i "^[a-z]:\\\\" | sort | uniq** and press **Enter**.
58. This displays all the files and directories stored on the suspect machine, as shown in the following screenshot:

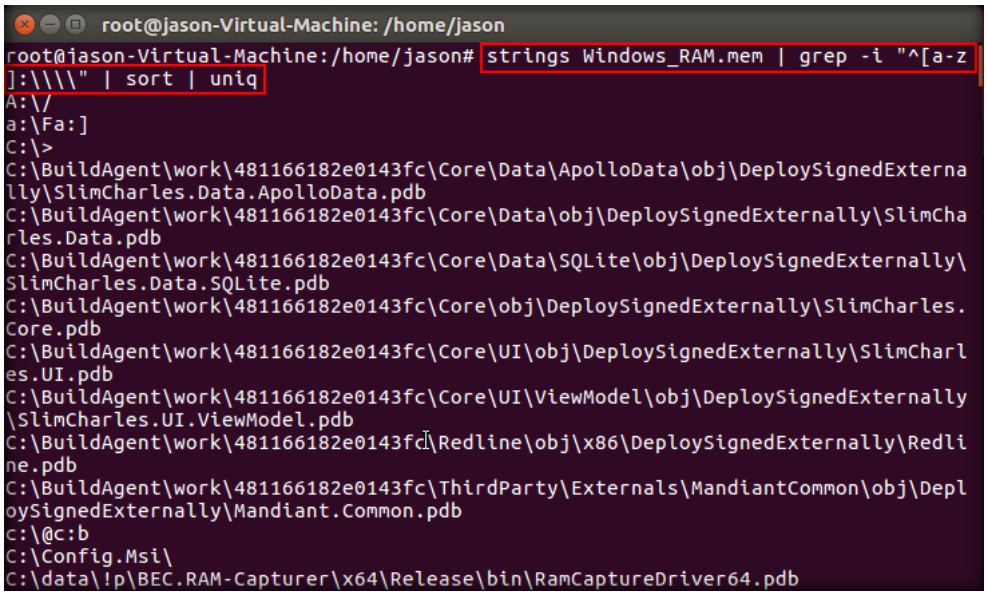


FIGURE 2.33: Output of the strings command, displaying directories/path

59. We must examine the output to see if there are any suspicious files found on the machine.
60. To extract all the **URLs** stored in the suspect machine, enter the command: **strings Windows_RAM.mem | egrep "^https?://" | sort | uniq**.

61. The command lists all the visited **URLs** in its output, as shown in the screenshot:

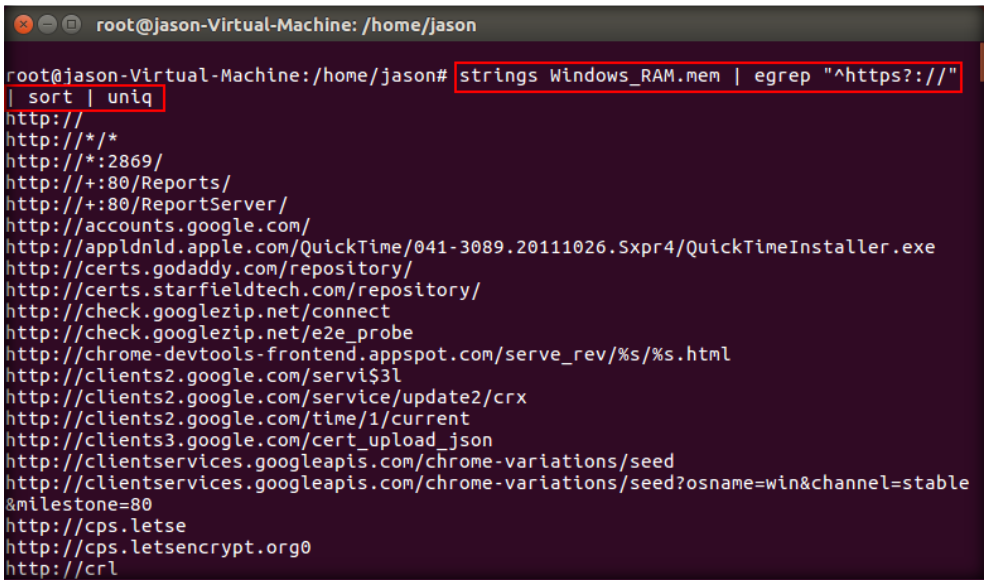


FIGURE 2.34: Output of the strings command, displaying visited URLs

62. We must examine the output to determine the presence of any suspicious URLs on the machine.

63. Examining the IP addresses can help us determine if there are any malicious connections associated with the machine. To view the IP addresses, type **strings Windows_RAM.mem | egrep -oE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'** and press **Enter**.

64. Strings command extracts all the entries associated with IP addresses and additional entries that have an IP address format, i.e., [octet.octet.octet.octet], as shown in the following screenshot:

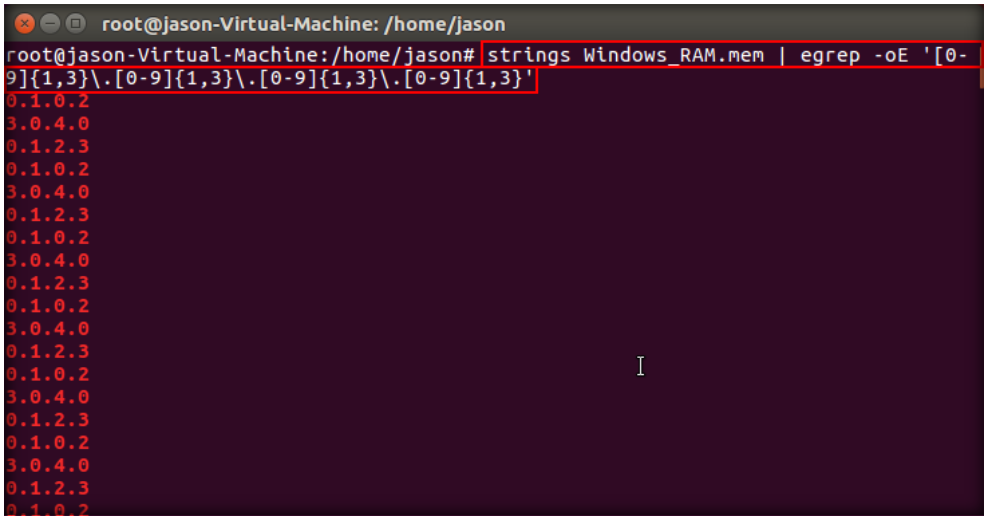


Figure 2.35: Output of the strings command, displaying entries with the IP address format

65. Scroll down the result to find any **legitimate IP address entries** recorded on the image.

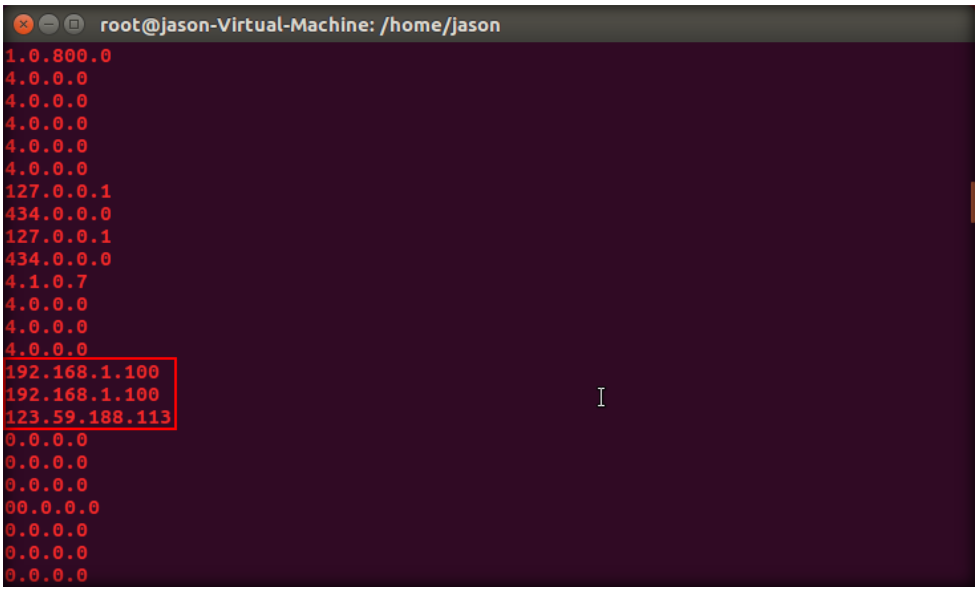


Figure 2.36: Output of the strings command, displaying recorded IP addresses

66. If you find an entry that might reveal information related to the investigation, you can use the strings command to view such information.

67. In this lab, we will use the IP address of the machine (in this case, **172.20.20.9**) to filter data containing this address. So, type **strings Windows_RAM.mem | egrep "172.20.20.9"** and press **Enter**. This examines the image to filter all the entries that contain the IP address as shown in the following screenshot:

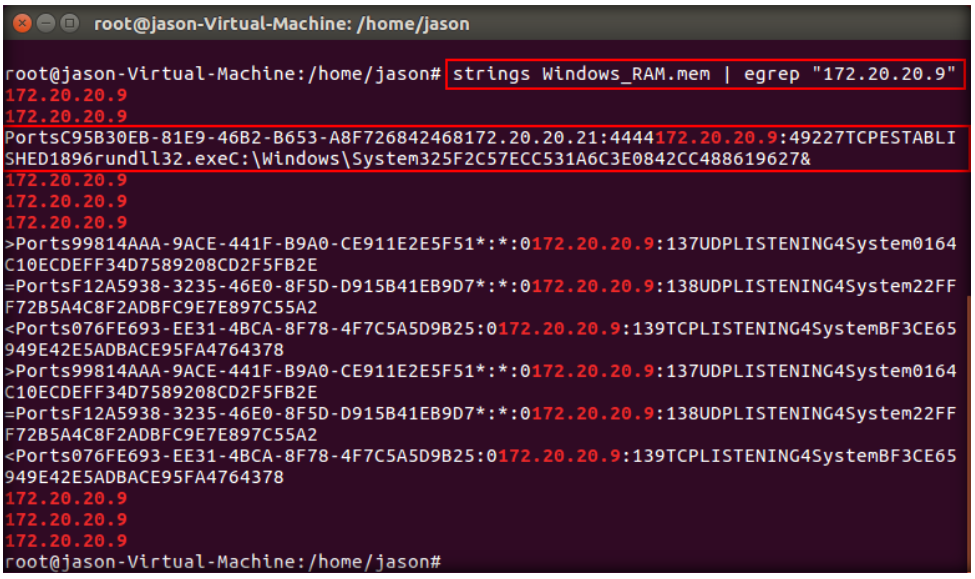


Figure 2.37: Viewing information associated with a specific IP address

68. Decoding the above entry reveals that a network connection has been established with the machine under investigation, i.e., **172.20.20.9** from **172.20.20.21** through port no. **4444**. The process associated with the connection is **rundll32.exe** located in **C:\Windows\System32**.
69. In this manner, RAM images act as important evidence for uncovering crucial data during investigation.

Lab Analysis

Document the complete results of this lab.

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

Lab

3

Examining Web Browser Artifacts

The history, cookies, and cache of a browser store important data pertaining to users' browsing activities. In the event of a cyber-crime, investigating the history, cookies, and cache of web browser(s) helps investigators retrieve valuable artifacts to solve the case.

ICON KEY

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

Lab Scenario

Stuart is an employee at a technology firm. His company witnessed a dip in his performance over a period and found his conduct at the workplace to be extremely suspicious. They consulted an expert forensic investigator to examine Stuart's workstation. As part of the investigation, the investigator must examine the caches, cookies, and browsing history stored in the browser(s) used on the suspect's workstation. Examining and analyzing caches, cookies, and browsing history can help an investigator gather valuable evidence to solve a case of cyber-crime.

To be an expert forensic investigator, you must know how to examine and analyze the browsing history, cookies, and caches stored in browsers.

Lab Objectives

The objective of this lab is to investigate and extract web-browser artifacts such as browsing history, cookies, and cache.

Lab Environment

This lab requires:

- A computer running **Windows Server 2016** virtual machine
- Administrative privileges to execute commands
- A web browser with internet access
- **ChromeCacheView** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Google Chrome\ChromeCacheView**

- **ChromeHistoryView** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Google Chrome\ChromeHistoryView**
- **ChromeCookiesView** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Google Chrome\ChromeCookiesView**
- **MZCacheView** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Mozilla Firefox\MZCacheView**
- **MZHistoryView** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Mozilla Firefox\MZHistoryView**
- **MZCookiesView** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Mozilla Firefox\MZCookiesCiew**

Note: You can download the latest version of **ChromeCacheView** from the link https://www.nirsoft.net/utils/chrome_cache_view.html

You can download the latest version of **ChromeHistoryView** from the link https://www.nirsoft.net/utils/chrome_history_view.html

You can download the latest version of **ChromeCookiesView** from the link https://www.nirsoft.net/utils/chrome_cookies_view.html

You can download the latest version of **MZCacheView** from the link https://www.nirsoft.net/utils/mozilla_cache_viewer.html

You can download the latest version of **MZHistoryView** from the link https://www.nirsoft.net/utils/mozilla_history_view.html

You can download the latest version of **MZCookiesView** from the link https://www.nirsoft.net/utils/mzcv.html

If you are using the latest version of software for this lab, then the steps and screenshots demonstrated in the lab might differ.

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 15 minutes

Overview of the Lab

This lab familiarizes you with the tools **ChromeCacheView**, **ChromeHistoryView**, **ChromeCookiesView**, **MZCacheView**, **MZHistoryView**, and **MZCookiesView** and helps you understand how to retrieve artifacts pertaining to cache files, browsing history, and cookies from **Google Chrome** and **Mozilla Firefox** browsers.

Lab Tasks

1. Login to the **Windows Server 2016** virtual machine.

Note: The artifacts that are retrieved from browser analysis will depend on the websites visited by a user. Therefore, the browsing history, cookies, and cache files that are retrieved might vary in each lab environment. For the purpose of lab simulation, before beginning the lab tasks, it is recommended that you randomly browse various websites on each of the browsers being investigated, i.e., **Google Chrome** and **Mozilla Firefox**. This will enable you to detect and retrieve the browsing artifacts corresponding to the websites you browsed/visited.

TASK 1

Retrieving Artifacts from Google Chrome

2. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Google Chrome**.
3. In the **Google Chrome** folder, you will find folders for three tools: **ChromeCacheView**, **ChromeHistoryView**, and **ChromeCookiesView**.
4. First, open the **ChromeCacheView** tool folder and launch **ChromeCacheView.exe**.

Note: If an **Open File - Security Warning** window appears, click **Run.**

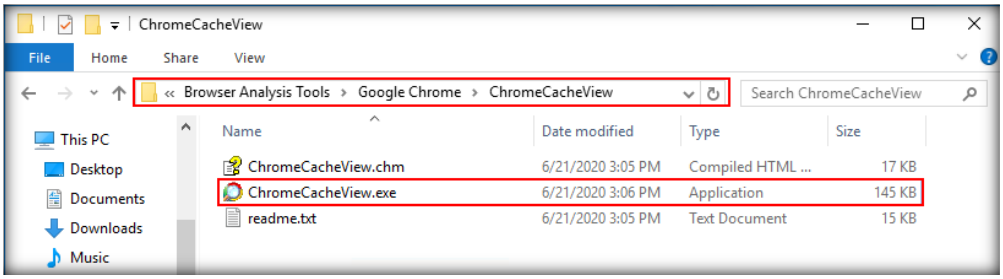


Figure 3.1: ChromeCacheView tool folder

5. The tool displays information such as **Filename**, **URL**, **Content Type**, **File Size**, **Last Accessed**, **Expire time**, **Server Time**, **Server Name**, **Server Response**, and **Server IP Address**.

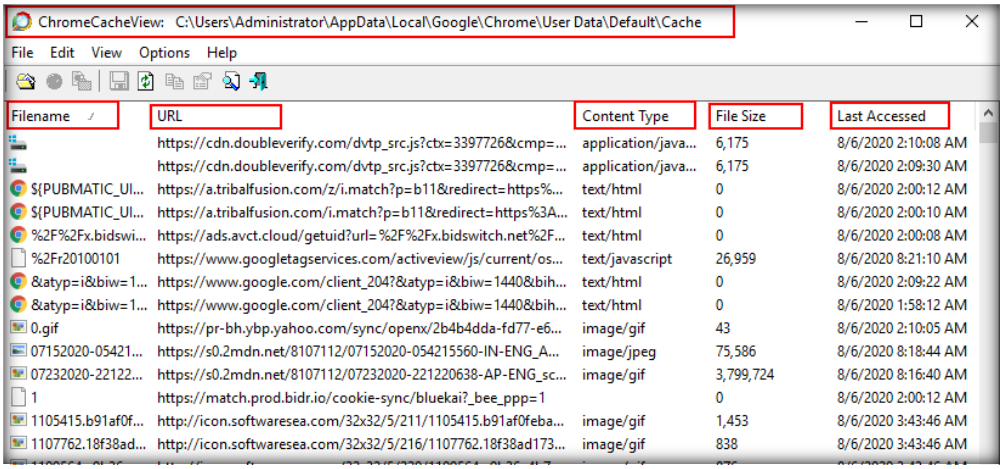


Figure 3.2: ChromeCacheView tool

6. Now, close the **ChromeCacheView** window, navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Google Chrome\ChromeCookiesView**, and launch **ChromeCookiesView.exe**.

Note: If an **Open File – Security Warning** window appears, click **Run.**

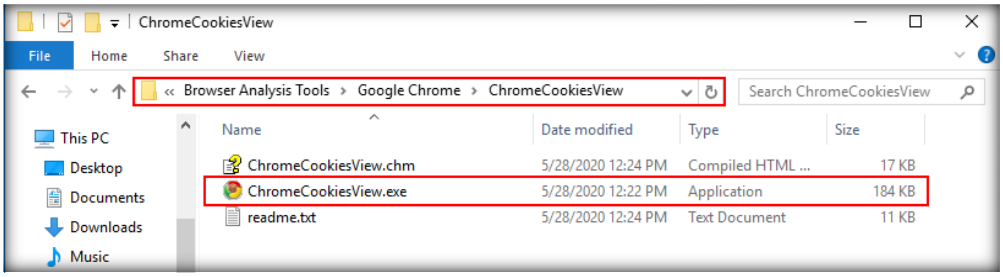


Figure 3.3: ChromeHistoryView tool folder

7. The tool displays information such as **Host Name**, **Path**, **Name**, **Value**, **Secure**, **HTTP Only**, **Last Accessed**, etc. for each visited web page.

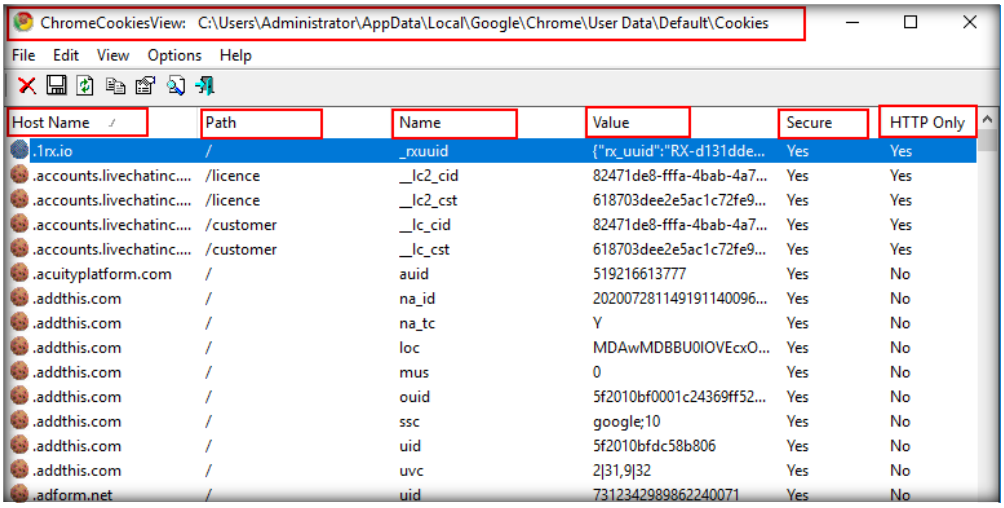


Figure 3.4: ChromeHistoryView tool

8. Now, close the **ChromeCookiesView** window, navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Google Chrome\ChromeHistoryView**, and launch **ChromeHistoryView.exe**.

Note: If an **Open File – Security Warning** window appears, click **Run.**

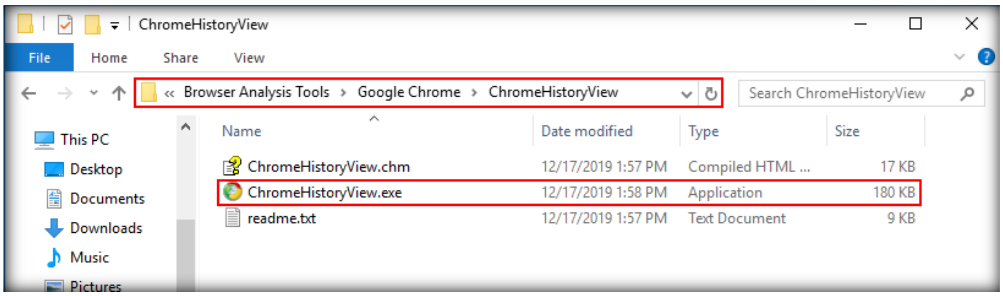


Figure 3.5: ChromeCookiesView tool folder

9. **ChromeHistoryView** displays information such as **URL**, **Title**, **Visited On**, **Visit Count**, **Visit ID**, **History File**, etc. for each cookie.

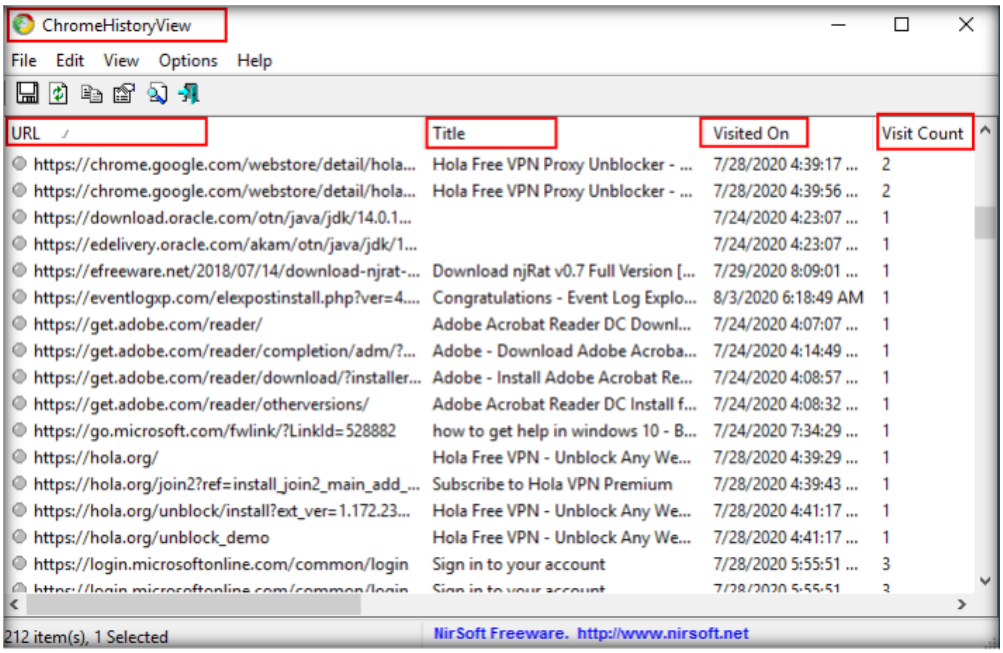


Figure 3.6: ChromeCookiesView tool

10. In this manner, you can view the Chrome browser artifacts to detect any suspicious entries in the browsing history.
11. Similarly, we will extract **Firefox** browser artifacts.
12. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Mozilla Firefox**.
13. In the **Mozilla Firefox** folder, you will find folders for three tools: **MZCacheView**, **MZHistoryView**, and **MZCookiesView**.
14. First, open the **MZCacheView** folder and launch **MZCacheView.exe**.

TASK 2

Retrieving Artifacts from Mozilla Firefox

Note: If an **Open File - Security Warning** window appears, click **Run.**

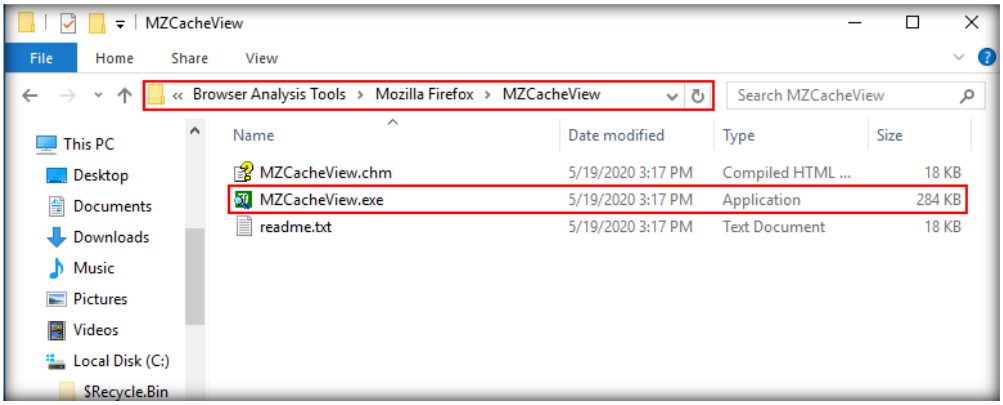


Figure 3.7: MZCacheView tool folder

15. **MZCacheView** displays information such as **Filename**, **Content Type**, **URL**, **File Size**, **Fetch Count**, **Last Modified**, **Last Fetched**, **Expiration Time**, **Server Name**, **Server Response**, **Server Time**, **Content Encoding**, **Cache Name**, **Missing File**, **Cache Control**, etc. for each cache file.

Note: The application can take some time to load all the cache entries.

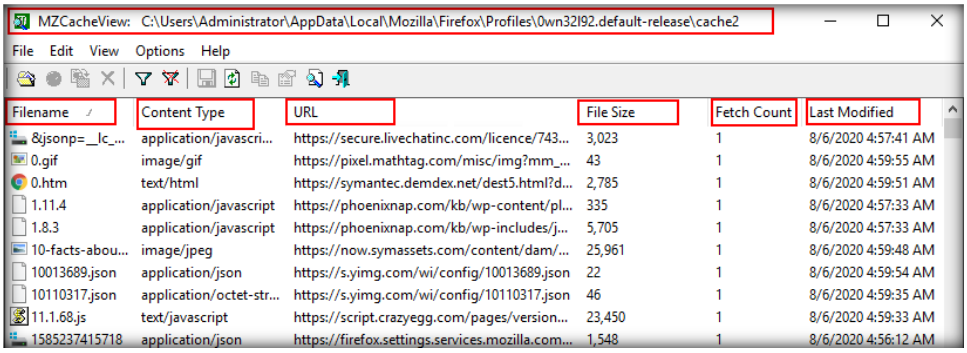


Figure 3.8: MZCacheView tool

16. Close **MZCacheView** window. Now, navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Mozilla Firefox\MZCookiesView** and launch **mzcv.exe**.

Note: If an **Open File – Security Warning** window appears, click **Run**.

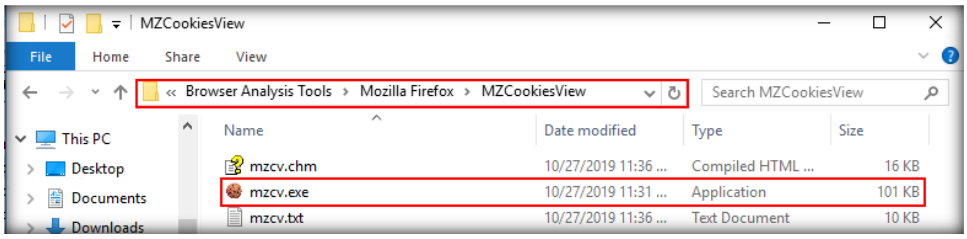


Figure 3.9: MZCookiesView tool folder

17. Upon launching, the **MZCookiesView** application window will appear, displaying a list of all cookies that have been generated by the user's visits to various websites. It will also display information such as **Domain/Host**, **Path**, **Name**, **Value**, **Expiration Date**, **Secure (Yes/No)**, **Last Accessed**, **Created Time**, etc. for each cookie.

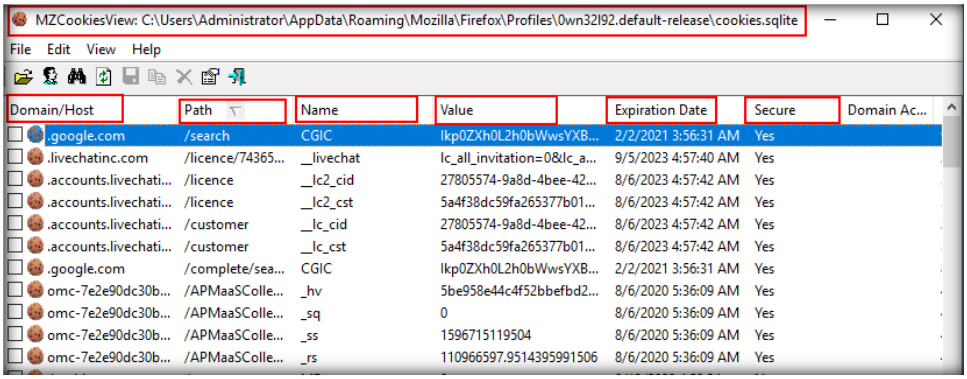


Figure 3.10: Mozilla cookies

18. Now, close the **MZCookiesView** window, navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools\Mozilla Firefox\MZHistoryView**, and launch **MozillaHistoryView.exe**.

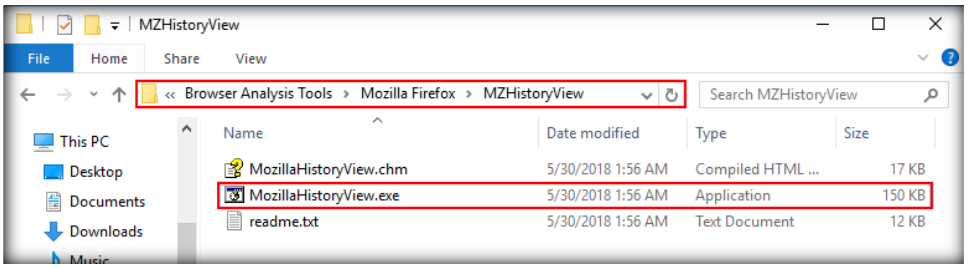


Figure 3.11: MozillaHistorView.exe

19. Upon launching **MZHistoryView.exe**, the application will first open a **Select History Filename** window. In this window, you will see a default directory path provided in the field against the **ellipsis** button. Leave the directory path set to default. All the other options are unchecked by default. Leave the options as set to default and then click **OK**, as shown in the screenshot:

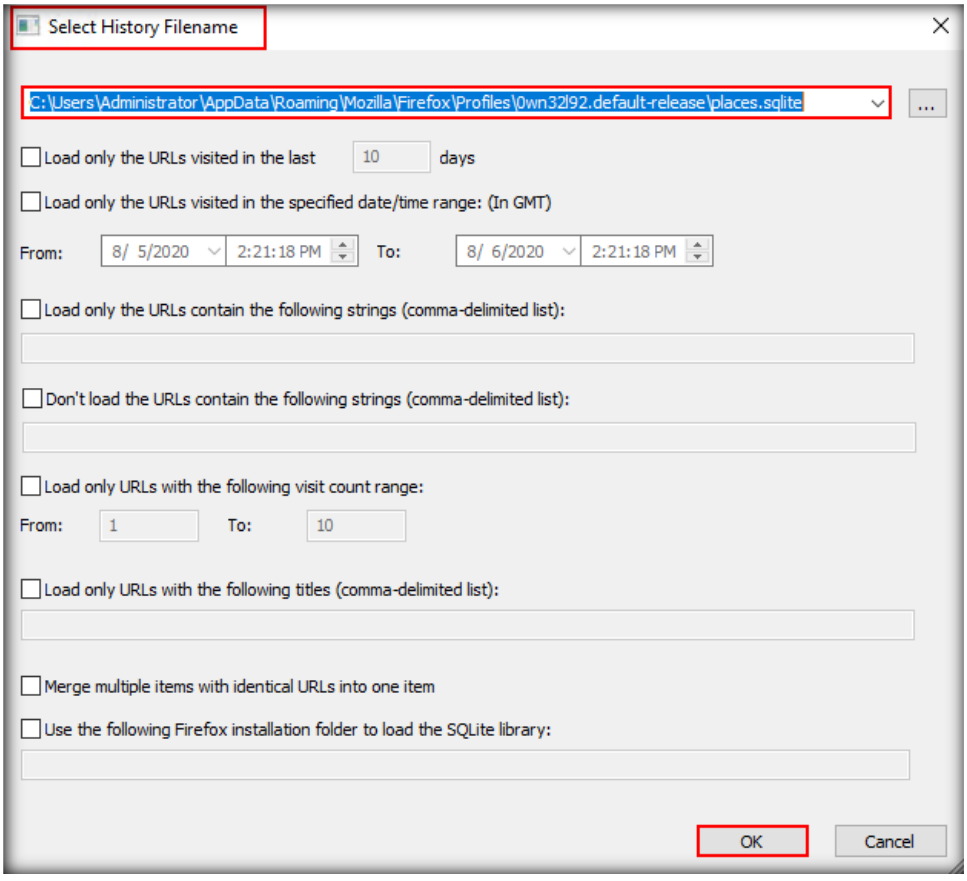


Figure 3.12: Select History Filename window

20. The main application window will now appear, displaying information such as **URL**, **First Visit Date**, **Last Visit Date**, **Visit Count**, **Referrer**, **Title**, **Visit Type**, etc. for each visited website, as shown in the screenshot:

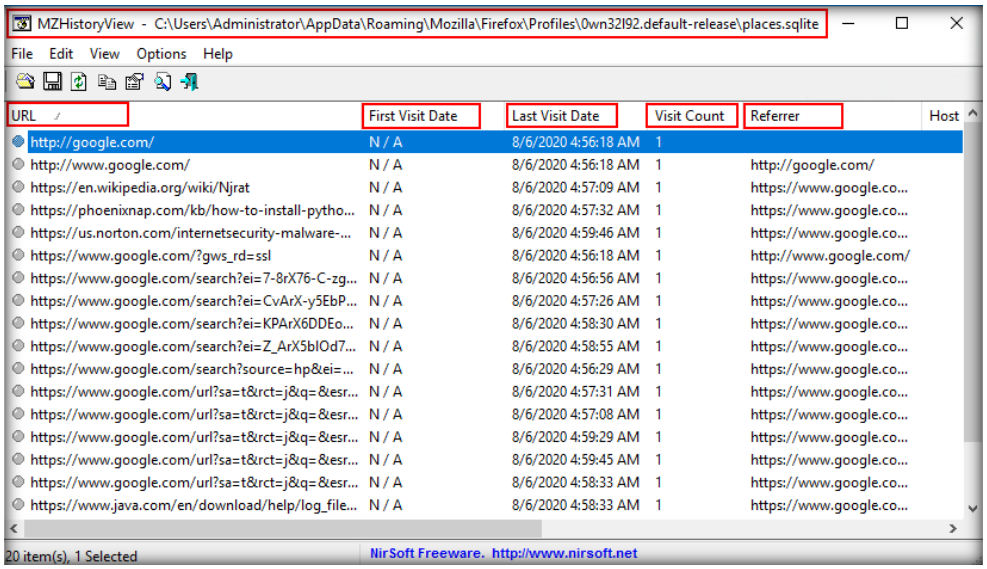


Figure 3.13: MZHistoryView tool

21. In this manner, you can view the Mozilla Firefox browser artifacts to detect any suspicious entries in the browsing history

22. To save the artifacts pertaining to any browser, regardless of whether they are cache files, browsing history files, or cookies, select the items containing evidentiary data, right-click, and then select the **Save Selected Items** option from the context menu.

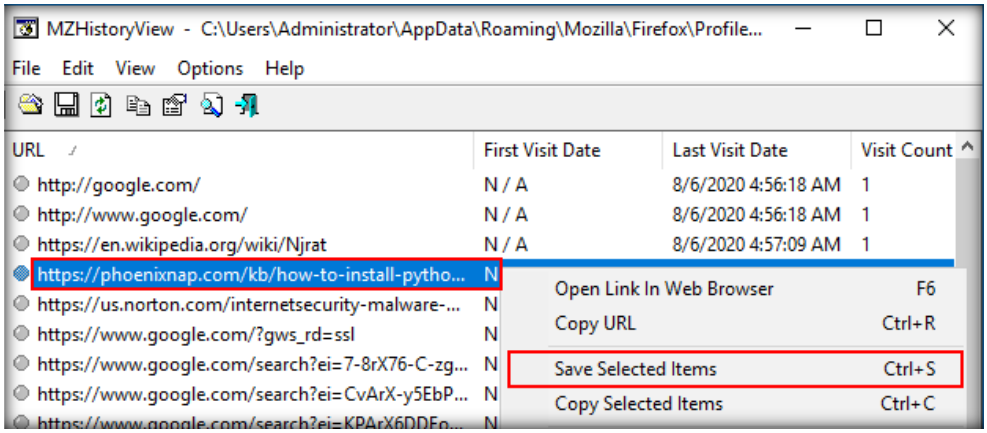


Figure 3.14: Saving selected artifacts

23. A **Select a filename to save** window will appear, displaying the default location where the file will be saved. The default location will be the respective tool folder in **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Browser Analysis Tools**. You may also choose another location to save the file. In the **Select a filename to save** window, depending on your requirement, name the file (here, it is **Mozilla Browsing History**) to be saved and then click **Save**. The file will be saved to the default/selected location as a **.txt** file.

Note: For demonstrative purposes, we have only presented the saving of a single item. You can also select multiple items.

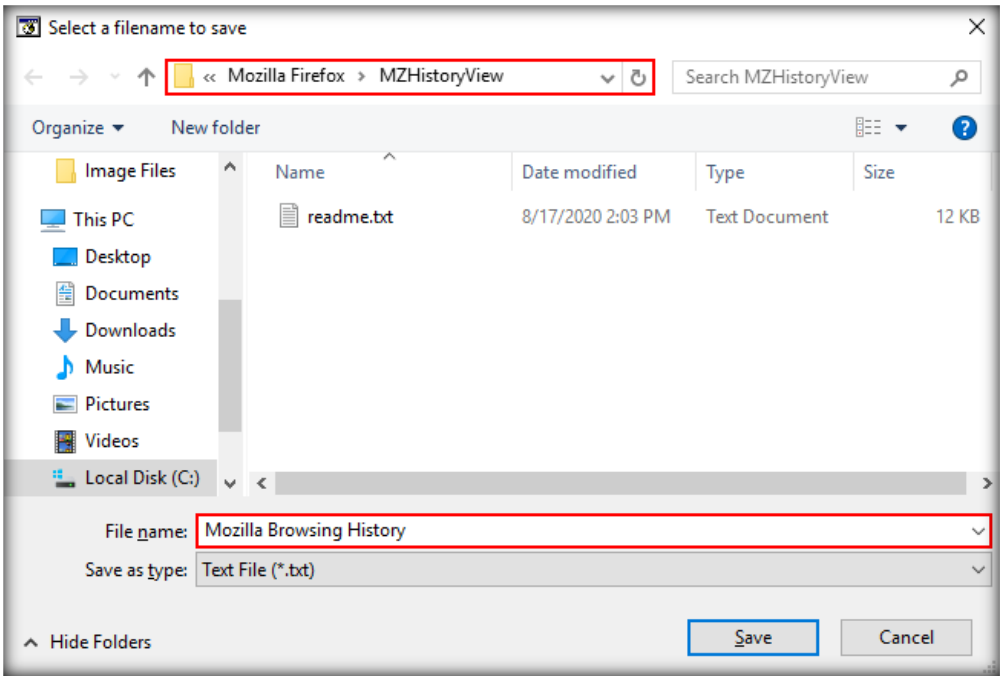


Figure 3.15: Select a filename to save

24. In this manner, you can view the browser artifacts to detect any suspicious entries in the browsing history during the course of investigation.

Lab Analysis

Document the complete results of this lab.

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

Lab

4

Discovering and Extracting Forensic Data from Computers

During forensic investigation, the investigator must identify any suspicious activity on the target system and extract evidence that could be useful for investigation.

ICON KEY

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

Lab Scenario

Searching for an evidence file in a system is like searching for a needle in a haystack, unless you know what to look for and where. A workaround is to find a helpful tool. A system stores logs for events, along with the identity of programs that perform any small task. A forensic investigator should be able to extract evidence from data. This lab will help you use the OSForensics tool, which will assist in finding evidence from huge amounts of data.

Lab Objectives

The objective of this lab is to help you learn how to investigate a suspect's computer to locate evidence for a crime.

Lab Environment

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics

This lab requires:

- A computer running **Windows Server 2016** virtual machine
- Administrative privileges to execute commands
- A web browser with internet access
- **OSForensics** installer located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\OSForensics**

Note: You can download the latest version of **OSForensics** from the link https://www.osforensics.com/download.html

If you are using the latest version of software for this lab, then the steps and screenshots demonstrated in the lab might differ.

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 25 minutes

Overview of the Lab

This lab familiarizes you with the process of searching for and extracting artifacts of evidentiary value during a forensic investigation on a computer or from any other source of forensic evidence.

Lab Tasks

TASK 1

Creating a New Case

1. Login to the **Windows Server 2016** virtual machine.
2. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\OSForensics**, double-click **osf.exe** to launch the setup and follow the wizard-driven installation instructions to complete the installation of the tool.

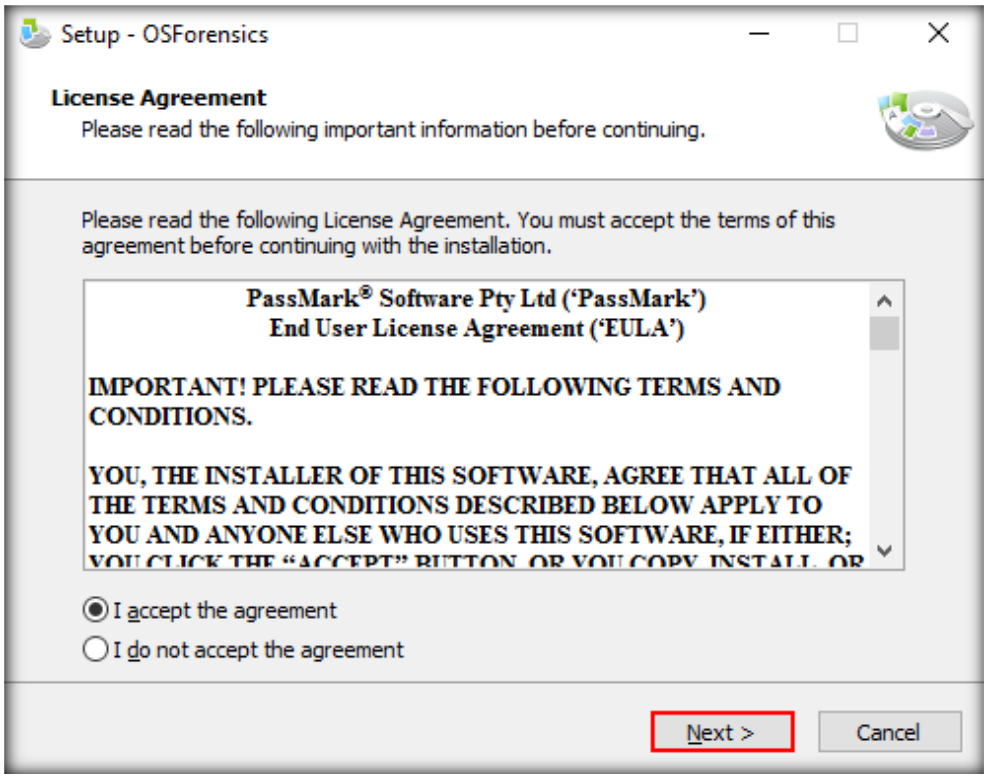


Figure 4.1: OSForensics setup wizard

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

3. In the final step of installation, check the **Launch OSForensics** option and click **Finish**.

4. **OSForensics** GUI appears, along with **PassMark OSForensics** pop-up. In the pop-up, click **Continue Using Trial Version**.

By default, a case is created in the OSForensics folder situated in the user's My Documents folder. Upon the creation of a case, a subfolder will be created at the target location that will contain the case. There is no need to select an empty folder.

FIGURE 4.2: Main window of OSForensics

5. Our first task is to create a case using this tool. Click the **Create Case** icon in the tool's main window to create a new case.

Once a case is created or opened, the contents of the case can be managed from this window. Case items can be opened or deleted, and additional properties can be viewed. The property viewer also allows for editing user-defined properties of the item.

FIGURE 4.3: Create a new case in OSForensics

6. A **New Case** window appears; fill out the required fields in the window. Ensure that you select the radio button for the **Live Acquisition of Current Machine** option in the **Acquisition Type** category. You may choose to save the new case folder either in **Default** or **Custom** locations. Click **OK**.

Note: In this lab, we have set the default location for the case folder.

The start window shows a brief description of each feature as you hover your mouse over it; clicking once will take you to that feature. Additionally, you can click the buttons on the left. A button with a pulsating green light next to it indicates that the feature is currently performing a task. A blue light indicates that the task has been completed, but you have not yet navigated back to that window to view the result.

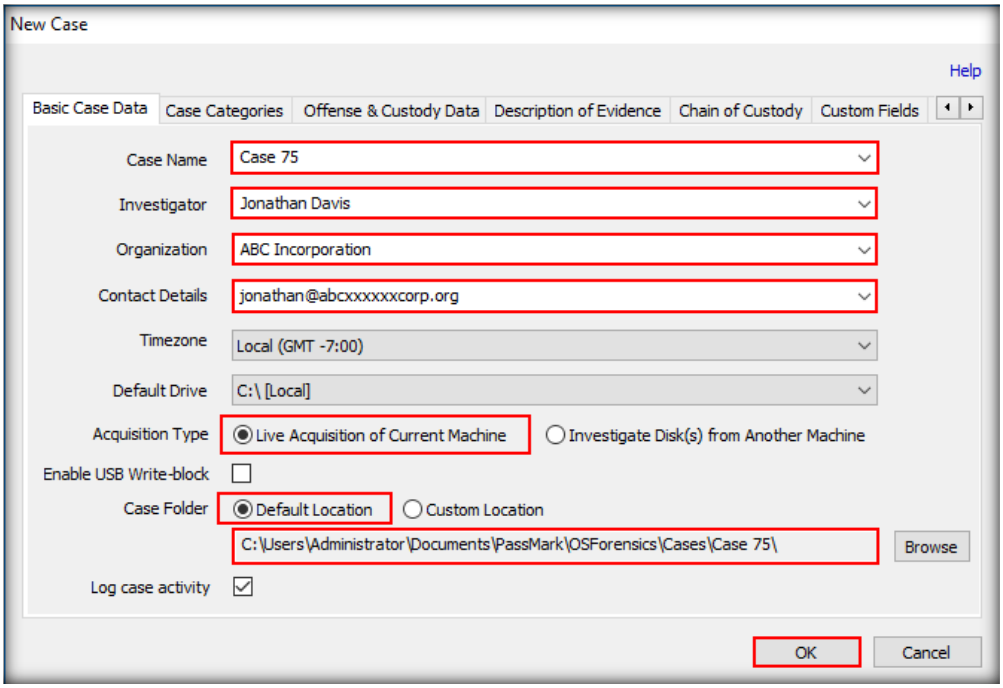


FIGURE 4.4: OSForensics New Case window

7. A new case is created as shown in the below screenshot:

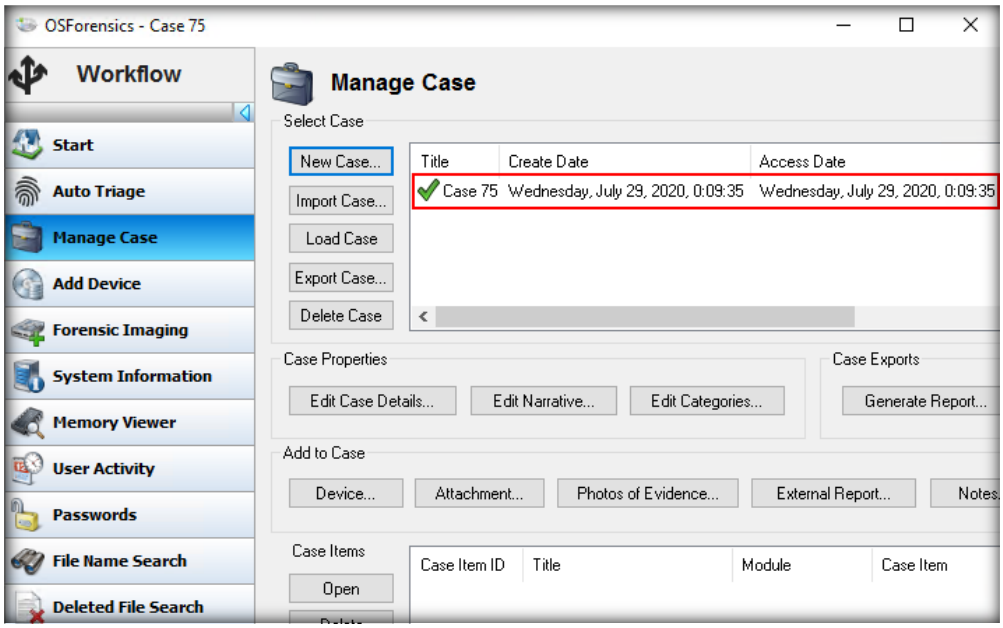


FIGURE 4.5: New case created in OSForensics

TASK 2

File Name Search

8. The **OSForensics** tool can help investigators in searching and locating files on a system. To start searching for files, click **File Name Search** in the left pane of the window.

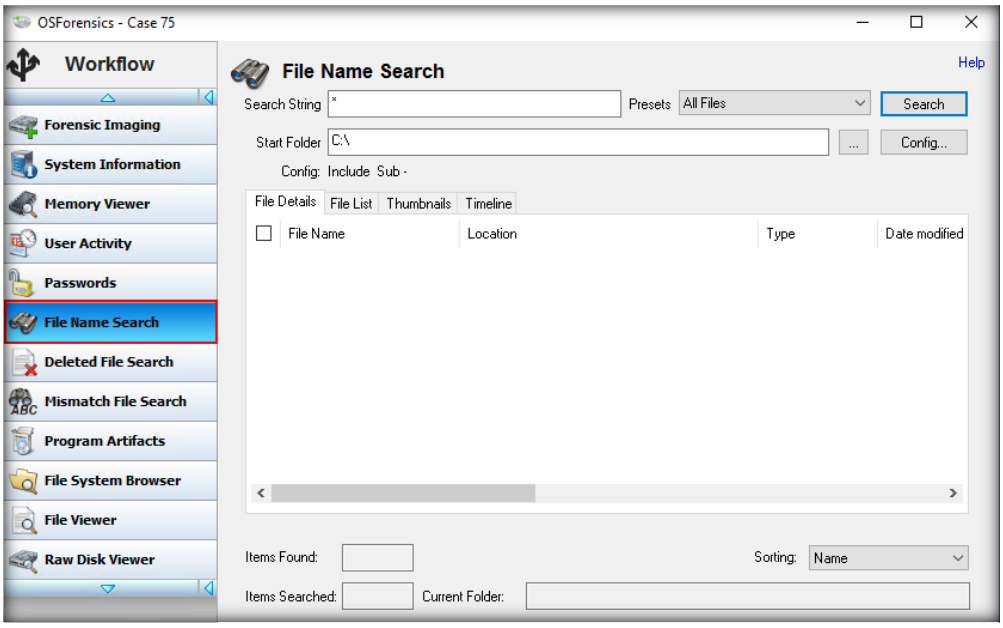


FIGURE 4.6: File Name Search in OSForensics

A basic search simply involves entering a search string and location. Any files or folders that contain the search string within their name will be displayed in the search results. For instance, searching for "File" will return "file.txt," "test.file," or "MyFile.doc." The basic search is case insensitive.

9. To locate files, type a file name or its extension in the **Search String** field under the **File Name Search** section. Alternatively, you may select the file type from the **Presets** drop-down list.

10. In this lab, we will be searching for image files. So, choose the **Images** option from the **Presets** drop-down list.

11. In the **Start Folder** field, specify the path to search for image files by clicking the **ellipsis** button and choosing the location (here, we are specifying the location **C:\CHFI-Tools\Evidence Files\Image Files** to search for images in it). Then, click the **Search** button.

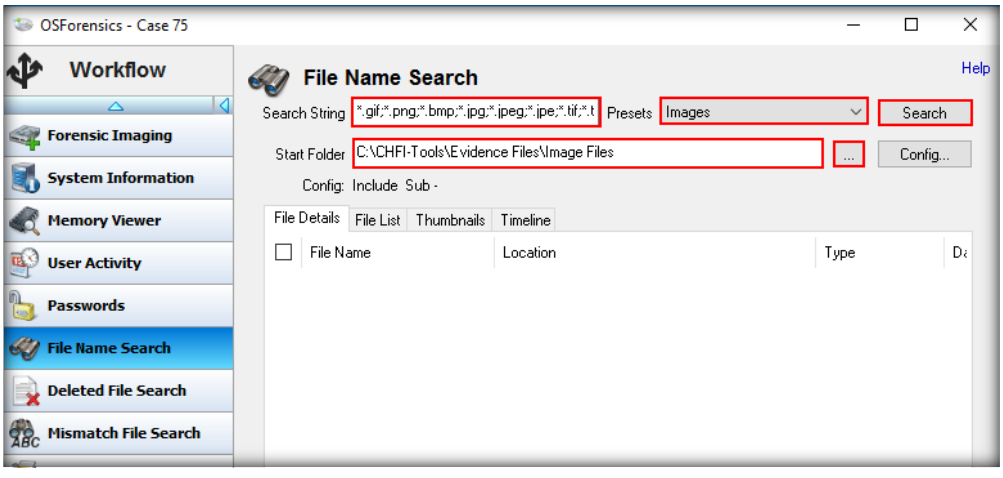


FIGURE 4.7: Searching for image files in OSForensics File Name Search

If a wildcard is entered anywhere in the search field, wildcard matching is enabled on all search terms. When wildcard matching is enabled, you will need to explicitly add "*" to the start and end of the search term if you are attempting to match a word that may appear in the middle of a filename.

12. The **File Name Search** section now displays the list of files contained in the selected location, as shown in the screenshot below. By default, the tool displays these files under the **File Details** tab. You may also view this list of files under the **File List** and **Thumbnails** tabs by clicking on them.

🗁 On clicking the **Config** button, you will be taken to the **File Name Search Configuration** window, where more advanced options can be selected.

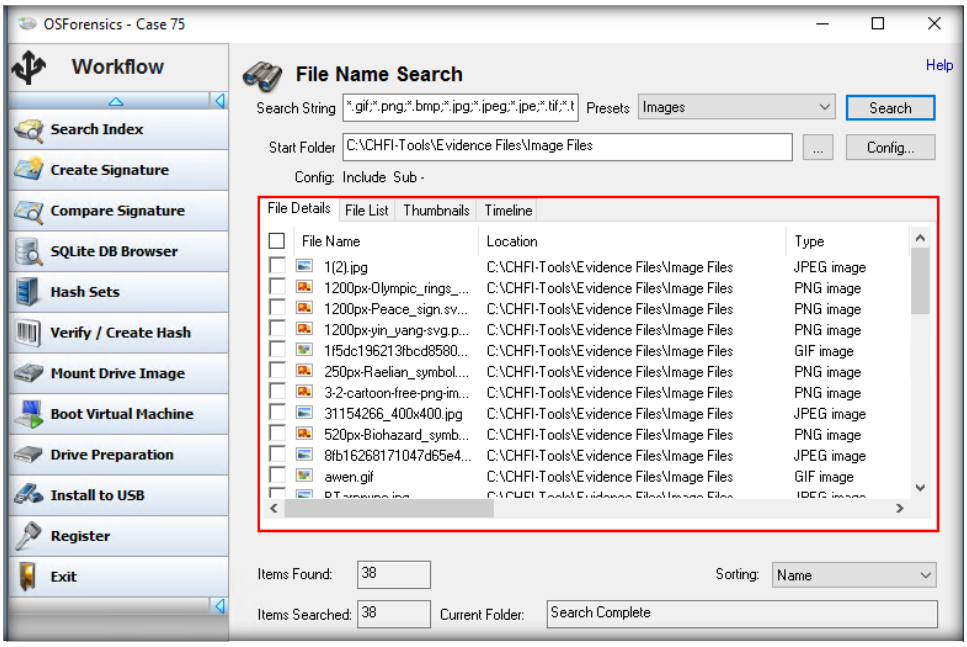


FIGURE 4.8: Display of results in OSForensics File Name Search

Note: The **Timeline** tab allows you to view/sort files according to their modified, accessed, or created times.

13. Alternatively, if you wish to search only for a single image on the entire system, select **Images** from the **Presets** drop-down menu, enter the name of the desired image file in the **Search String** field, and then click **Search**, as shown in the screenshot below. Here, we are searching for the JPEG image file named **Kitty**.

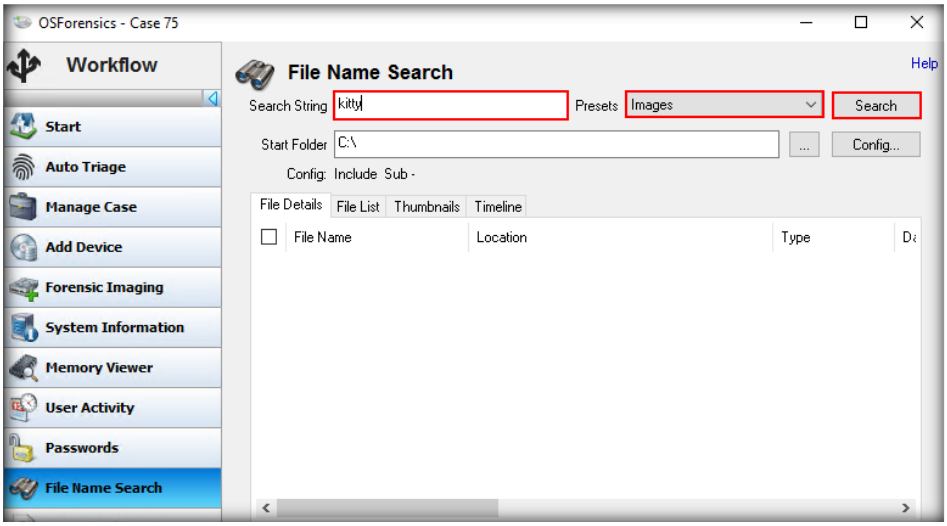


FIGURE 4.9: Searching for a single file

14. The tool will search for the specified image file and display the search result in the **File Name Search** section under the **File Details** tab. The search result will be displayed along with the path/location where the searched file is stored.

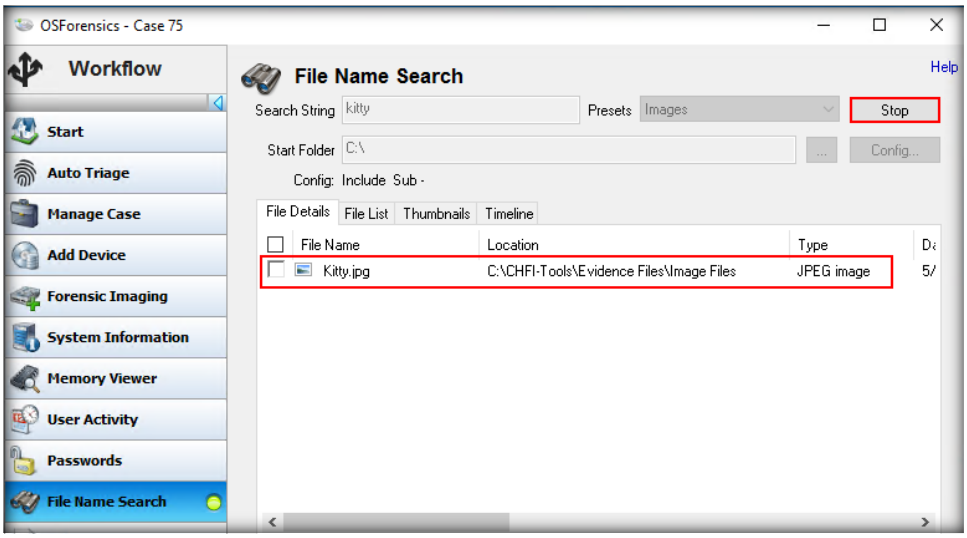


FIGURE 4.10: Search Result Generated

15. To view the image, double-click it. The tool will display the image through a pop-up, as shown in the following screenshot:

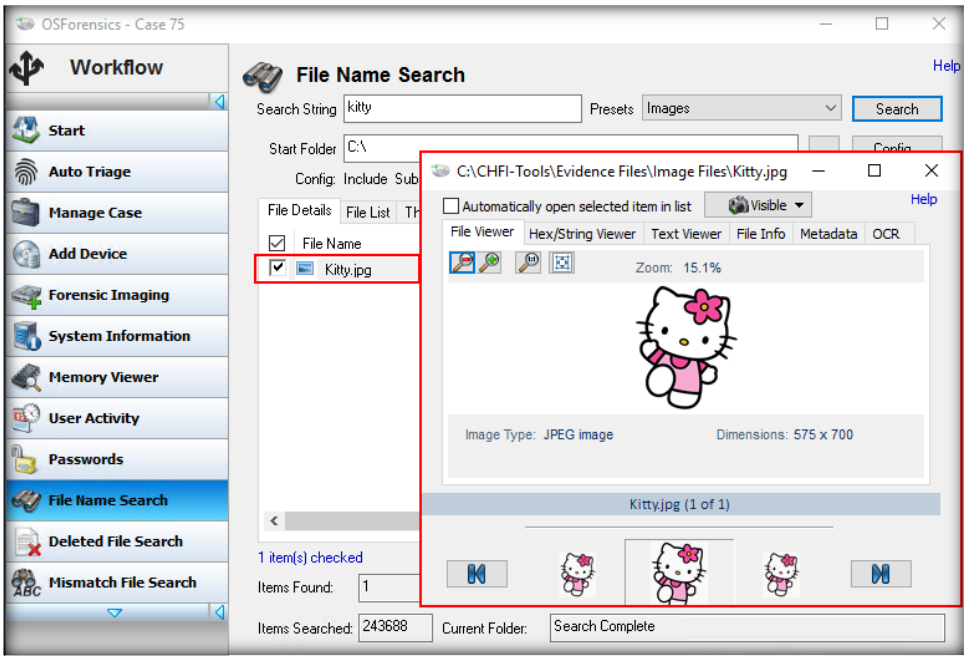


FIGURE 4.11: Open the searched image file

Note: Similarly, the tool also allows you to search for all the image files that are present on a disk partition. For demonstrative purposes and to save time, we have confined our search operation mostly to the **Image Files** sub-folder within the **Evidence Files** folder.

TASK 3

Creating Index

16. We will now proceed to the task of creating an index of files. Here, we demonstrate the creation of an index for image files. Creating an index of files helps investigators expedite their file search.
17. To start creating an index of files, click **Create Index** in the left pane of the tool window.
18. The **Create Index** section appears in the right pane. In this section, select the **Use Pre-defined File Types** option for creating the index and check the required options listed under it for selecting the file types that you wish to index (here, we have selected the **Images** option). Then, click **Next**.

The **Indexer Advanced Configuration Window** allows users to configure various indexing parameters. This window can be accessed by clicking the **Config** button in the **Create Index** window.

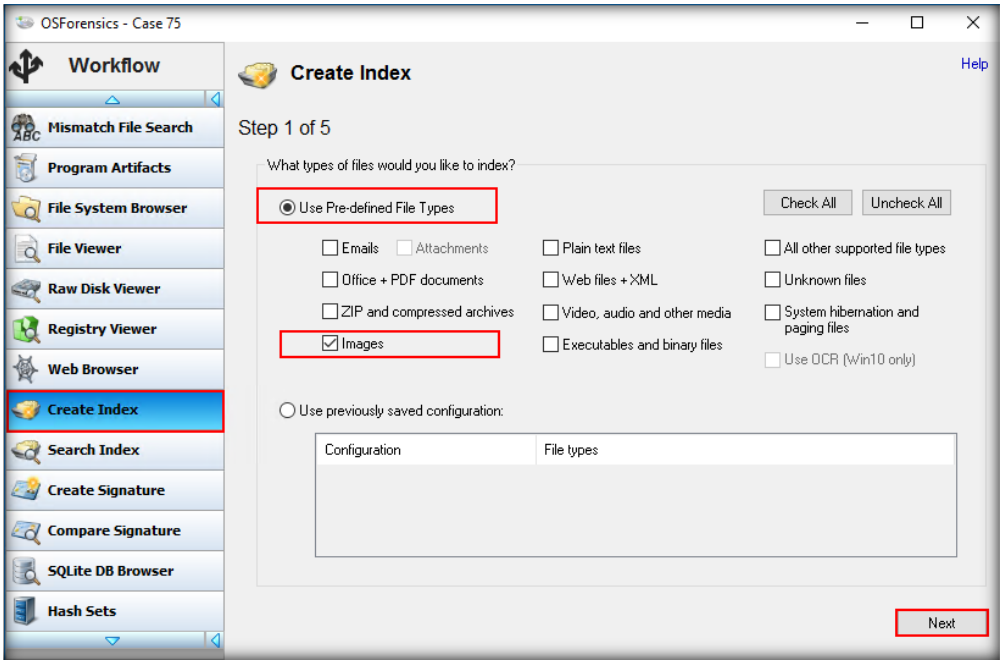


FIGURE 4.12: Step 1 for creating an index in OSForensics

19. Now, click the **Add** button to select the target drive or folder in which the files which you wish to index are stored.

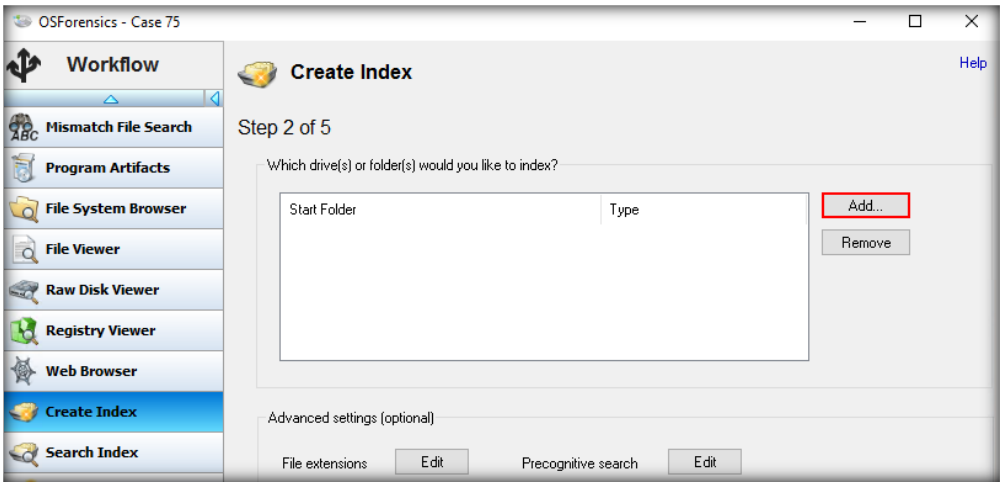


FIGURE 4.13: Step 2 for creating an index in OSForensics

20. When you click the **Add** button, the **Add Start Location** pop-up appears. From this pop-up, select the **Specific Folder** radio button, and click the **ellipsis** button to provide the path to the target folder. For this lab, select the **C:\CHFI-Tools\Evidence Files\Image Files** folder and click the **OK** button.

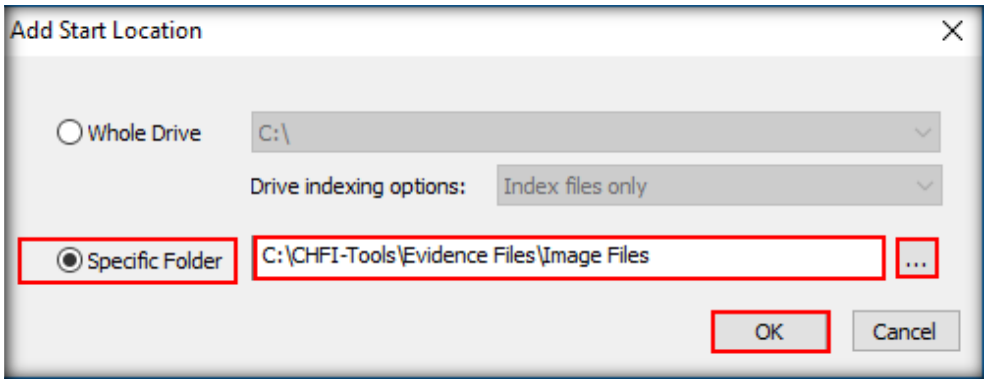


FIGURE 4.14: Continuation of Step 2 for creating an index in OSForensics

📖 The list of file types for which the contents will be scanned is configured here. Typical file extensions are added to the list by default. To add a new file extension, click the **Add** button.

21. The **Start Folder** is now configured. Click **Next**.

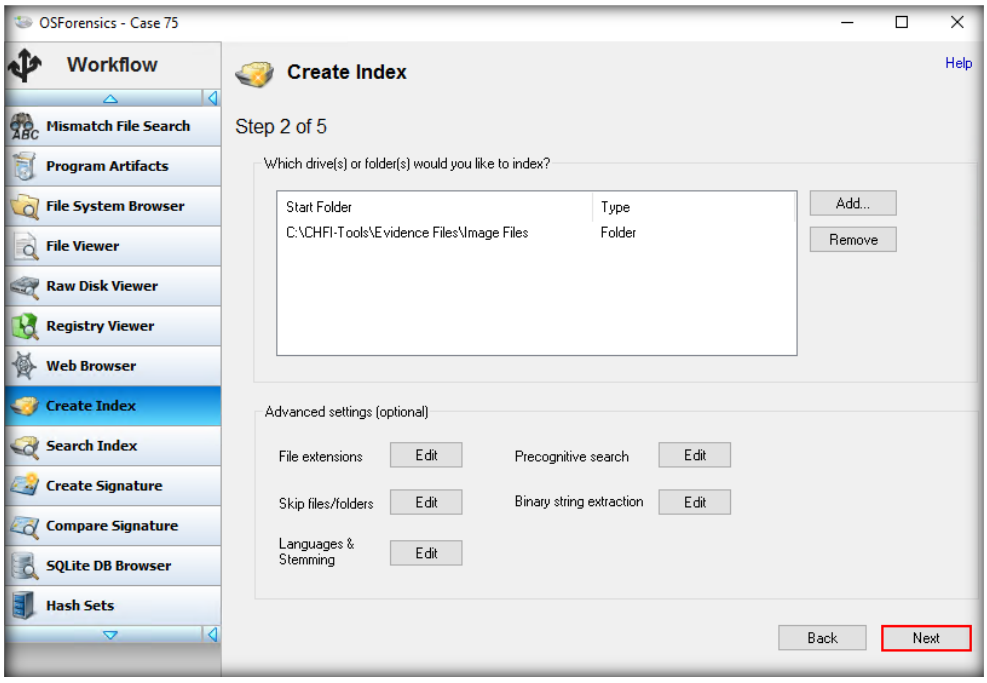


FIGURE 4.15: Continuation of Step 2 for creating an index in OSForensics

📖 Pages and folders containing particular words can be excluded from the scan by adding the words to the list. Note that the folder to which the created index files are written is also automatically added so that the indexing process does not index the files it creates. This folder is a sub-folder of the currently active case folder.

22. In the **Step 3** of the wizard, select the **Don't Know (Pre-Scan Required)** option, ensure that the **Use RAM drive for temporary files to speed up indexing** option is checked, and click **Next**.

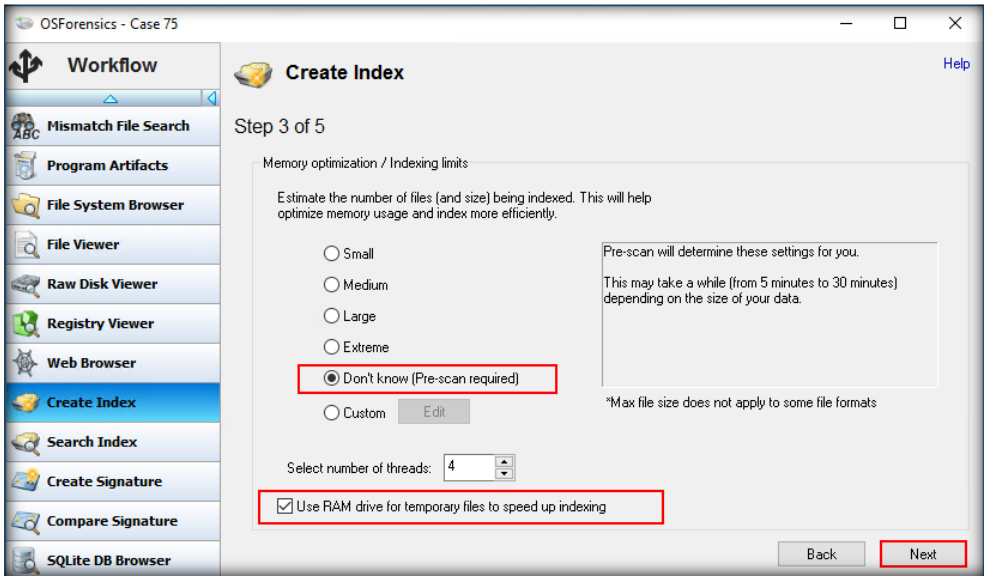


FIGURE 4.16: Step 3 for creating an index in OSForensics

23. In **Step 4** of the wizard, specify an **Index Title** (here, we have specified the **Index Title** as **Case 75**), enter **Index Notes** (optional), and then click **Start Indexing**.

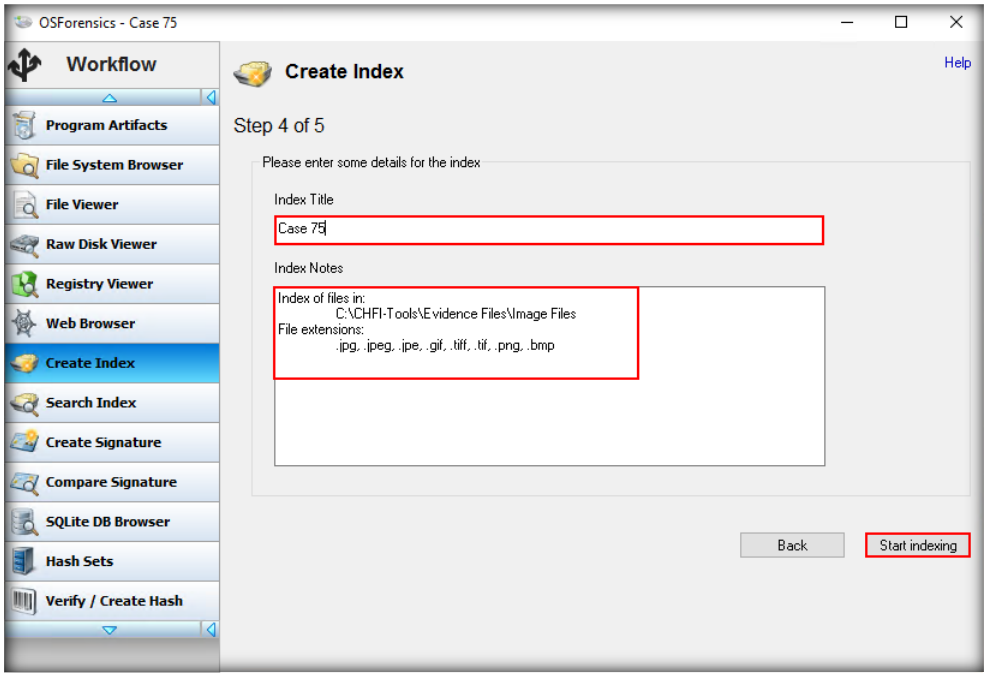


FIGURE 4.17: Step 4 for creating an index in OSForensics

Binary String Extraction Level:

When attempting to extract words out of binary data, the indexing process must make a decision as to what is a word and what is random data. Changing this option will determine how lenient or strict the indexer is when making this decision. Leaving this on default, the most stringent option, is recommended, as this will aggressively remove irrelevant data and keep the index at a manageable size. The **Code Words** setting is useful if you are attempting to find information such as passwords missed by the default option.

24. The application will now start performing a pre-scan on the specified folder.

Note: If a **Maximum page limit in Free Version** dialog box appears while scanning, click **OK** to close it.

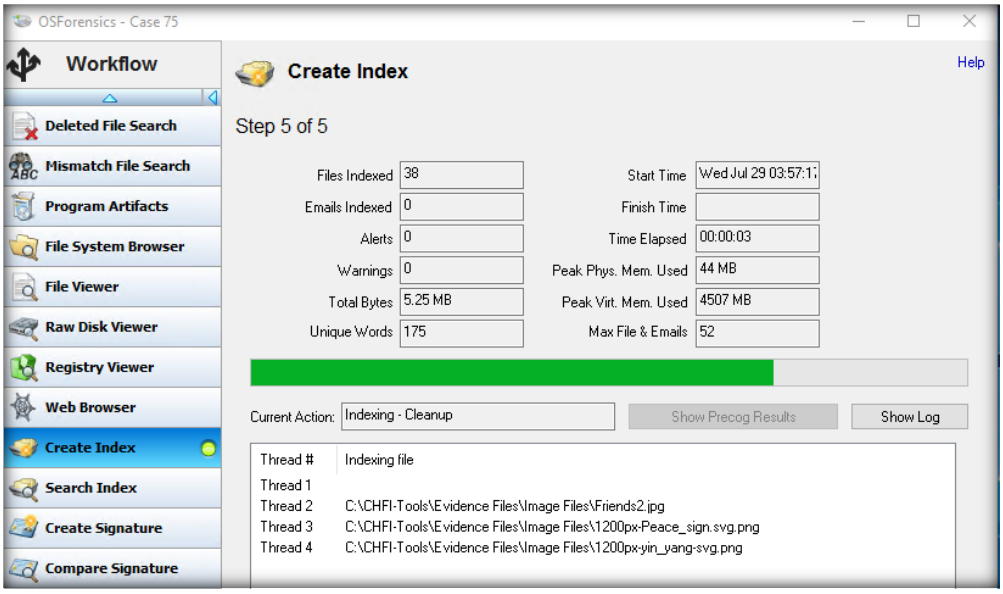


FIGURE 4.18: Step 5 for creating an index in OSForensics, which involves a pre-scan

25. Once the indexing is completed, the application changes the status of the indexing process to **Finished**, as can be seen in the **Current Action** field.

If you are indexing an active system drive (the drive that Windows is running from), this is quite common as many programs and Windows itself will be using the files on the drive, making them inaccessible. Usually, these files are system files without much interesting text in them; therefore, this should not be a problem.

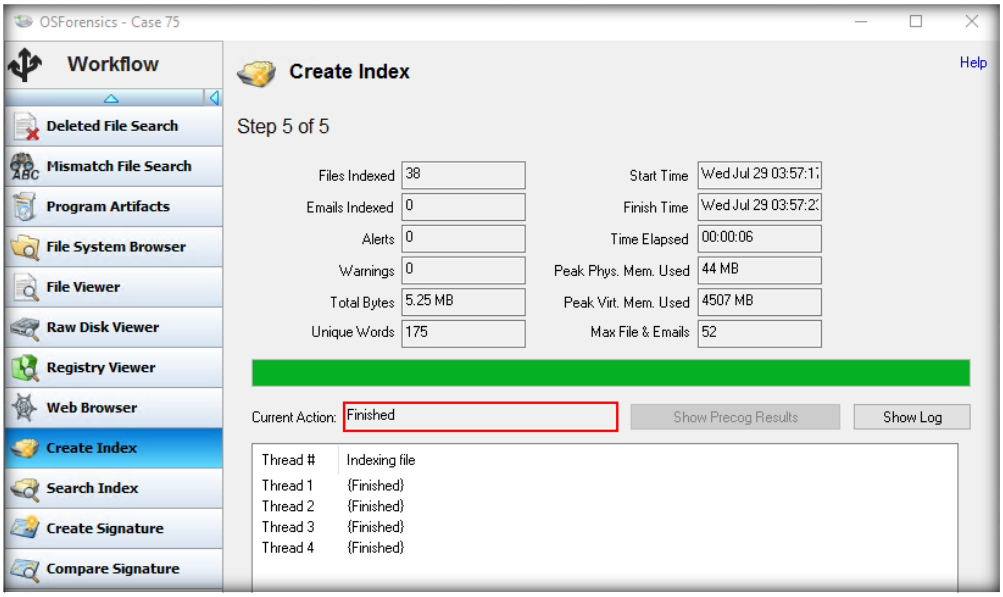


FIGURE 4.19: Completion of the pre-scan in Step 5 for creating an index in OSForensics

26. We will now search the indexed files. To search for the indexed files, select **Search Index** from the left pane of the tool window. The **Search Index** section now appears in the tool window. The index field at the top displays the name of the case we created (i.e., **Case 75**). Click **Search**. The tool will load the image files that you have indexed under the **Images** tab.

Note: The trial version of OSForensics returns a limited number of results. So, if the search results exceed this number, an **OSForensics - Notice** pop-up might appear, where you need to click **OK**.

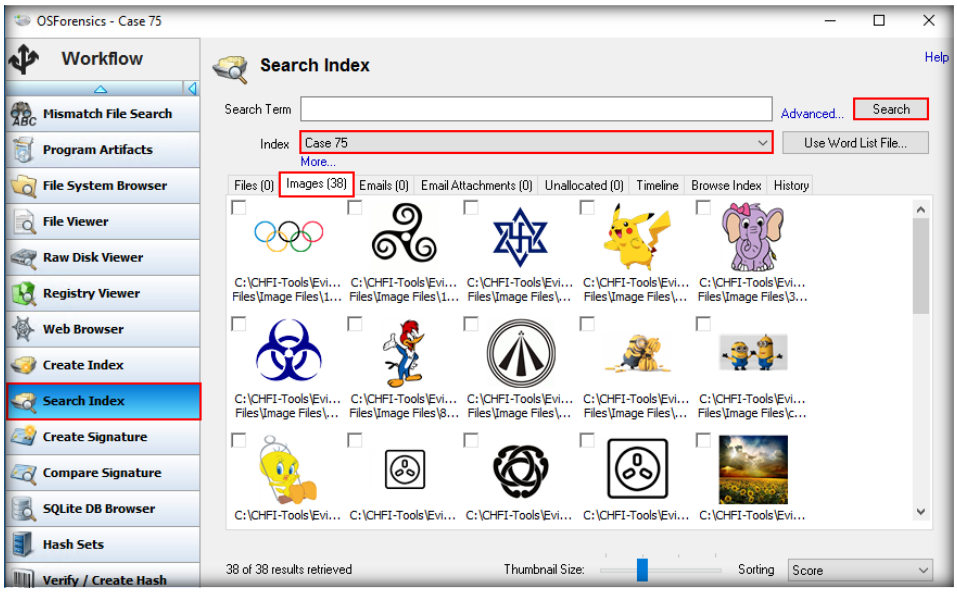


FIGURE 4.20: Search index in OSForensics (image files)

During the pre-scan step, OSForensics tries to detect the number of files that will be included in the index and sets this as the maximum which the indexing process will scan. Because the pre-scan is a basic and fast scan, it may sometimes get this wrong. If this is the case, you should try indexing again by setting the maximum pages manually in the advanced options.

27. Similarly, you can also index other files such as emails and search for them using the **Search Index** option. In that case, the indexed emails will be displayed under the **Emails** tab.

TASK 4

Searching User's Recent Activities

28. Now, we will gather all the user activities that occurred on the system.

29. Click **User Activity** in the left pane to scan for evidence such as browsed websites, USB drives, recent downloads and wireless networks.

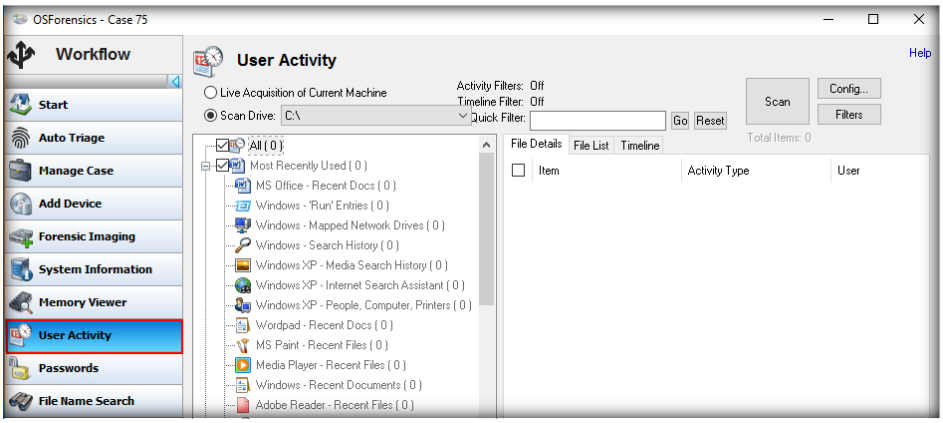


FIGURE 4.21: Click User Activity

30. In this lab, we will scan for activities that took place in the **C:** drive. Select the **Scan Drive** radio button, ensure that the **C:** drive is selected from the **Scan Drive** drop-down menu, and then click **Scan** to scan for activities such as browsed websites, USB drives, recent downloads, and wireless networks in the drive.

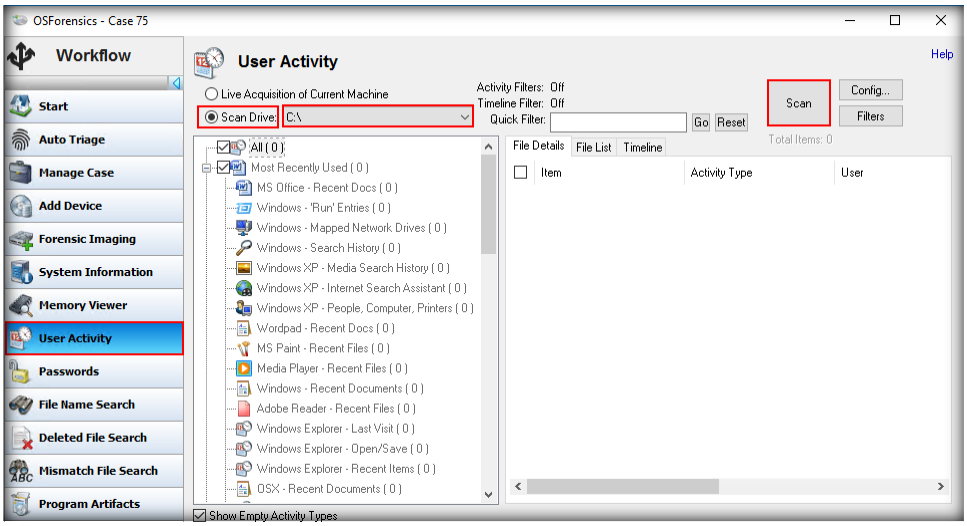


FIGURE 4.22: Scanning user activity

Note: If a **Warning** pop-up appears, click **Yes**. If an **OSForensics Error** appears during the scan, click **OK**.

31. If a **Select device to add** window appears, click **OK**

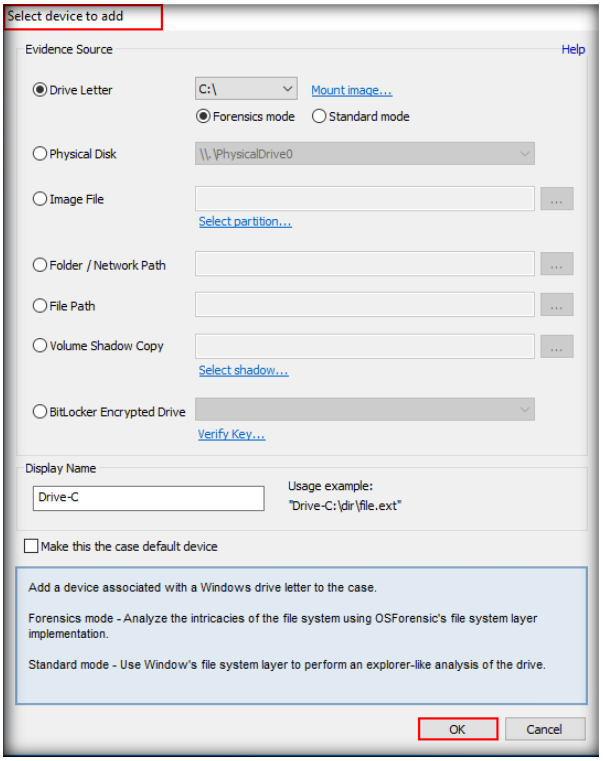


FIGURE 4.23: Select a device to add

32. The application takes some time to scan, and upon completion of the scan, a **User Activity - Summary** window pops up, displaying a summary of information pertaining to user activity such as **Browser History**, **Downloads**, **Most Recently Used**, and **Installed Programs**. Click **OK** to close the **User Activity – Summary**.

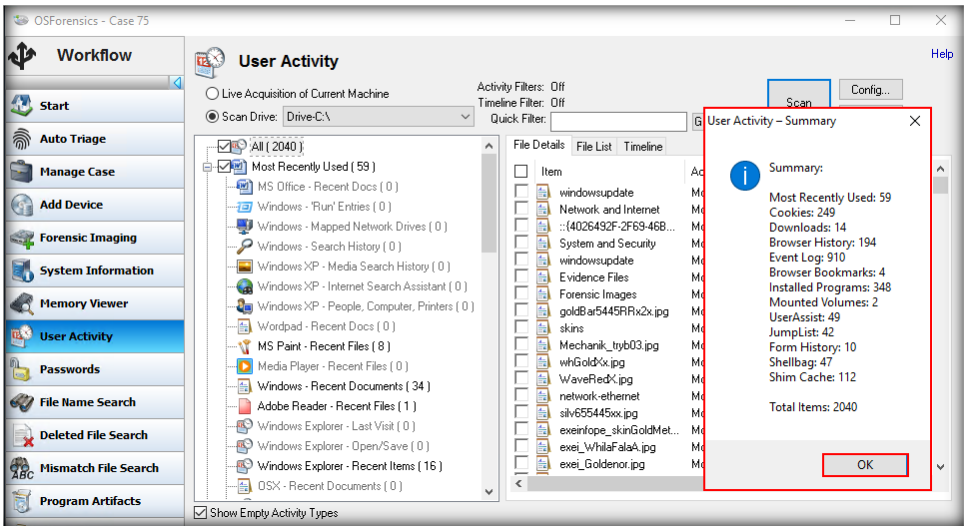


FIGURE 4.24: View User recent activity in OSForensics

33. If you wish to save any of the above user-activity information on your system, such as **Browser History**, then scroll down the items list in the left pane below the **Scan Drive** option and expand the **Browser History** node. Select the browser for which you wish to retrieve the history (Here, we are selecting **Chrome)**. The browser history pertaining to this browser will be displayed in the right pane of the tool window.

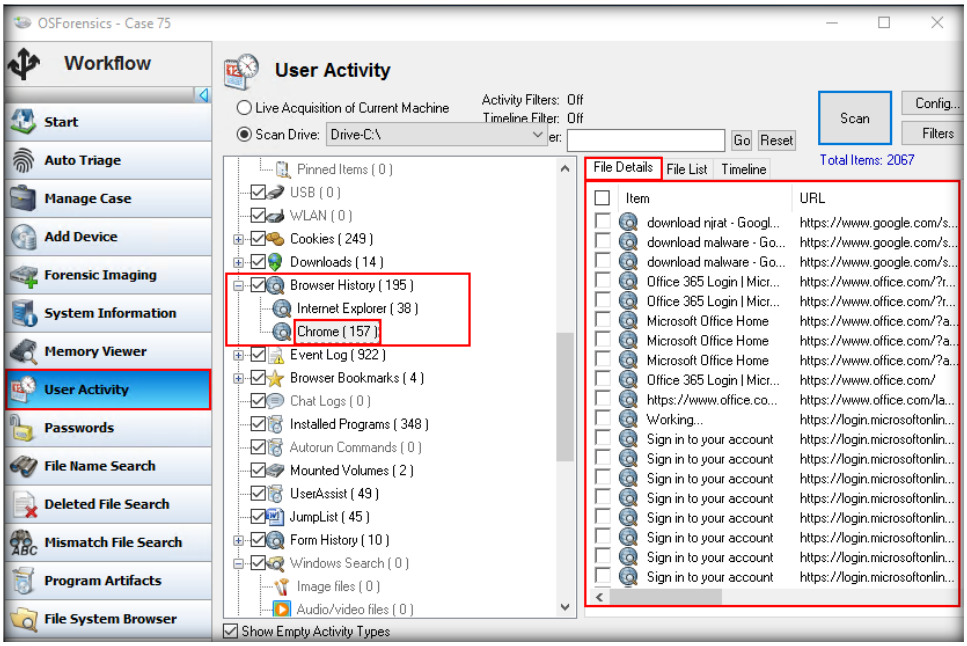


FIGURE 4.25: Find browser history

OSForensics scans known locations for web-browser profiles and their related history and cache files to detect cookies, visited URL history, downloads, and saved logins and passwords.

34. To retrieve the entire browser history, select any one of the files in the right pane and right click it. From the context menu, click **Export list of all results to**, and from the resulting drop-down menu, click **txt**.

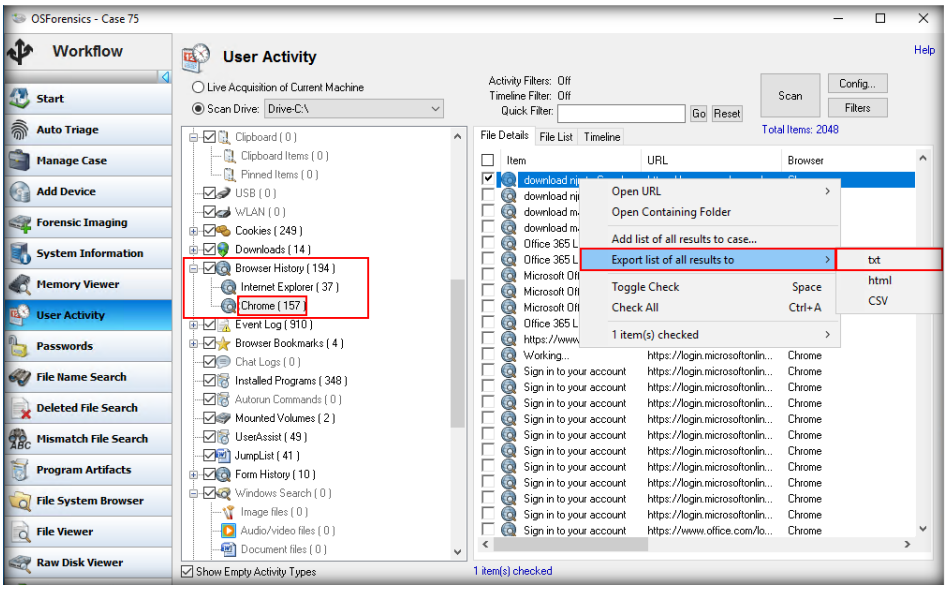


FIGURE 4.26: Exporting browser history

35. An **Export List to...** window will appear. Navigate to **C:\Users\Administrator\Documents\PassMark\OSForensics\Cases\Case 75**, name the file (here, we are naming it as **Browsing History**), and click **Save** to export the browsing history.

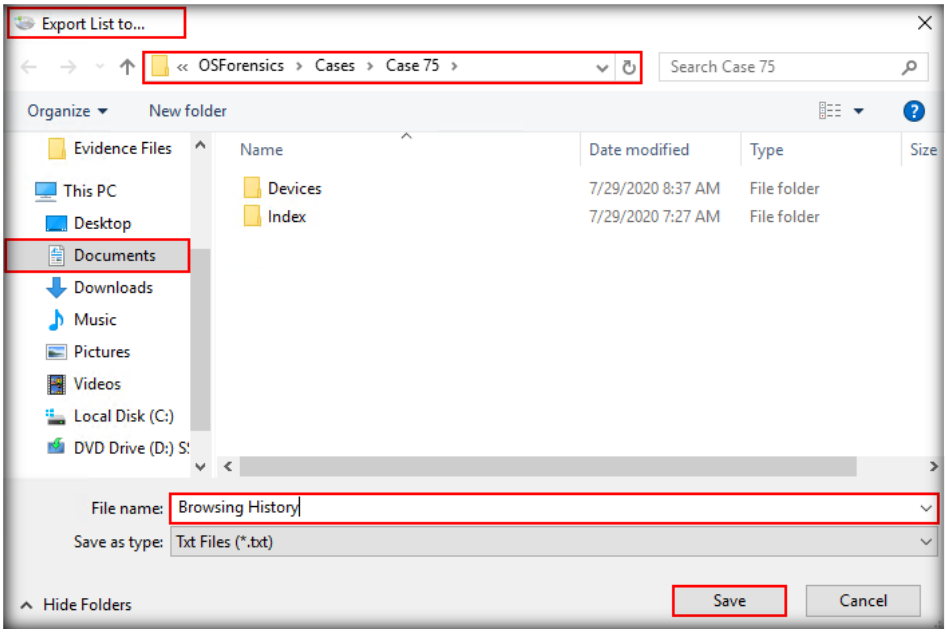


FIGURE 4.27: Save browsing history

Note: Since this is a trial version of the **OSForensics** tool, the number of items you can export to your system is limited. So, if you see the **User Activity export limit in Trial Version** pop-up appear as shown in the

screenshot below, click **OK** to close it. If you wish to export all the files, you need to upgrade to the professional version of the tool.

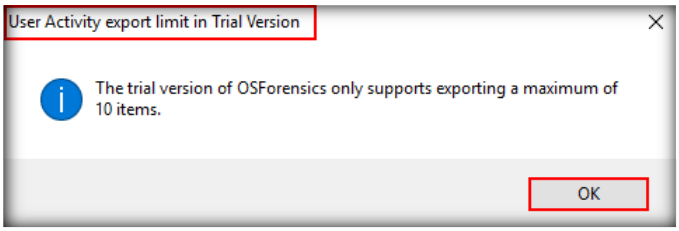


FIGURE 4.28: Export limit in the trial version of OSForensics

36. The history will be exported and saved to a text file. Navigate to the case folder and double-click the **Browsing History** file to view the information stored in it.

37. We will now examine how to recover deleted files using this tool.

TASK 5

Searching Deleted Files

38. To recover deleted files from the file system, click **Deleted File Search** in the left pane, select a disk on which you wish to perform the deleted file search from the **Disk** drop-down menu (here, we are choosing **Partition 1, C:**), and click the **Search** button.

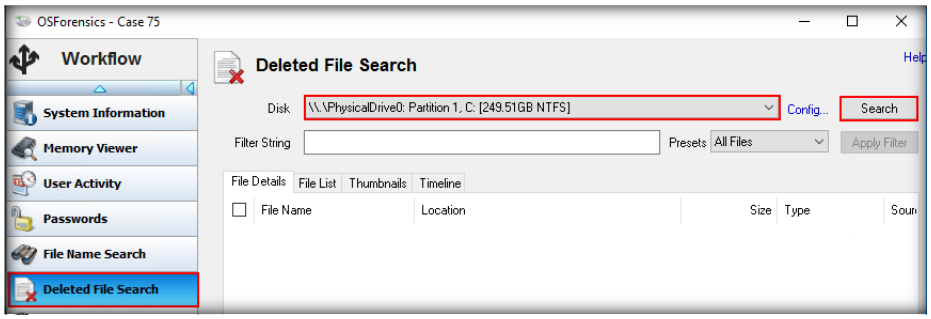


FIGURE 4.29: Deleted File Search in OSForensics

39. The application will run a scan on the selected drive, retrieve the deleted files, and display them under the **Thumbnails** tab shown in the following screenshot:

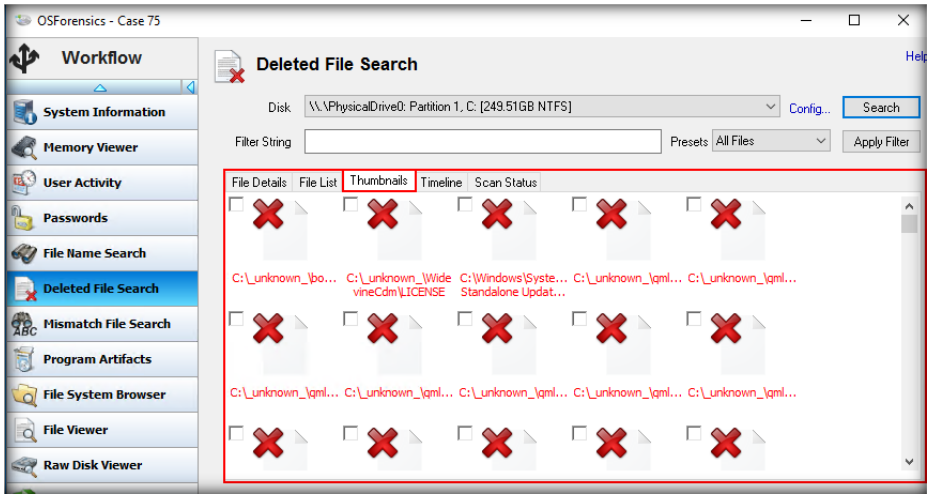


FIGURE 4.30: Display of deleted files in OSForensics

Note: Alternatively, you can view the retrieved files under the **File Details** tab or **File List** tab. Under the **Scan Status** tab, you will see all the details pertaining to the scan.

40. To save and view contents of the deleted files that have been retrieved by the tool, follow the same procedure as described in the above steps and name the file being exported as **Recovered Deleted Files**.

TASK 6

Searching Mismatch Files

41. We will now detect files with an extension mismatch. For this, we will be using the **Mismatch File Search** feature. This feature is helpful in cases where an attacker has saved malicious files to a system by modifying their extensions.

42. To locate files having contents that do not match the file extensions, click **Mismatch File Search** in the left pane of the tool window. The tool will display the **Mismatch File Search** section in the right pane, as shown in the screenshot:

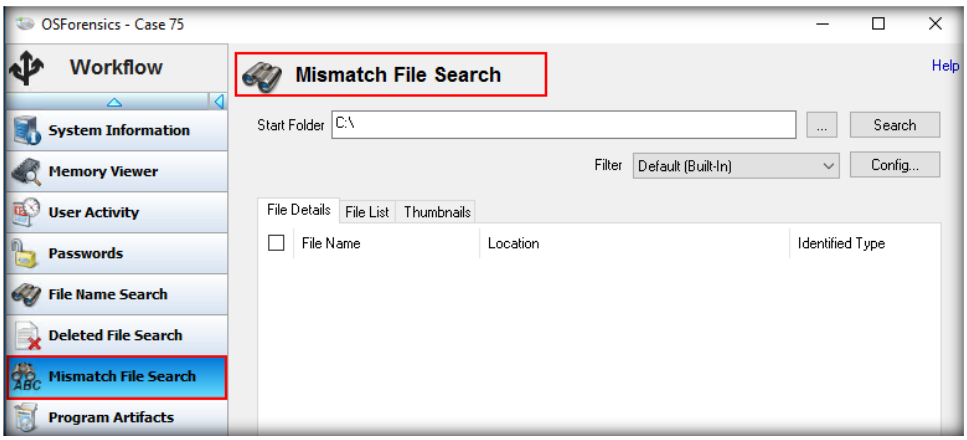


FIGURE 4.31: Mismatch File Search section

43. Click the **Ellipsis** button pertaining to the **Start Folder** field to provide the location in which files with extension mismatch are to be searched for. For this task, we will be searching for files with extension mismatch in the **Image Files** folder. So, navigate to **C:\CHFI-Tools\Evidence Files\Image Files** and click the **Search** button.

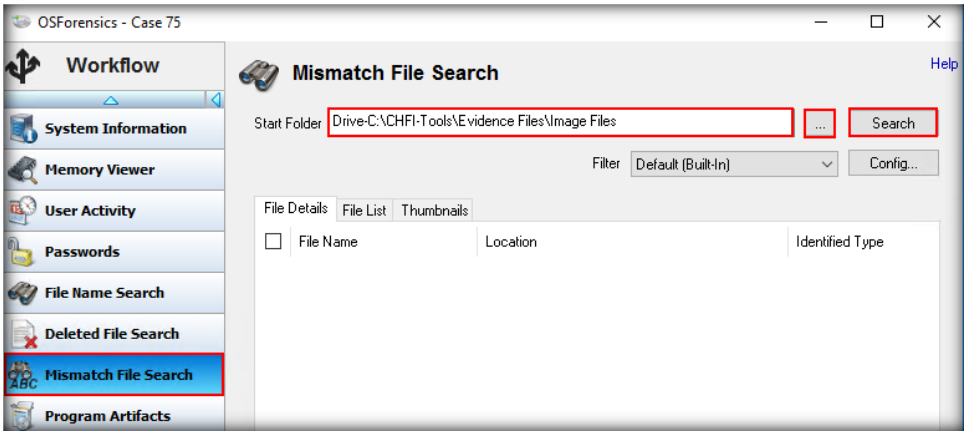


FIGURE 4.32: Providing the path and clicking Search

44. The tool will scan the selected directory and display all the files with extension mismatch under the **File Details** tab by default. Click on the **File List** tab for the ease of viewing the files and their details as shown in the following screenshot:

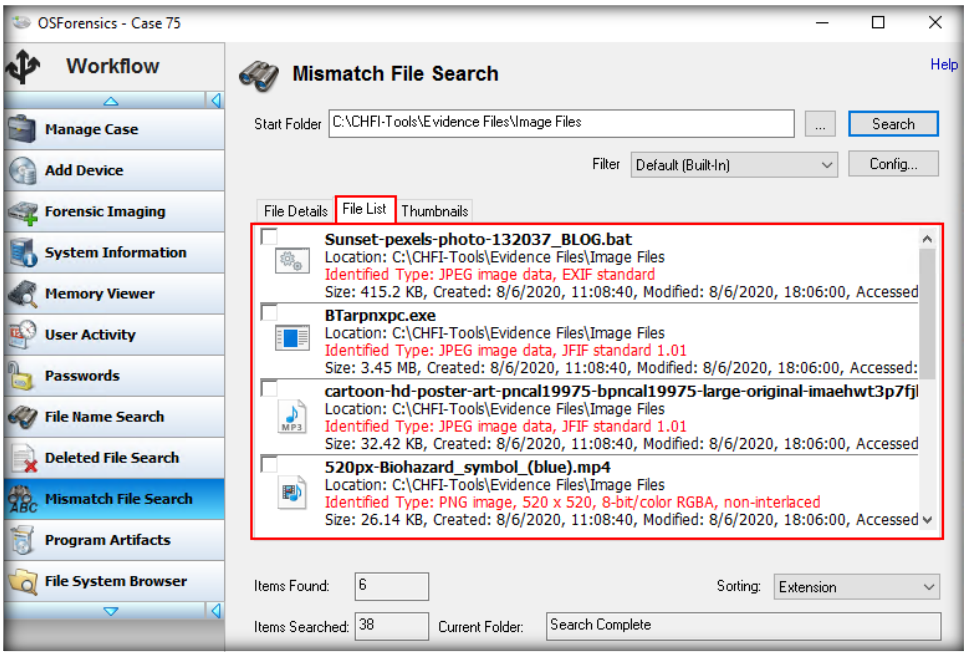


FIGURE 4.33: Mismatch File Search in OSForensics

Connected USB devices:

USB devices that have been connected to the computer include USB memory sticks, portable hard drives, and other external USB devices such CD-ROM drives. The manufacturer name, product ID, serial number and the last connection date should be displayed for each device.

TASK 7

Viewing Active Processes Running on the Computer Memory

45. We will now proceed to the task of viewing running processes on the system.
46. To view the processes running on the system, click **Memory Viewer** in the left pane of the tool window. The tool will now display the **Memory Viewer** section.

Note: If an **OSForensics – Warning** pop-up appears, click **OK**.

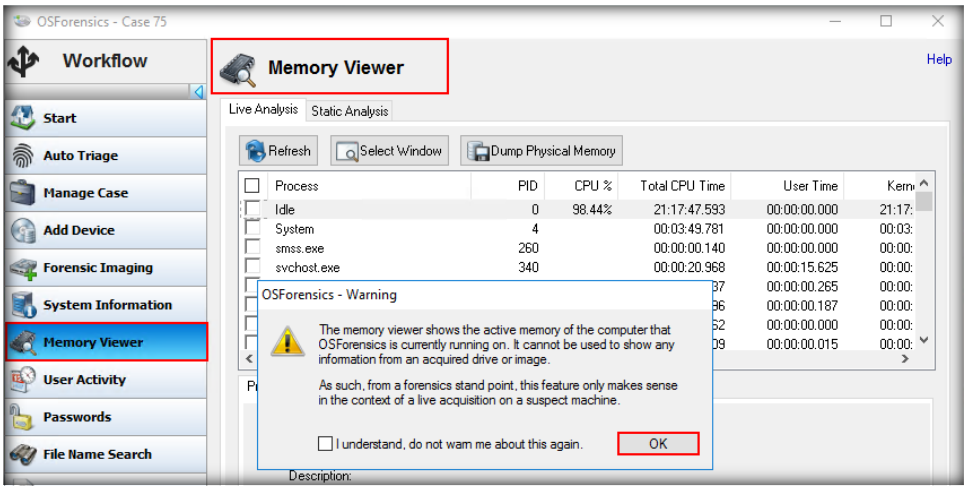


FIGURE 4.34: Memory viewer warning pop-up in OSForensics

OSForensics checks several known registry locations that store MRU data; these include locations for Microsoft Office, Microsoft WordPad, Microsoft Paint, Microsoft Media Player, Windows Search, recent documents, connected network drives, and the Windows Run command.

47. Select a **Process** from the list of running processes, and then select the **Memory Space** tab to view the complete memory details of that process.

Performing a process memory dump saves the contents of a process' virtual memory space (both in physical memory or paged out to the hard disk) into a file. This is useful especially if there is a particular process that the user has identified to contain information of potential interest.

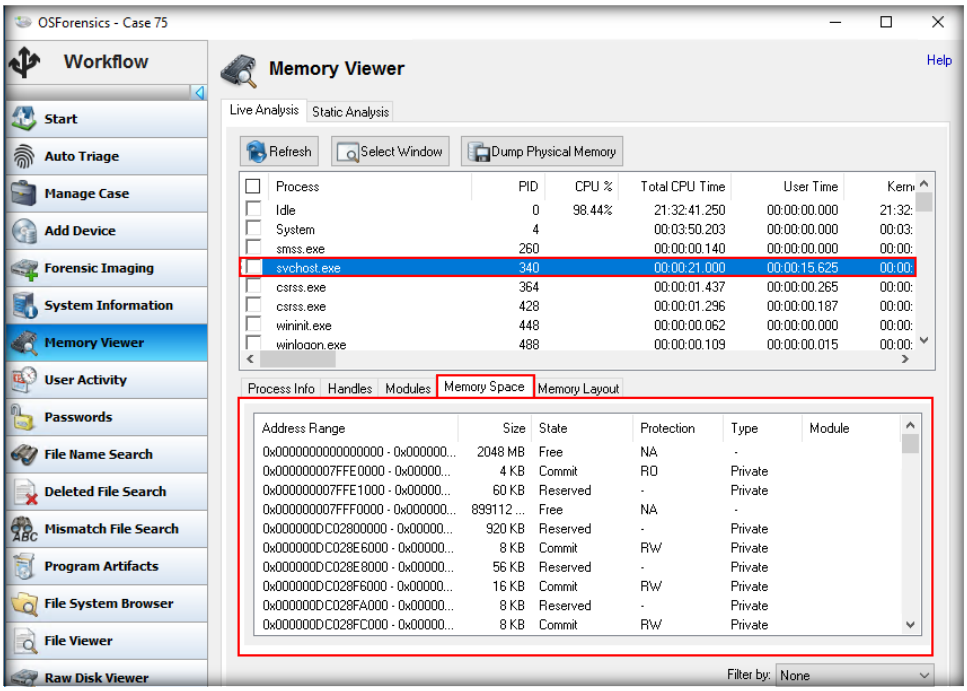


FIGURE 4.35: Memory Space tab in the Memory Viewer of OSForensics

48. To view the process information pertaining to the selected file, click on the **Process Info** tab. The information pertaining to that process will be displayed as shown in the screenshot:

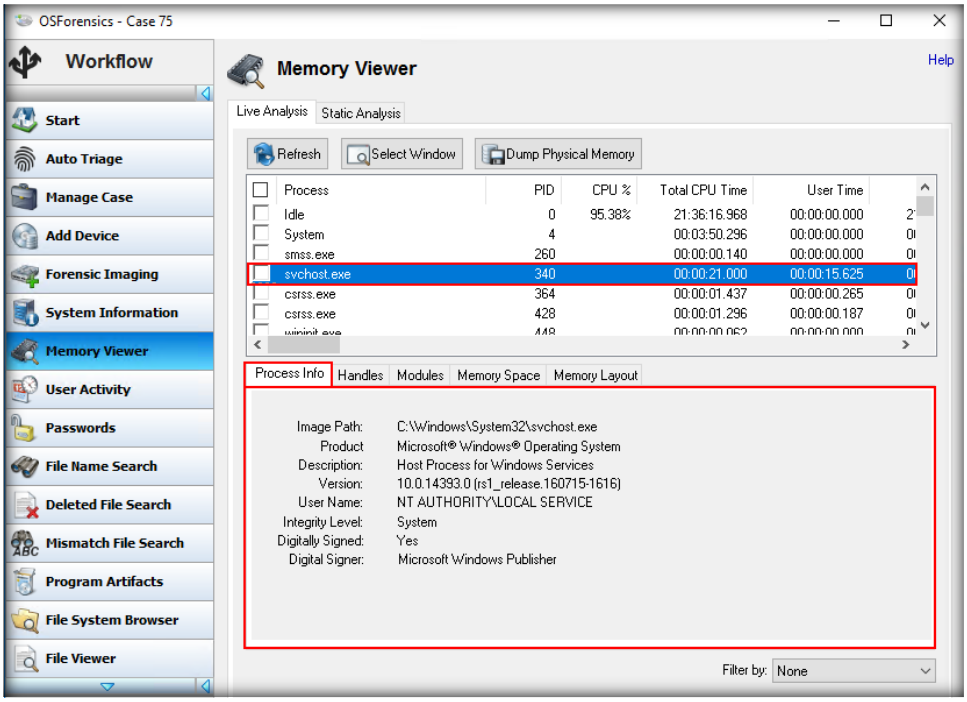


FIGURE 4.36: Viewing process info

TASK 8

Viewing System Information

49. We will now proceed to the task of viewing system information.
50. To retrieve detailed information about the core components of the system, click **System Information**. The tool will display the **System Information** section.

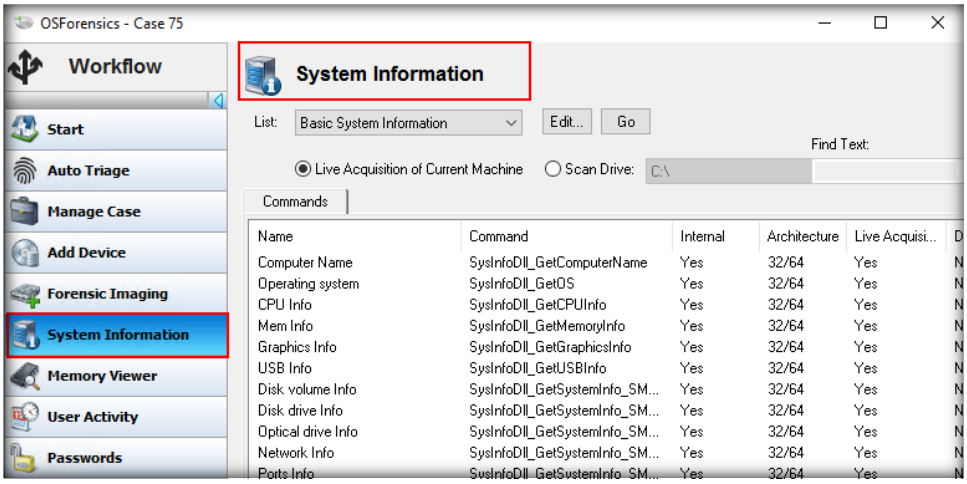


FIGURE 4.37: System Information section

51. From the **List** drop-down menu, select any of the options based on the type of information you wish to retrieve and then click **Go**. The options displayed in the list drop-down menu are **Basic DOS Commands**, **Basic system information**, **System Information From Registry**, **BitLocker Detection**, **Recover BitLocker Keys**, **Python Scripts**, and **All commands**.
52. Here, we select **All Commands** from the **List** drop-down menu. Ensure that the radio button against **Live Acquisition of Current Machine** option is selected and then click **Go**.

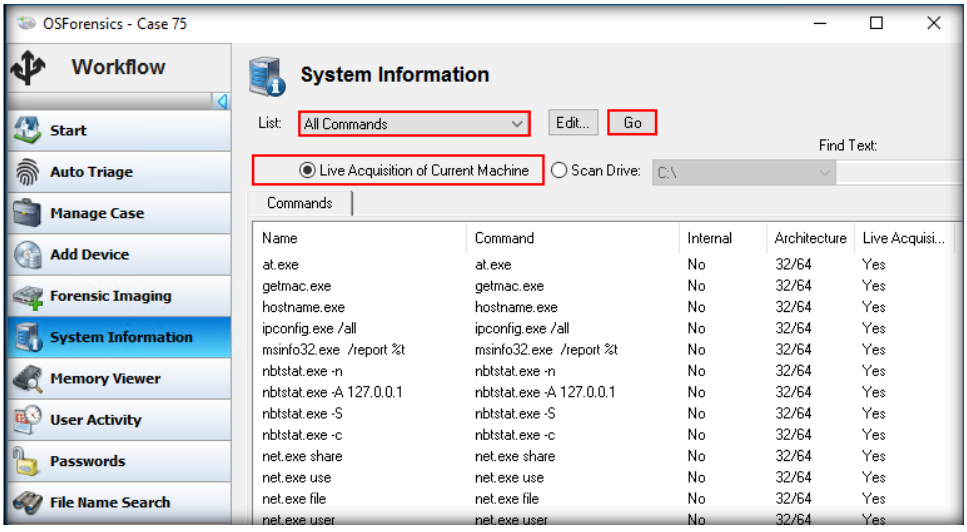


FIGURE 4.38: Select All Commands in the System Information section of OSForensics

New third-party tools can be easily added to the test suite. Many applications can be helpful in retrieving system information. These tools must be installed first if these commands are to run correctly.

Architecture specifies whether this command should be restricted to 32- or 64-bit systems.

Live system specifies whether this command is safe to run during a live acquisition.

53. Since we have selected the **All Commands** option, upon clicking **Go**, the application will display all the commands that were executed on the system and their respective outputs under the **Result 1 – All Commands (Live)** tab, as shown in the following screenshot:

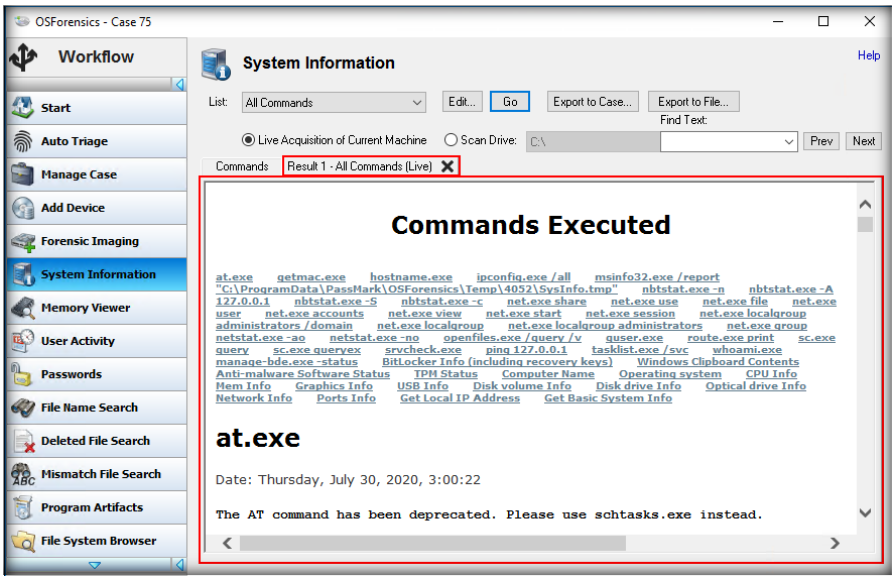


FIGURE 4.39: Display of all executed commands

54. We will now proceed to the task of verifying the integrity of files by calculating their hash values.

TASK 9

Verifying File Integrity

55. To verify the integrity of files by calculating their hash values, click **Verify/Create Hash** from the left pane of the window. The **Verify/Create Hash** section will now appear in the window. Ensure that the **File** option is selected.

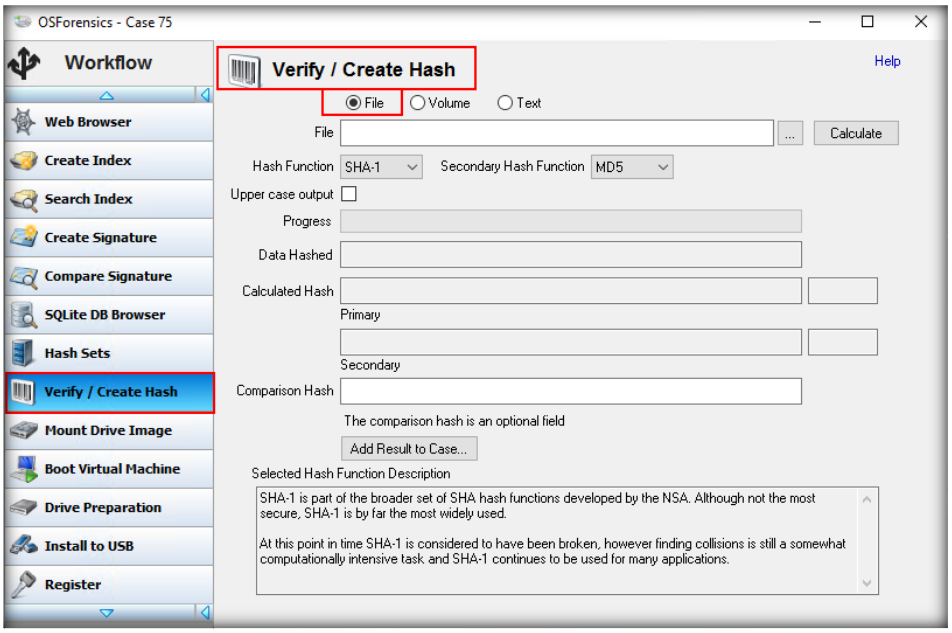


FIGURE 4.40: Verify/Create Hash section in OSForensics

To calculate a hash for a file, simply input the file path, choose one of the available hash functions, and click **Calculate.** To verify the calculated hash with a known hash value, copy the known hash value into the **Comparison Hash** field.

56. Click the **Ellipsis** button to specify a **File** to calculate its hash value. Here, we are specifying the file **Friends2.jpg** from the location **C:\CHFI-Tools\Evidence Files\Image Files\Friends2.jpg** to calculate its hash value. From the **Hash Function** dropdown, select **MD5**, and from the **Secondary Hash Function** dropdown, select **SHA1**. Then, click **Calculate**.

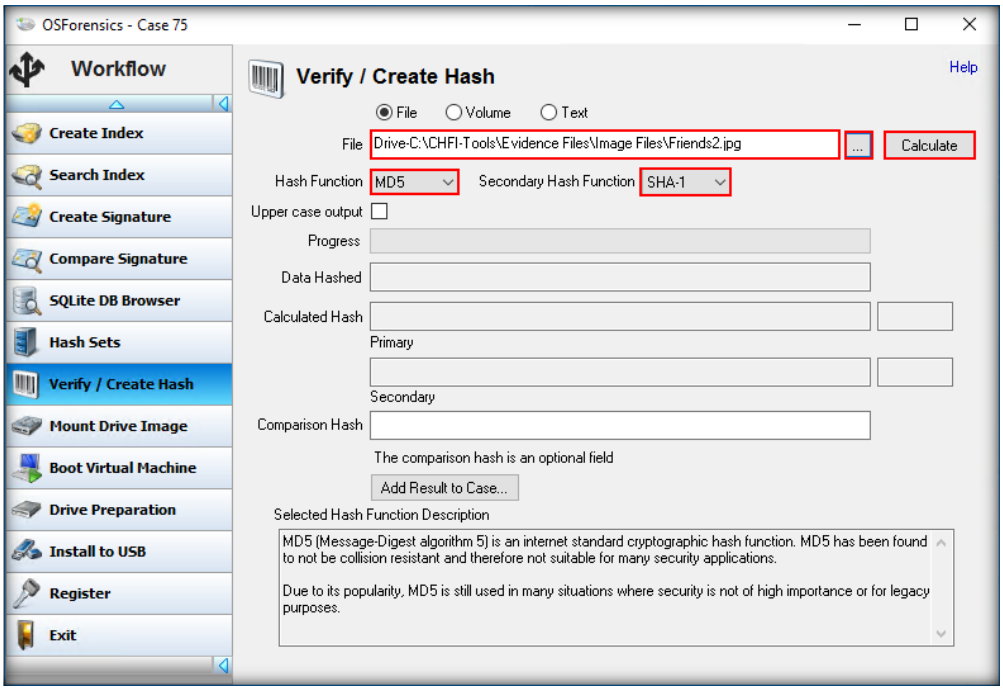


FIGURE 4.41: Providing path for a file

57. The file's hash value appears in the **Calculated Hash** section in both the **Primary Hash Function (MD5)** as well as **Secondary Hash Function (SHA-1)** fields, as shown in the following screenshot:

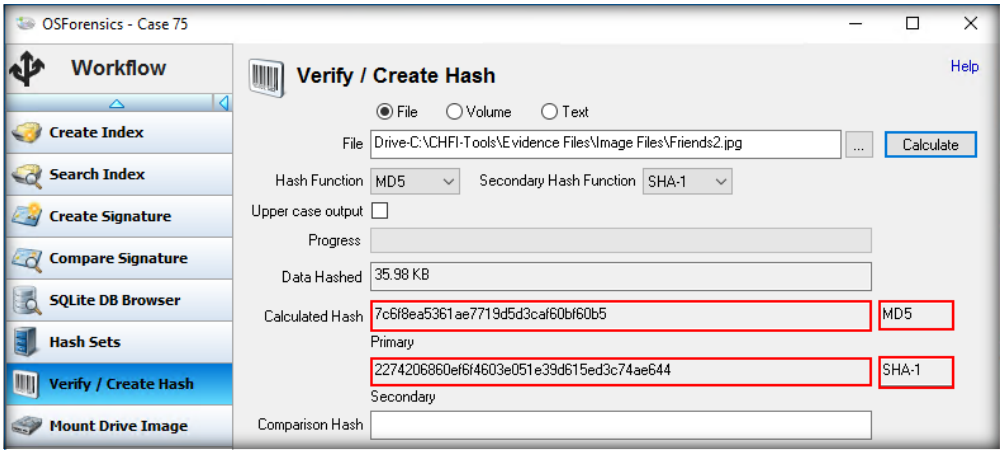


FIGURE 4.42: View generated hashes in OSForensics

To create a hash of a line of text, select the text option and type or paste the text you wish to hash into the text field.

58. To verify the integrity of the **Friends2.jpg** file, we can input its pre-existing hash value in the **Comparison Hash** field. Navigate to **C:\CHFI-Tools\Evidence Files\Image Files** and open the file **Hashes.txt**, which

contains the pre-existing hashes of some of the image files. Copy the hash value for the **Friends2.jpg** file.

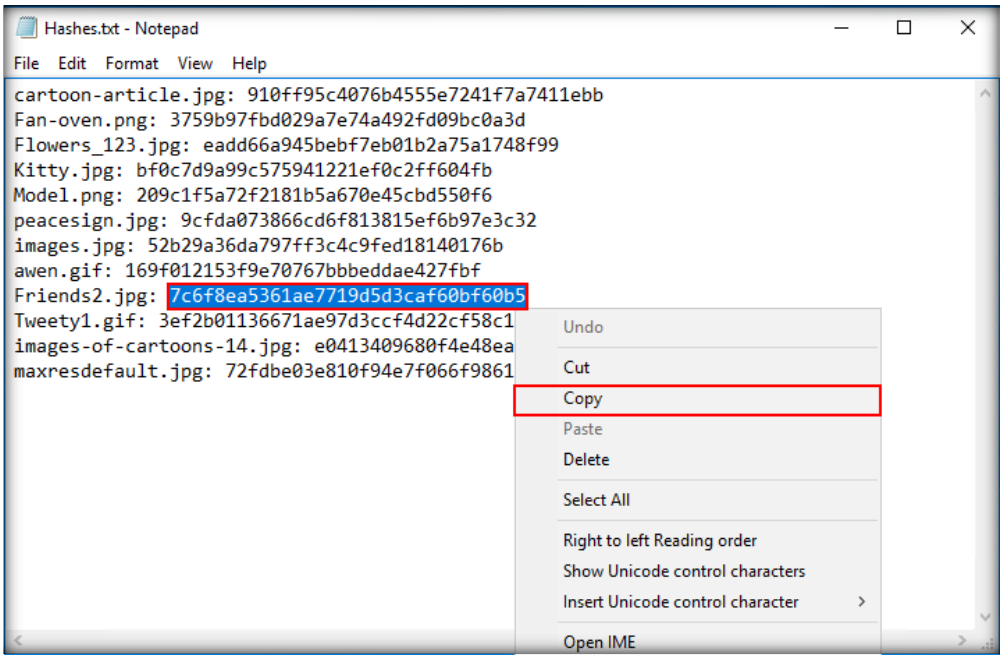


FIGURE 4.43: Copy the hash value for Friends2.jpg

59. Now, paste the copied hash value of the **Friends2.jpg** file in the **Comparison Hash** field. You will see that a tick mark appears against the **Comparison Hash** field, which indicates that the pre-existing and calculated MD5 hash values of the selected file match with each other. This implies that the file's integrity is intact.

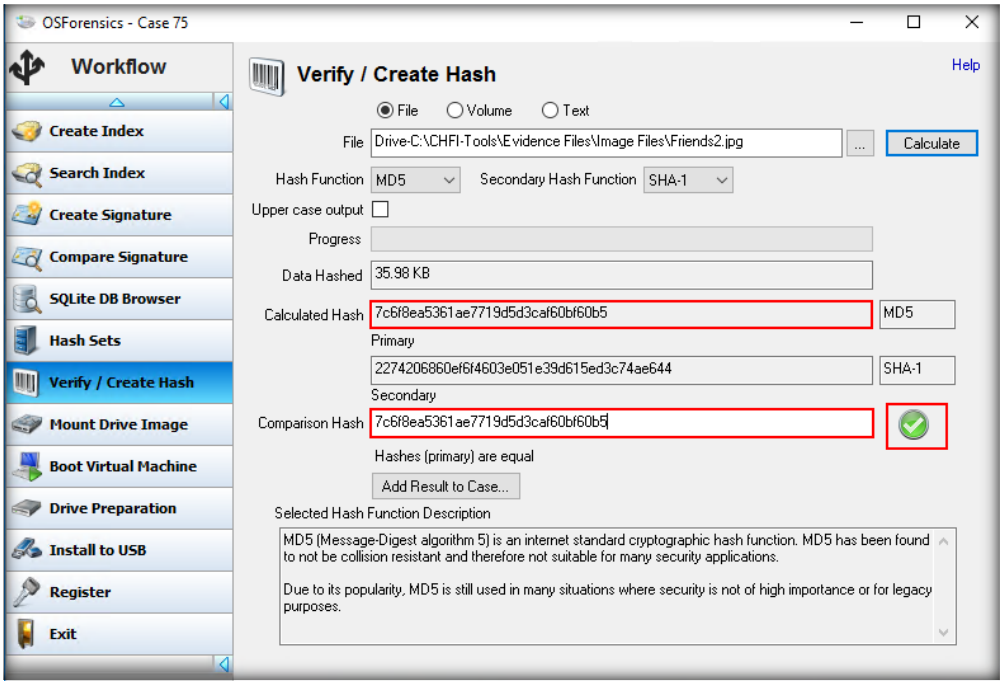


FIGURE 4.44: Matching of MD5 hashes

Note: Conversely, if the file's integrity is not intact, the tool will generate a warning icon in the form of a cross mark, indicating that the file has been tampered with to act as a payload for an attack.

60. In this manner, you can perform investigation on a system or the folders and partitions within the system to gather data of interest to the forensic investigation.

Lab Analysis

Document the complete results of this lab.

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

Lab

5

Extracting Information about Loaded Processes on a Computer

A process is an instance of a computer program that is being executed. Forensic investigators can examine these processes running on a computer for any malicious activity.

ICON KEY

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

Lab Scenario

Processes are instances of computer programs running on a system and contain the code required for activity. Any program or malware will have various methods that will combine to give the result. An investigator should know the default processes to separate them from suspicious ones. To be an expert computer forensics investigator, you must understand how to extract information about loaded processes on a victim computer, which can be of importance during forensic investigation.

Lab Objectives

The purpose of this lab is to help you learn how to investigate loaded processes. In this lab, you will learn how to use Process Explorer.

Lab Environment

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics

This lab requires:

- A computer running **Windows Server 2016** virtual machine
- Administrative privileges to execute commands
- A web browser with internet access
- **Process Explorer** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\Process Explorer**

Note: You can download the latest version of **Process Explorer** from the link https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer

If you are using the latest version of software for this lab, then the steps and screenshots demonstrated in the lab might differ.

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 15 minutes

Overview of the Lab

This lab familiarizes you with the Process Explorer tool and helps you understand how to examine information pertaining to loaded processes on a victim's system.

Lab Tasks

1. Login to the **Windows Server 2016** virtual machine.
2. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\Process Explorer**.
3. Double-click the **procexp.exe** file and accept the license agreement.

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

4. The **Process Explorer** GUI appears, displaying the details of all the processes running on the machine, as shown in the screenshot:

TASK 1

Viewing System Information

Process Explorer also has a powerful search capability that can quickly show which processes have particular handles opened or DLLs loaded.

The unique capabilities of Process Explorer make it useful for tracking down DLL-version problems or handle leaks, and it provides insight into the way Windows and applications work.

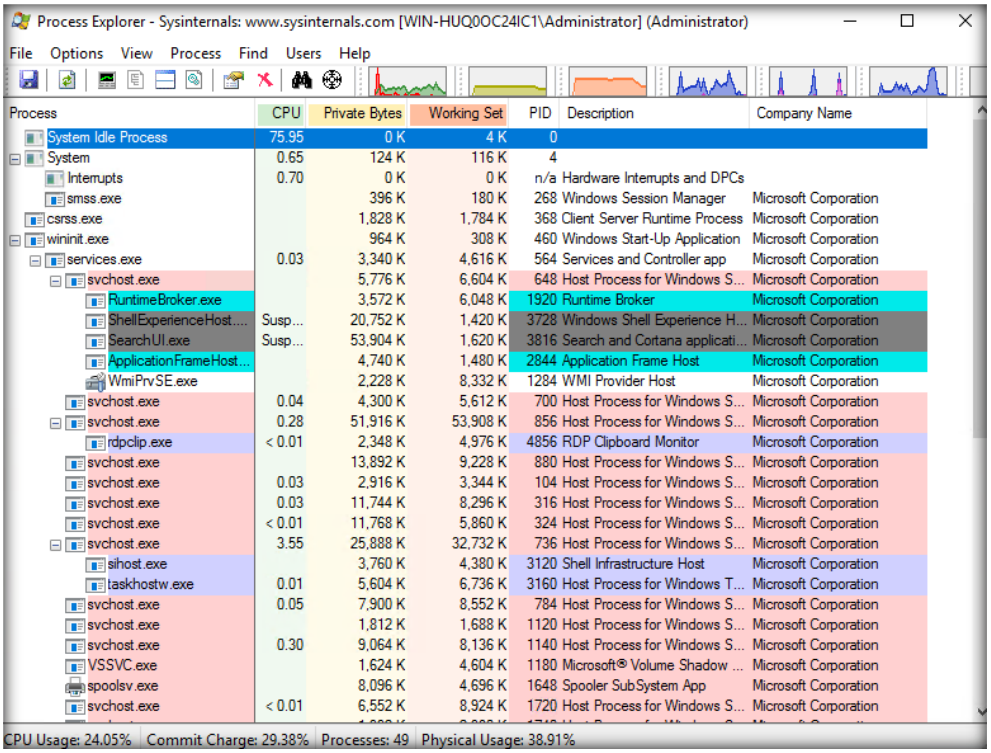


FIGURE 5.1: Main screen of Process Explorer

5. Process Explorer lists all the running processes in the left pane and the details of each process (such as CPU usage and PID) in the right pane.

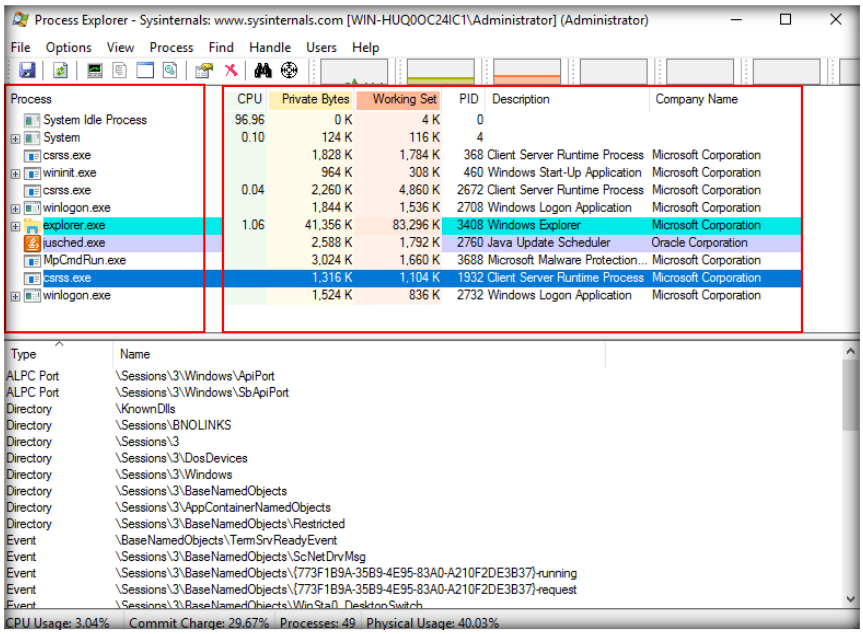


FIGURE 5.2: Process details in Process Explorer

TASK 2

Viewing DLLs

6. To view DLLs, click the **View DLLs** icon  from the toolbar or navigate to **View → Lower Pane View → DLLs** from the **Menu** bar.

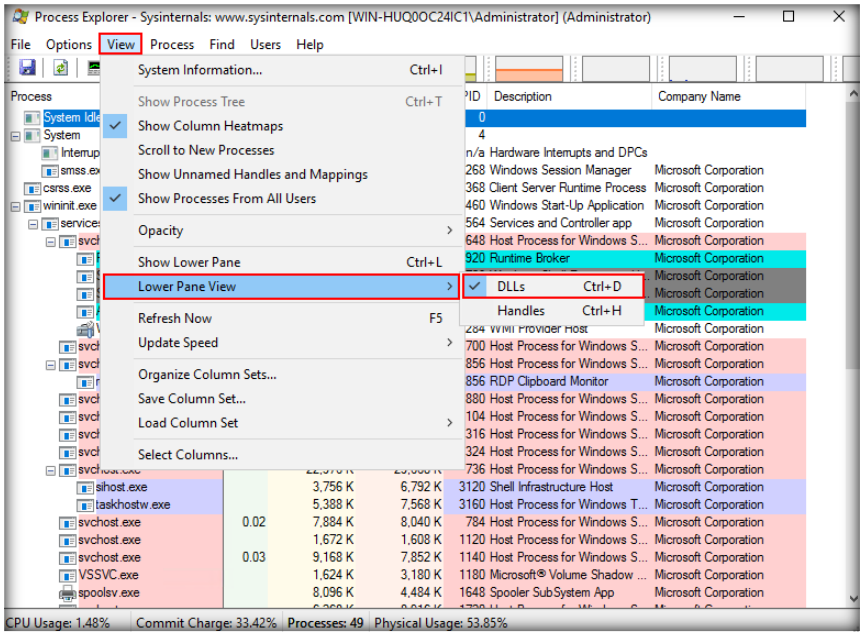


FIGURE 5.3: ViewDLLs in Process Explorer

7. The Process Explorer window will now have an upper pane that lists the processes and their details and a blank lower pane. From the upper pane, **select a process** for which you wish to view the DLLs. Now, you will be able to view the list of DLLs for that process in the lower/bottom pane of the window.

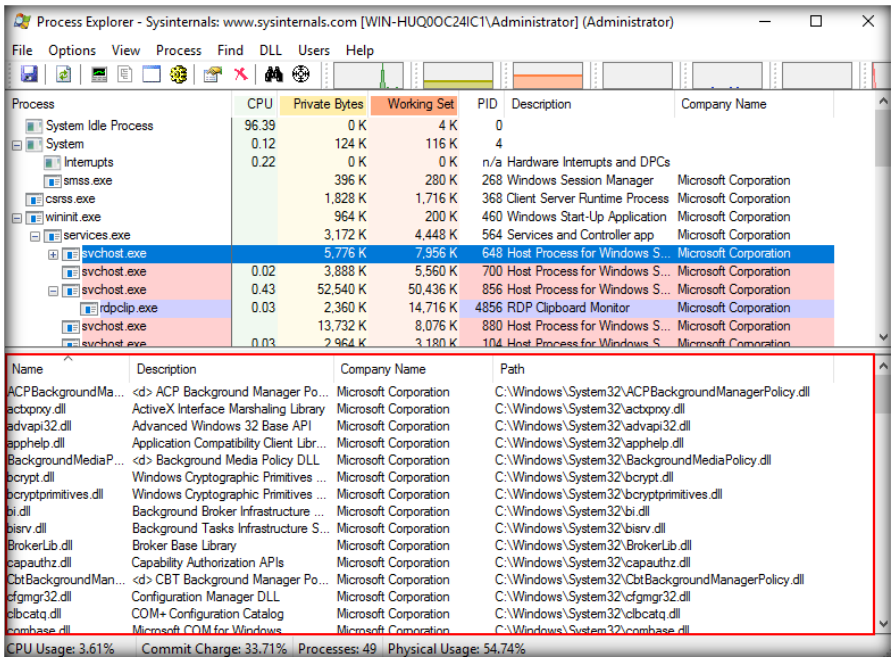


FIGURE 5.4: View DLLs Process Explorer

8. To view DLL properties, select a DLL, choose **DLL → Properties** from the **Menu** bar or right-click the required DLL from the DLL list displayed for a process, and select **Properties...**.

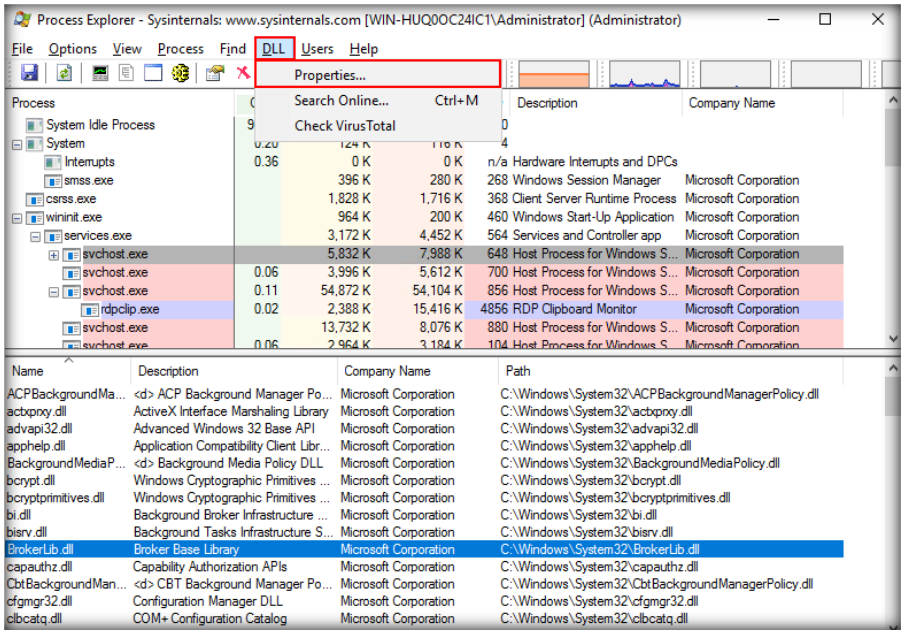


FIGURE 5.5: View DLL properties in Process Explorer

📖 Highlight Relocated DLLs:
When you select the Relocated DLLs entry in the **Options | Configure Highlighting** dialog box, any DLLs that are not loaded at their programmed base address are highlighted in yellow. DLLs that cannot load at their base address because other files are already mapped there are relocated by the loader, which consumes significant CPU resources and makes parts of the DLL that are modified as part of the relocation un-sharable.

📖 On systems that include Terminal Services, Process Explorer displays a **User's** menu that lists the currently connected sessions. Process Explorer creates a menu entry for each session with a name that includes the session ID and the user logged into the session.

9. Performing the above step displays the DLL properties under **Image** and **Strings** tabs.

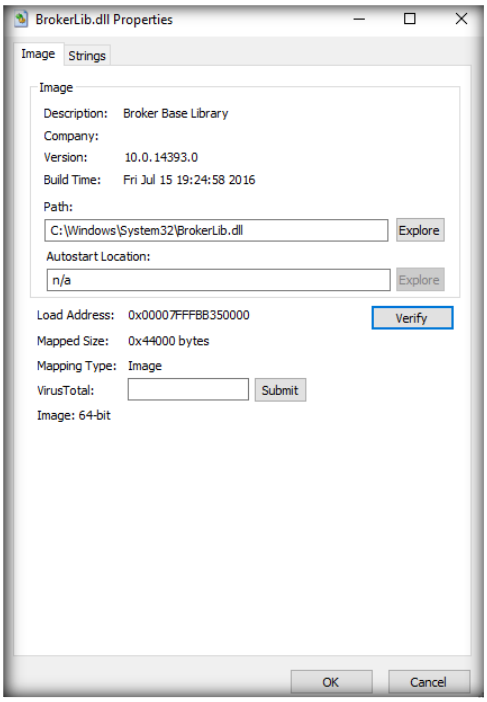


FIGURE 5.6: DLL properties window in Process Explorer

10. The **Image** tab contains details of the DLL such as the company name, version, and path.
11. Click **Verify** to check for the signature of a process.

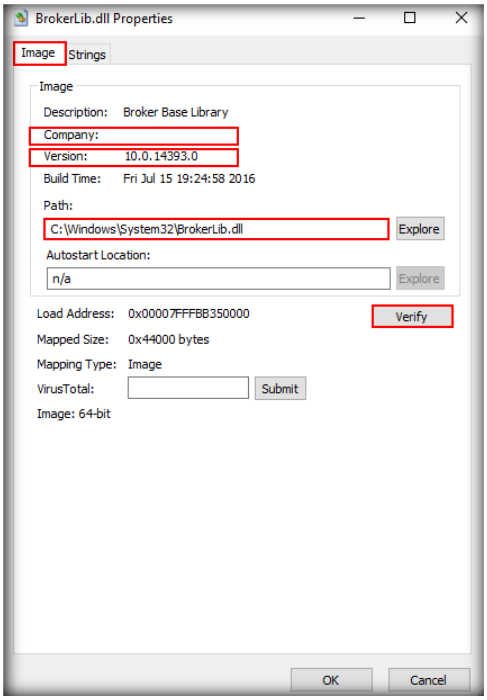


FIGURE 5.7: Image tab of the DLL properties window in Process Explorer

Show Unnamed Handles:
By default, Process Explorer shows only handles to objects that have names. Select the **Show Unnamed Handles** item under the **View** menu to have Process Explorer list all the handles opened by a selected process, including those to nameless objects. Note that Process Explorer consumes significantly more CPU resources when this option is selected.

Mini Graphs:
Process Explorer includes a toolbar and mini graphs for CPU, memory, and I/O history (for Windows 2000 or higher) at the top of the main window. They can be resized with respect to one another or dragged such that each is on a separate row. The mini-graphs show a history of system activity, and moving the cursor over a point on a graph displays a tooltip containing the associated time and process information for a point in time. For example, the tooltip for the mini-CPU graph shows the process that was the largest consumer of CPU resources. Clicking on any of the mini-graphs opens the **System Information** dialog box.

12. On clicking **Verify**, if the company's name is **Microsoft Windows**, then the process is determined as **legitimate**, as shown in the screenshot:

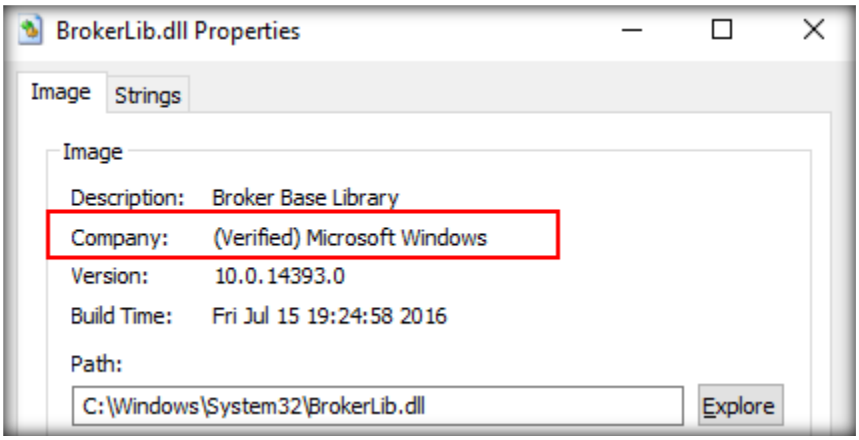


FIGURE 5.8: DLL properties window in Process Explorer showing a verified image signature

13. Now, click the **Strings** tab. This tab lists any Unicode strings found in the selected process. Examining the contents under this tab helps you to find if the process is associated with any malware.

14. When you click the **Strings** tab, you will have two options to examine: the **Image** and **Memory** strings. Click either the **Image** or **Memory** radio button at the bottom of the window to view either the image or memory strings.

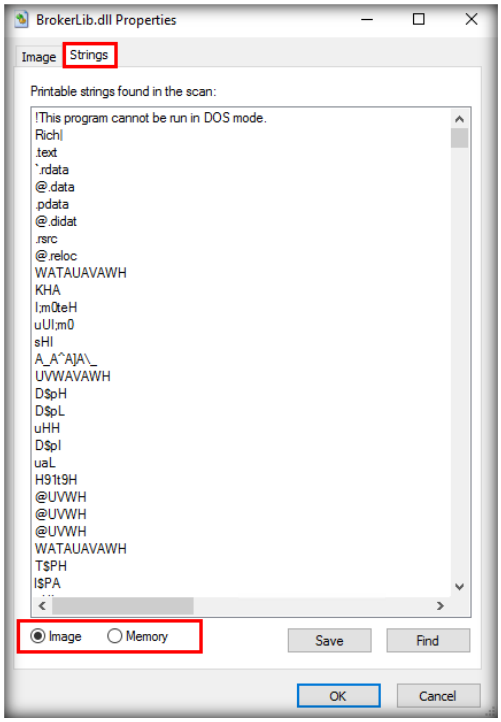


FIGURE 5.9: Strings tab of the DLL properties window in Process Explorer

By default, Process Explorer sorts processes into the system process tree. The process tree reflects the parent–child relationship between processes, where child processes are shown directly beneath their parent and right-indented. Processes that are left-justified are orphans; their parents have exited. To change the sort order, simply click on the column by which you wish to sort. To return the sort order to the process tree, choose **View → Show Process Tree** and click the **Process Tree** toolbar button or press **Ctrl + T**.

15. You can also save the **Image** and/or **Memory** strings in text file format. In this lab, we are saving the **Image** strings. To save **Image** strings, ensure that the **Image** radio button is selected and then click the **Save** button.

16. On clicking **Save**, the **Save Process Explorer Strings...** window appears. Specify a location (here, **Desktop**) to save the file and click **Save**.

On Windows NT–based systems, Process Explorer shows two artificial processes: Interrupts and DPCs. These processes reflect the amount of time the system spends in servicing hardware interrupts and Deferred Procedure Calls (DPCs), respectively. High CPU consumption by these activities can indicate a hardware problem or device driver bug. To view the total number of interrupts and DPCs executed since the system booted, add the **Context Switch** column. Another potentially useful metric is the number of interrupts and DPCs generated per refresh interval, which can be viewed upon adding the **CSwitches Delta** column.

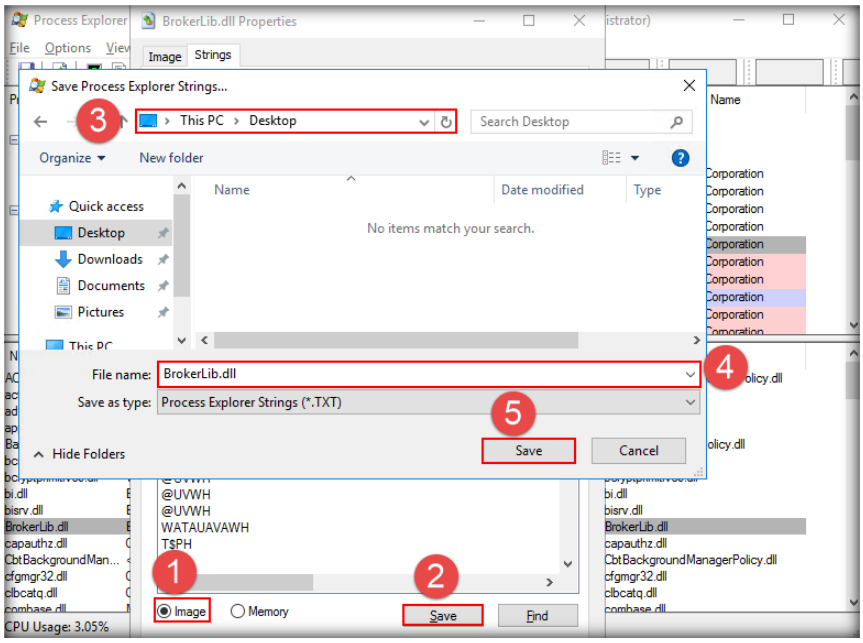


FIGURE 5.10: Save DLL strings in Process Explorer

17. On saving the file, click **OK** in the DLL's **Properties** window to close it

18. The **Search Online** option searches the selected DLL on the internet by launching an **Internet Browser**

19. To search online, choose a DLL, right-click on it and select **DLL → Search Online...** from the **Menu** bar

Find Windows Process:
You can highlight the process that owns a window visible on the desktop by dragging the target-like toolbar button over the window in question. Process Explorer will select the corresponding process entry in the process view.

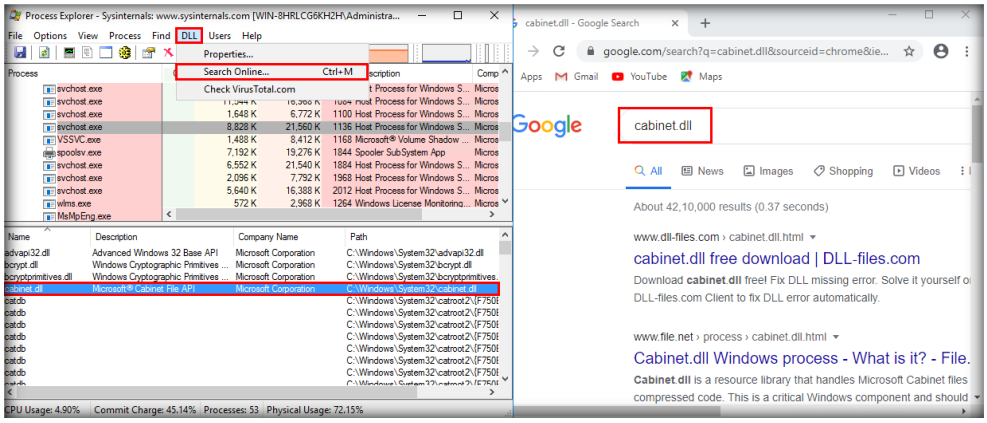


FIGURE 5.11: Search DLLs online in Process Explorer

20. To view the handles of a process, select the process and choose **View → Lower Pane View → Handles** from the **Menu** bar.

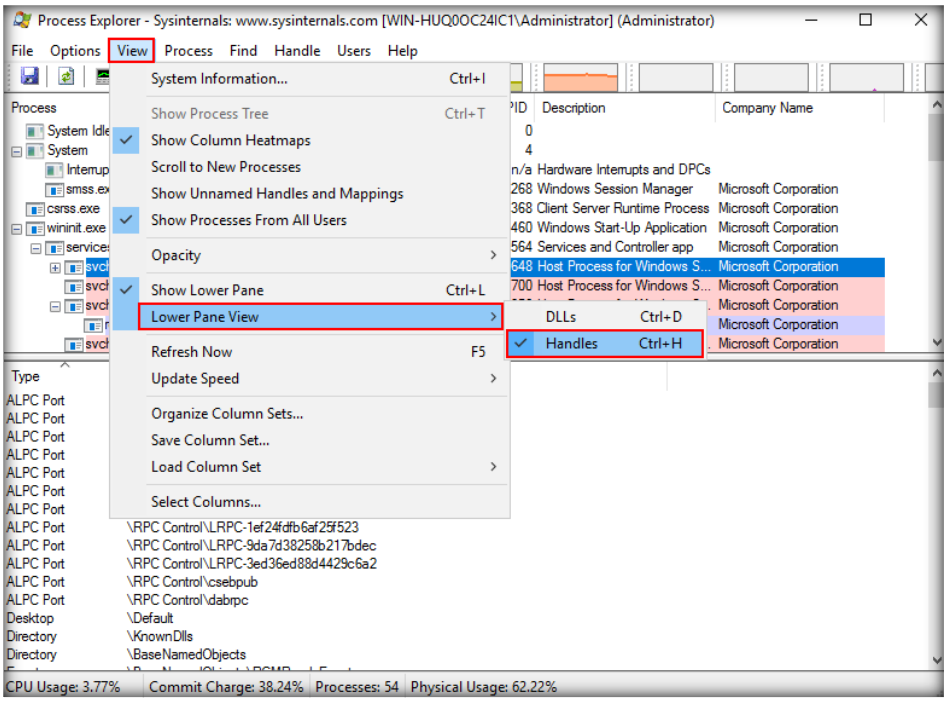


FIGURE 5.12: View handles in Process Explorer

Highlight Services: On Windows NT and higher, this option has Process Explorer show processes that are running Win32 services in the service process highlight color. The **Services** tab of the **Process Properties** dialog box shows the list of services running within a process.

21. To view handle properties, right-click the required handle and select **Properties**, or select the required handle and then choose **Handle → Properties** from the **Menu** bar.

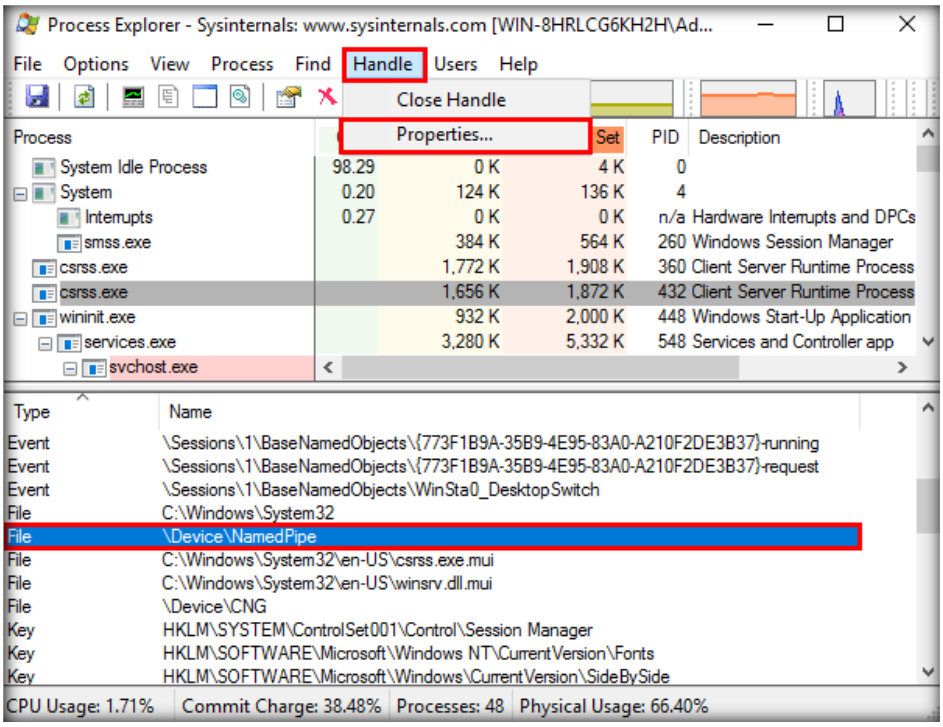


FIGURE 5.13: View handle properties in Process Explorer

Highlight .NET Processes: This option appears on Windows NT-based systems that have the .NET Framework installed. When the option is checked, managed applications (those that use the .NET Framework) are highlighted in the .NET process highlight color.

22. The Properties window appears for the selected handle. The **Details** tab displays basic information about the selected handle.

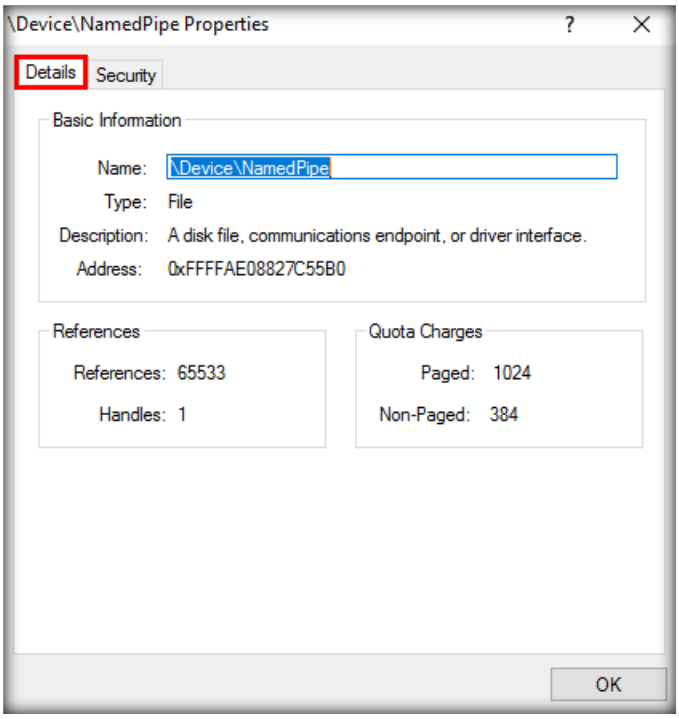


FIGURE 5.14: Details tab of the properties window for a handle in Process Explorer

Kill: This item terminates a process with the **Terminate Process** API. Note that a process terminated in this manner is not warned of its termination; therefore, it does not write any unsaved data.

23. The **Security** tab displays the level of security assigned to each group or user for the selected handle.

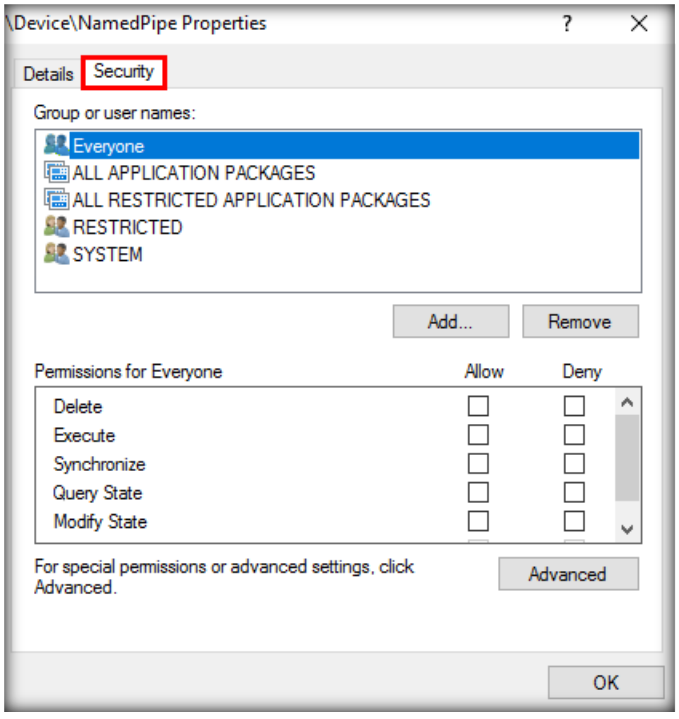


FIGURE 5.15: Security tab of the properties window for a handle in Process Explorer

Highlight Own Processes: In Windows NT and higher, checking this option results in Process Explorer showing the processes that are running in the same user account as Process Explorer in the own-process highlight color.

24. Click **OK** to close the **Properties** window.
25. In this manner, you can view all the processes as well as their associated DLLs and handles using Process Explorer. During forensic investigations, this tool is highly useful for examining a process (any malicious process, if found) and analyzing its associated DLL files and handles.

Lab Analysis

Analyze the DLLs and handles in the process and document the respective details.

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

Lab

6

Viewing, Monitoring, and Analyzing Events Occurred on a Windows Machine

Windows event logs contain useful information related to a system, its security, the applications on it, etc. Detailed examination of these logs helps investigators discover potential artifacts and construct a timeline for the analysis of events based on the logging information.

ICON KEY

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

Lab Scenario

Few attackers have infiltrated the security of a number of systems at the office of a product-based technology company. The attackers made certain changes to the hardware configurations of the systems, services running on the systems, operating systems, and some other critical programs running on those systems to steal sensitive information. The company sought the help of an independent digital forensic investigation agency to solve the case and secure all its systems.

Forensic investigators from the agency must now examine all the event logs pertaining to the affected systems, which include security logs, system logs, and application logs, to understand and analyze how this case of cyber-crime was perpetrated.

As an expert forensic investigator, you must know to examine the event logs of Windows systems.

Lab Objectives

The objective of this lab is to help forensic investigators learn how to view, monitor, and analyze various event logs. Here, we are demonstrating the analysis of security logs. Using this as an example, other log files, such as application logs and system logs can also be examined.

Lab Environment

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics

This lab requires:

- A computer running **Windows Server 2016** virtual machine
- Administrative privileges to execute commands
- A web browser with internet access
- **Event Log Explorer** installer located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\Event Log Explorer**

Note: You can download the latest version of **Event Log Explorer** from the link www.eventlogxp.com.

If you are using the latest version of software for this lab, then the steps and screenshots demonstrated in the lab might differ.

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 15 minutes

Overview of the Lab

This lab helps you understand how to examine Windows event logs using **Event Log Explorer**.

Lab Tasks

TASK 1

Installing and Launching Event Log Explorer

1. Login to the **Windows Server 2016** virtual machine.
2. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\Event Log Explorer** and you will see **elex_setup.exe** file.
3. Double-click **elex_setup.exe** to launch the setup. Select the language as **English**, accept the license agreement, and follow the wizard-driven installation steps to install the application.

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

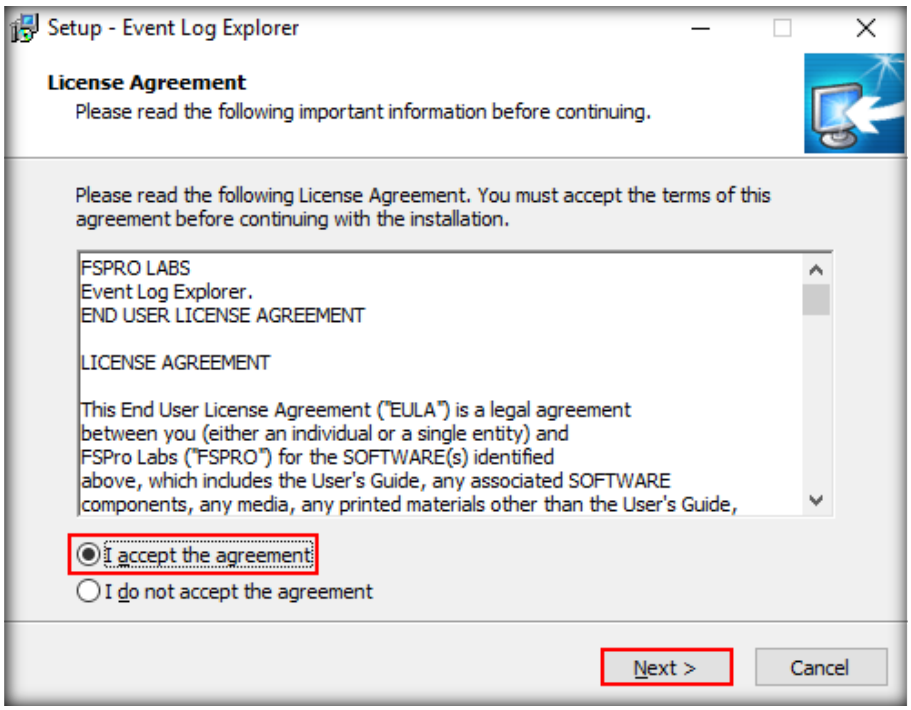


FIGURE 6.1: Event Log Explorer Setup Wizard

Note: If the setup wizard for the tool gives you an option to install the additional component **Elodea**, then select the option **No, do not install Elodea**, and then click **Next**.

4. In the final installation step, check **Launch Event Log Explorer** and click **Finish**.

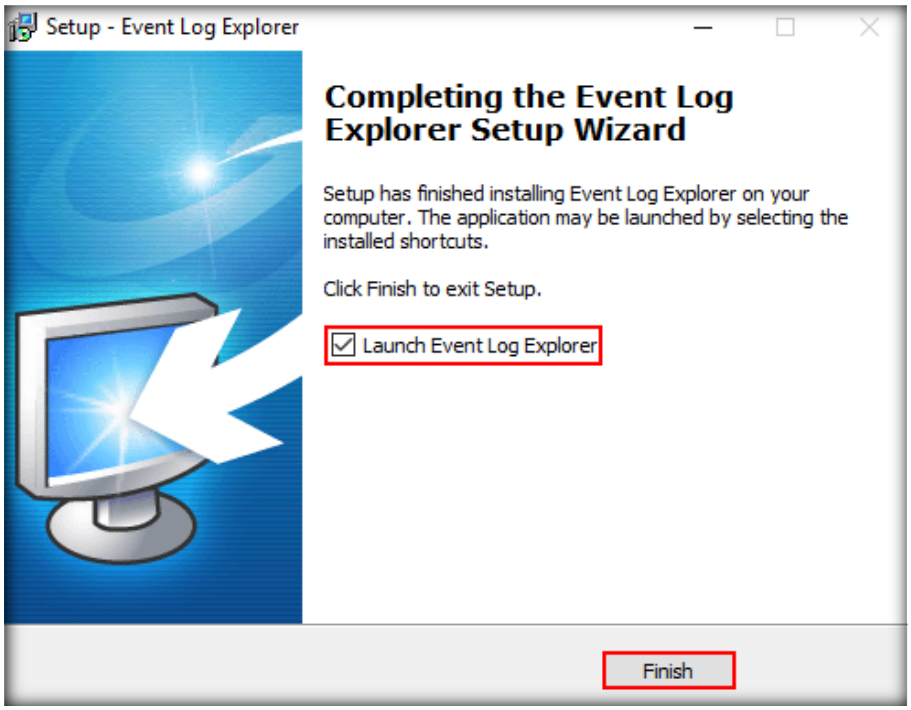


FIGURE 6.2: Finish the setup for Event Log Explorer

5. On completing the installation, a browser window automatically opens, displaying the **eventlogxp.com** web page. Close the browser window. Apart from the browser window, you will also see an **Event Log Explorer** pop-up appearing after completing the tool installation; in this pop-up, ensure that the **Continue evaluation** option is selected and then click **OK**.

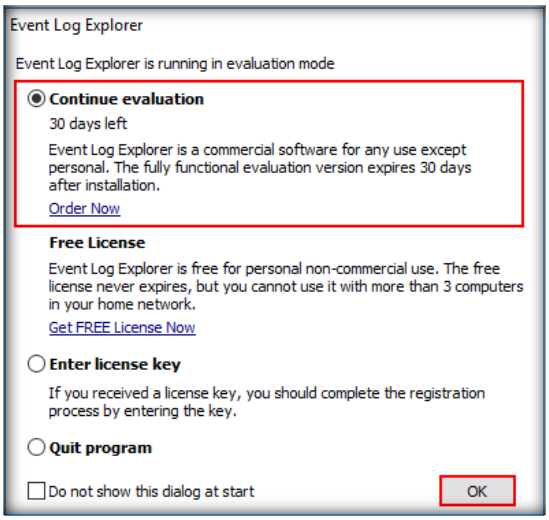


FIGURE 6.3: Select Continue evaluation option in Event Log Explorer pop-up

Event Log Explorer helps you to quickly browse, find, and report on problems, security warnings, and all other activities created within Windows.

6. The main window of **Event Log Explorer** opens, displaying an empty log view area and a **Computer Tree** pane on the left, which shows the name of your **Windows Server 2016** virtual machine at the top.

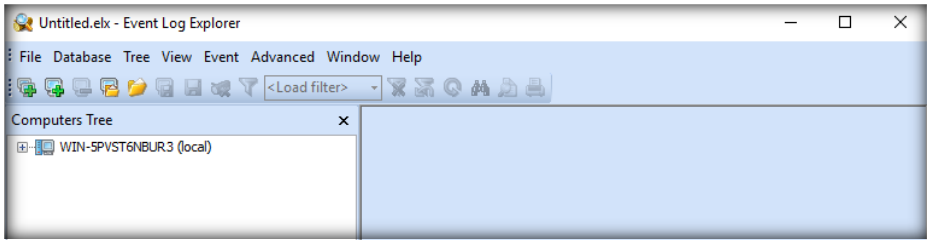


FIGURE 6.4: Main window of Event Log Explorer

Event Log Explorer provides a powerful event search and filtering engine.

7. We will be using the evidence file **Brute Force.evtx** to examine the event logs that have been recorded on the host machine (here, **Windows Server 2016** virtual machine).

TASK 2

Uploading the Evidence File (Event Log File)

8. Click on the **Open Event Log File** icon at the top of the window under the menu bar, as highlighted in the screenshot below:

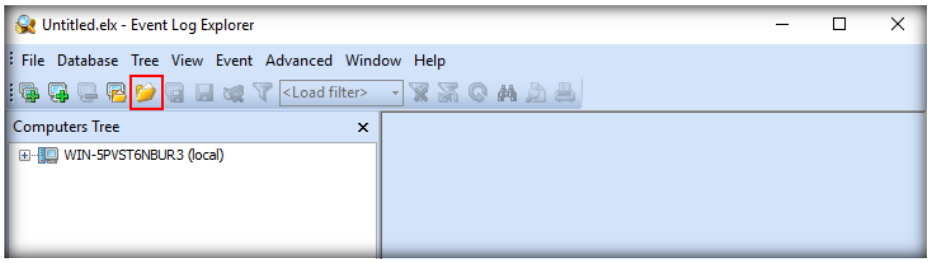


FIGURE 6.5: Click Open Event Log File icon

9. An **Open Backup Event Log File** window appears. Navigate to **C:\CHFI-Tools\Evidence Files\Event Logs**, select the file **Brute Force.evtx**, and then click **Open**, as indicated in the following screenshot:

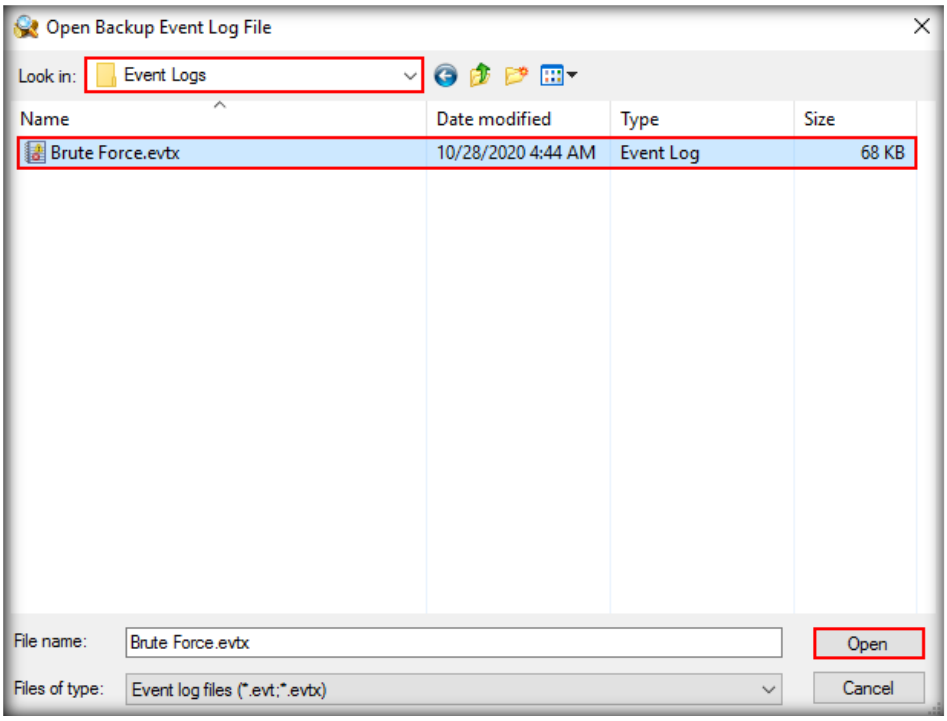
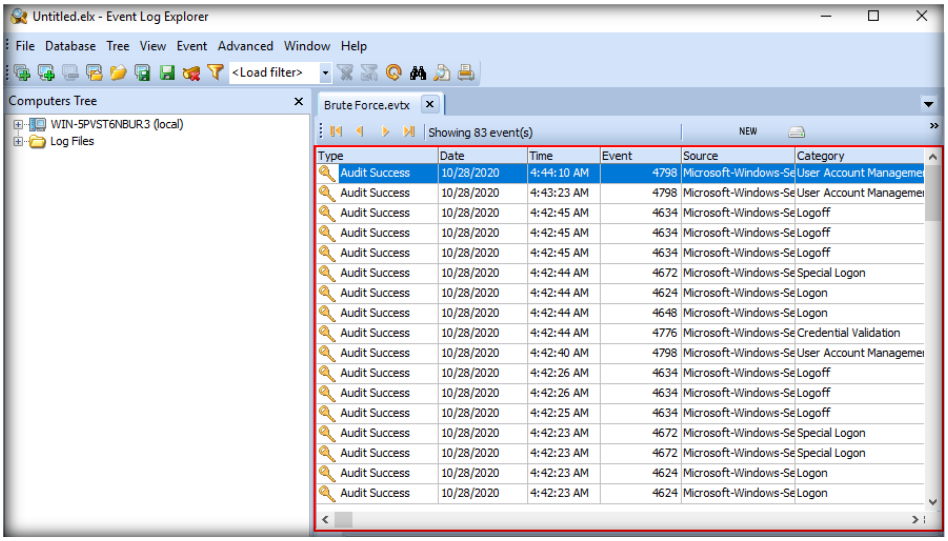


FIGURE 6.6: Select Brute Force.evtx evidence file and click Open

10. The **Event Log Explorer** application will now load the selected log file and display its log entries in the right pane of the application window, as indicated in the following screenshot:



Event Log Explorer can print event logs or even separate events. You can also export event logs to other formats. At the time of writing, Event Log Explorer supports exports to HTML, Microsoft Excel, and tab-separated text files.

FIGURE 6.7: View security events in Event Log Explorer

Note: The security events displayed in the right pane of the application window are auto arranged in such a manner that the latest events are listed at the top; on scrolling down the events list, you can find older events. Our

Main features and benefits of Event Log Explorer at a glance:

- Favorite computers and their logs are grouped into a tree.
- Event logs can be backed up manually and automatically.
- Event descriptions and binary data are displayed in the log window.
- Advanced filtering can be performed by any criteria, including event description text.
- The Quick Filter feature allows you to filter an event log with two mouse clicks.
- Log loading options can be set to pre-filter event logs.
- Logs can be color coded by Event ID.
- Logs can be printed and exported to different formats.
- Damaged EVT files can be read, and EVT files can be generated from event views.

objective here is to search for security event logs related to login events. Hence, we will search for events that reveal login attempts: both successful and unsuccessful.

11. If you wish to know what a particular security event is about, select it. The description pertaining to that event will be displayed in the **Description** pane in the lower section of the window. Here, on selecting one of the security event logs, we found that the system attempted to validate the credentials for the **Administrator's** account.

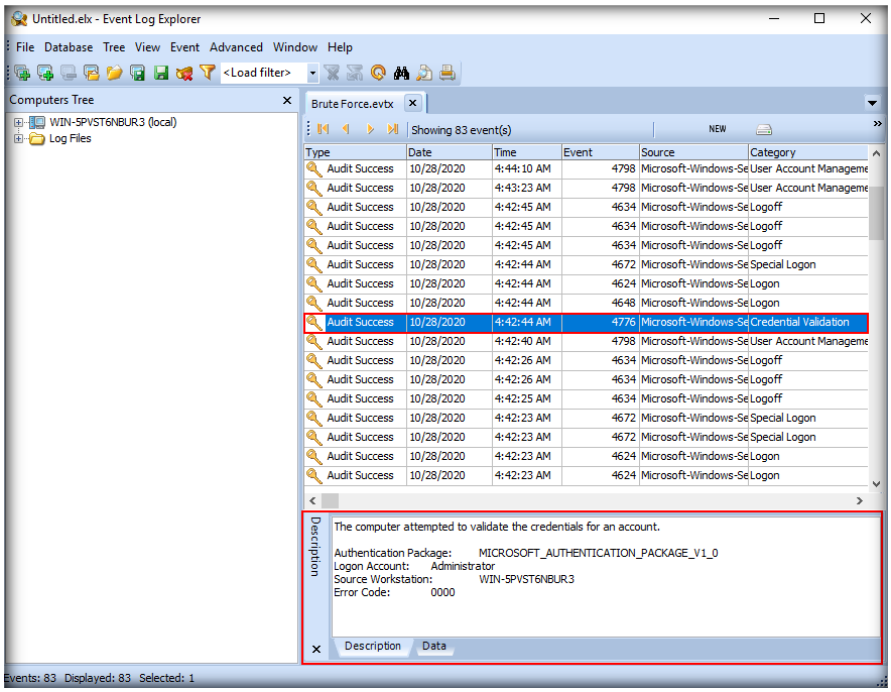


FIGURE 6.8: Select an event to view its description in the Description pane of Event Log Explorer

TASK 3

Examining Audit Failure Events

12. Now, scroll down the events. You should be able to see a series of **Audit Failure** events, as shown in the following screenshot:

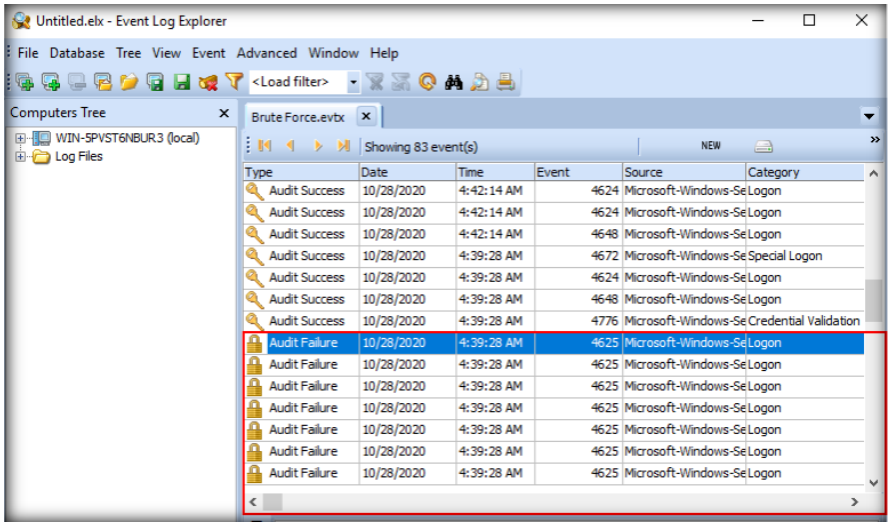


FIGURE 6.9: Scroll down to Audit Failure events in Event Log Explorer

13. To view the details associated with an **Audit Failure** event, double-click on the event. The tool will display the details of the event in an **Event Properties** window. Here, we double-clicked on the first Audit Failure event listed at the top of the stack in the screenshot above. The **Event Properties** window shows a failed login attempt, which is recorded with the **Event ID 4625**.

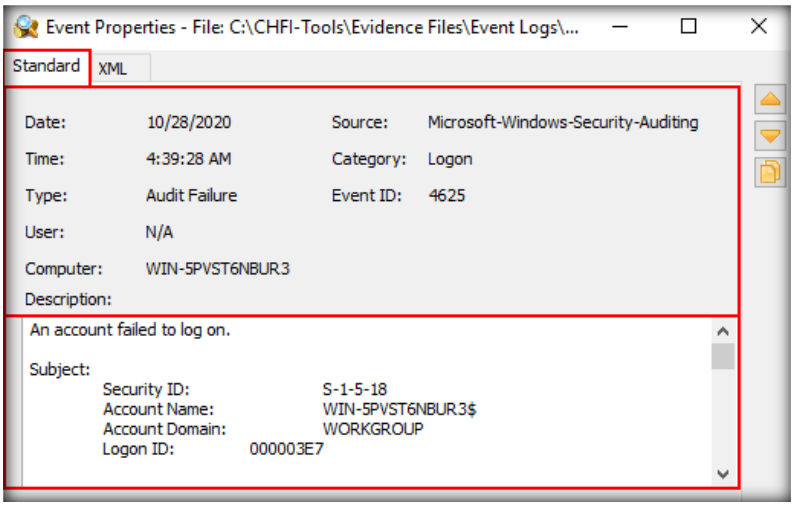


FIGURE 6.10: Details related to the selected event

Note: If there are multiple **Audit Failure** events showing failed login attempts and if they are followed by **Audit Success** events at the top showing successful login attempts, then this indicates that an attacker might have attempted to login to the host/local machine using **brute force** attack techniques.

14. In a real-time scenario, the number of events recorded can be very large, and manually examining each event and identifying suspicious/malicious events can be rather difficult.

15. So, we will apply a filter to the security events to check for all the failed login attempts. To apply the filter, close the **Event Properties** window and then click the **filter** icon below the menu bar, as highlighted in the following screenshot:

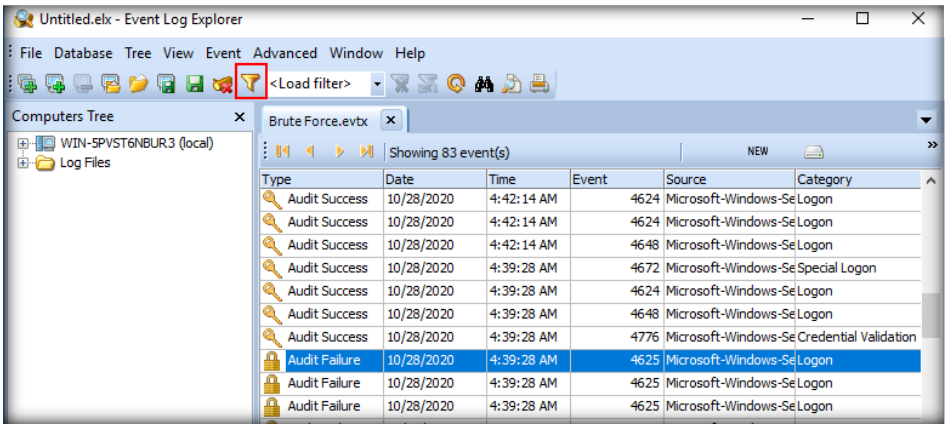


FIGURE 6.11: Click the Filter icon

16. A **Filter** window will open. In this window, leave only the **Audit Failure** option checked while unchecking all other options in the **Event types** section. Then, click **OK**.

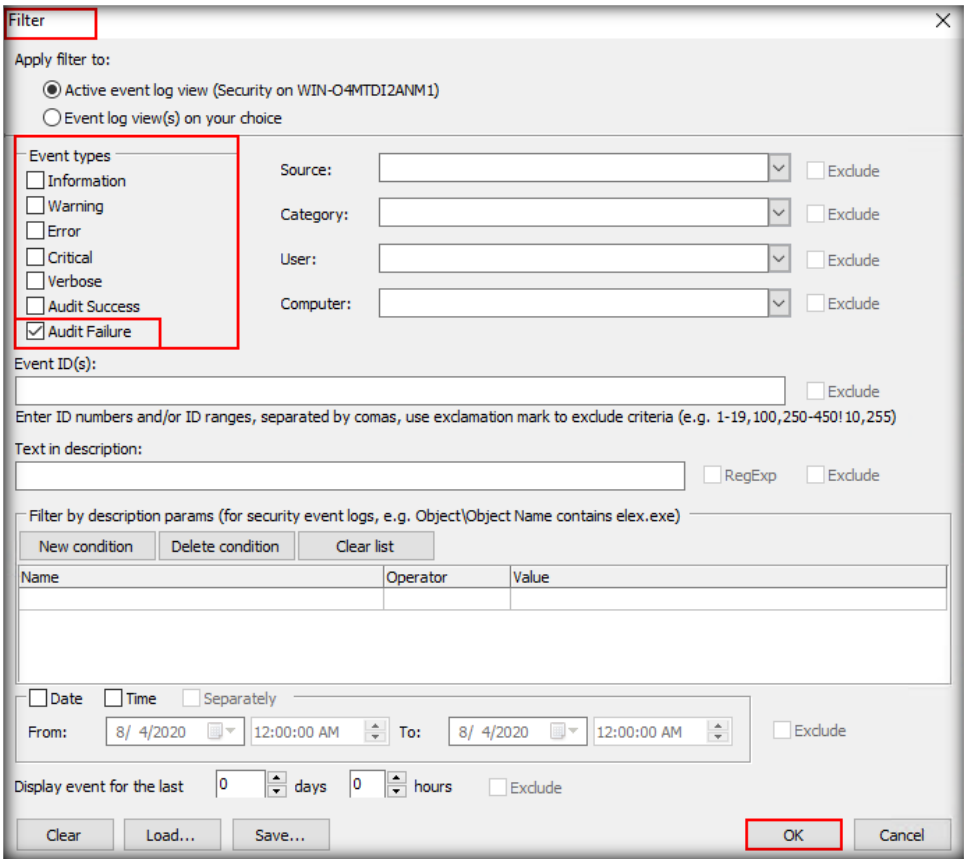


FIGURE 6.12: Apply filter settings

17. The application will now display only the **Audit Failure** events in the right pane of the tool window. You can view the description of the selected event in the **Description** pane. The **Description** pane, as seen in the screenshot below, displays the following:

    i) The first line shows a message that reads **An account failed to log on**, which implies that a user attempted to login to the local machine but the login attempt failed.

    ii) The **Subject** section contains the **Account Name** of the local machine (**WIN-5PVST6NBUR3**) on which the login attempt was made.

    iii) The **Account For Which Logon Failed** section contains the **Account Name** as **Administrator**, which is the username on the local machine. Thus, we can infer that the account with the username **Administrator** was targeted.

Event Log Explorer's friendly and powerful user interface lets you choose between two styles: multi-document or tabbed-document interfaces.

Depending on the user interface style, log views are presented either as MDI child windows or as tabs.

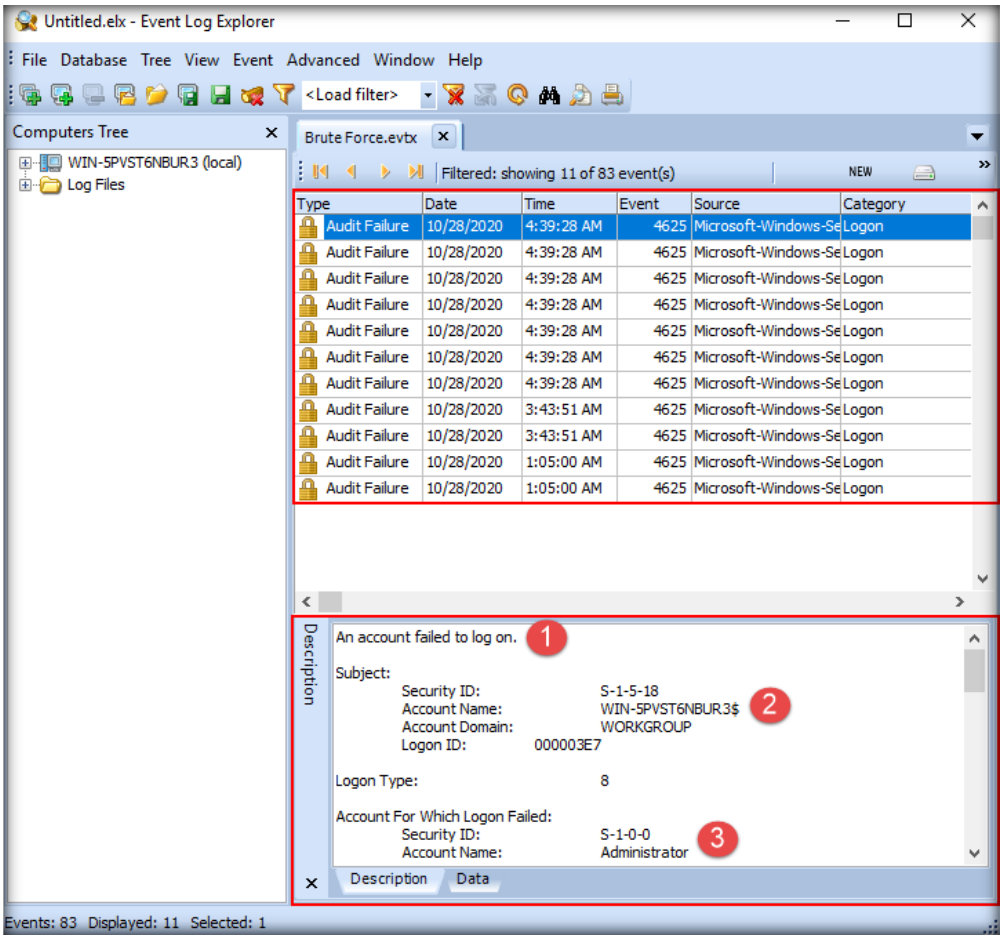


FIGURE 6.13: Failed login attempt

18. In the screenshot above, notice that the first seven **Audit Failure** events from the top bear the same timestamp, i.e., **4:39:28 AM** (The timestamps observed in your lab environment might differ from those demonstrated in this lab). Therefore, when we examine the details of each of the six other **Audit Failure** events with the same timestamp in the **Description** pane by sequentially selecting each, they reveal a failed login attempt, as seen in the step above.

Note: For demonstrative purposes, we have shown only the contents of the top-most **Audit Failure** event in the above step. In real-time, as an investigator, you must check for artifacts of unsuccessful login attempts in each of those **Audit Failure** event logs, which reflect the same timestamp or have occurred in a very short timeframe.

19. Now, we will examine the **Audit Success** events, keeping in mind the timestamps they reflect.

TASK 4

Examining Audit Success Events

20. So, click on the filter icon at the top. The **Filter** window will appear as seen previously. In the **Filter** window, check only the **Audit Success** option while leaving all other options under the **Event Types** section unchecked. Then, click **OK**.

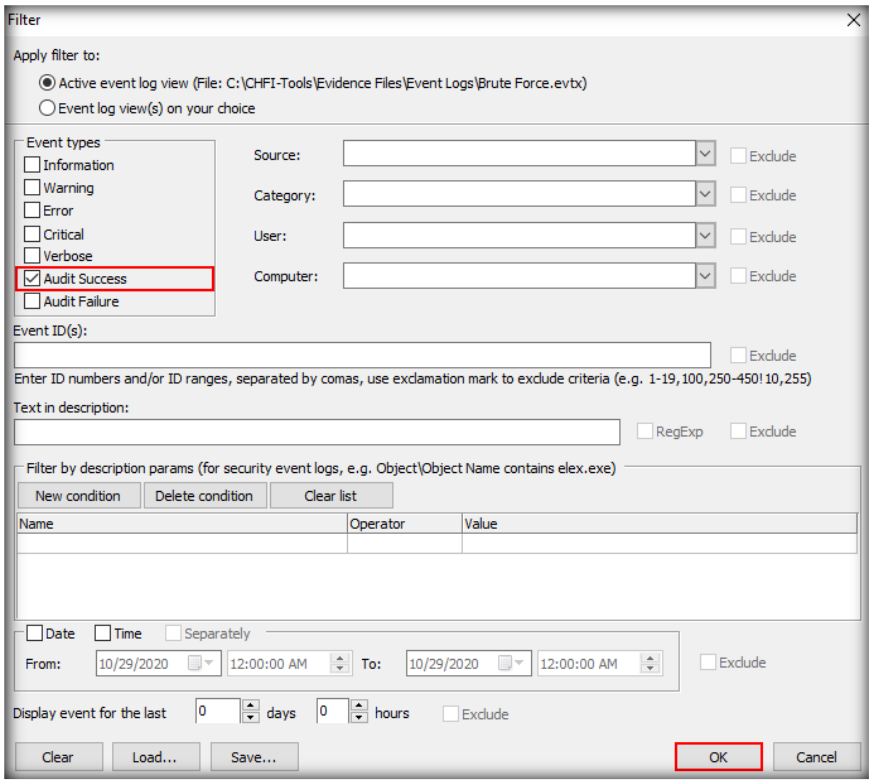


FIGURE 6.14: Keep Audit Success checked and click OK

21. The application will now list all the **Audit Success** events in the right pane of the window, as shown in the following screenshot:

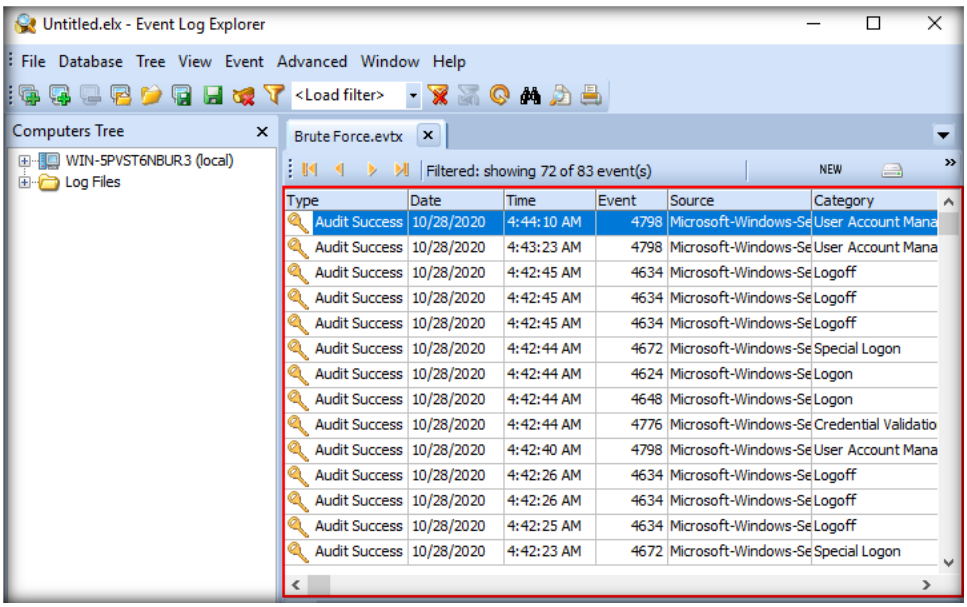


FIGURE 6.15: Examining Audit Success events

22. From our previous observation, we know that the multiple **Audit Failure** events revealing failed login attempts have the timestamp **4:39:28 AM**. Now, we need to begin examining those **Audit Success** events that start either at or after **4:39:28 AM**.

23. Scroll down in the right pane of the window to view **Audit Success** events that have the **4:39:28 AM** timestamp. We need to begin examining these event logs starting with the first **Audit Success** event that bears the timestamp **4:39:28 AM**.

24. Select the first **Audit Success** event with the Event ID **4776**. Its **Description** pane reveals that the computer attempted to validate the login credentials for the account **Administrator** on the local machine (identified as **WIN-5PVST6NBUR3**), as indicated in the screenshot below. This implies that an attempt to login to the local machine by a user has been validated.

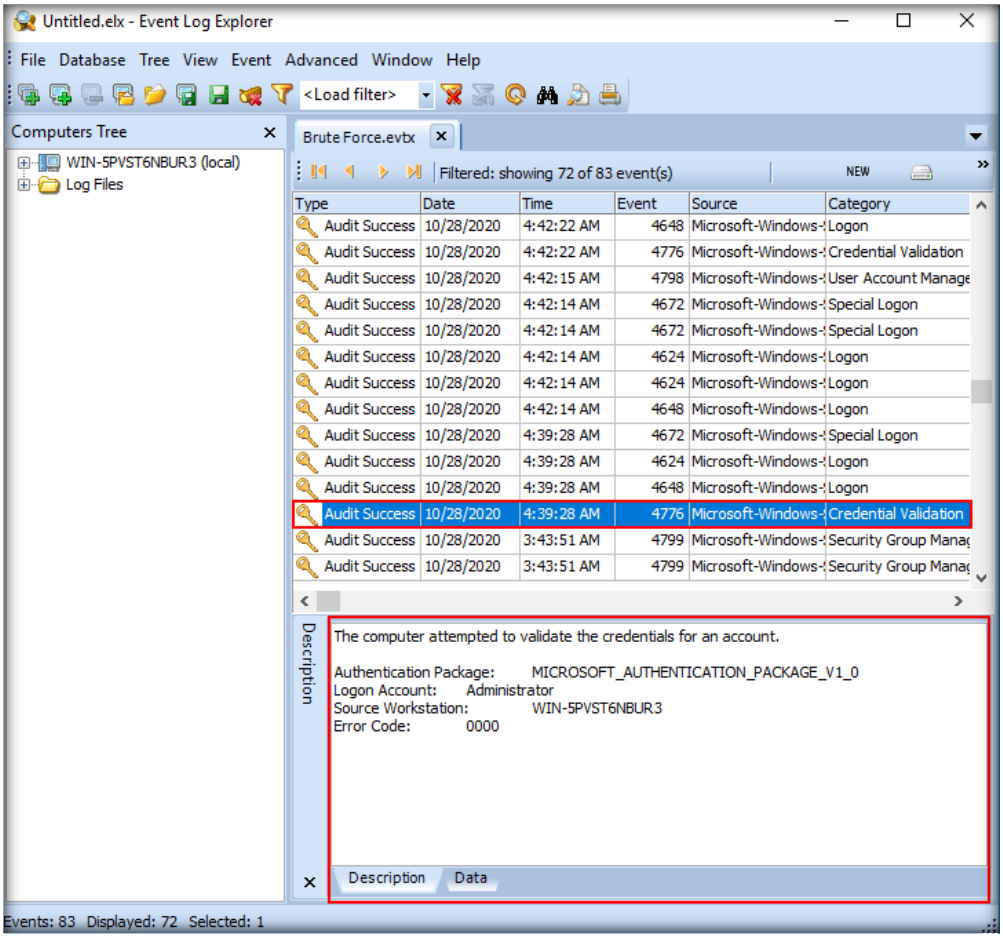


Figure 6.16: Account login validation attempt in an Audit Success event

25. Now, select the **Audit Success** event that is above the currently selected event and holds the same timestamp as mentioned previously. Its **Description** pane reveals that a logon was attempted using explicit credentials on the local machine, as indicated in the screenshot below. This implies that a logon attempt using clear, plaintext login credentials might have been made on the local machine. This might occur when an attacker/malicious user has got hold of the login credentials for a local machine.

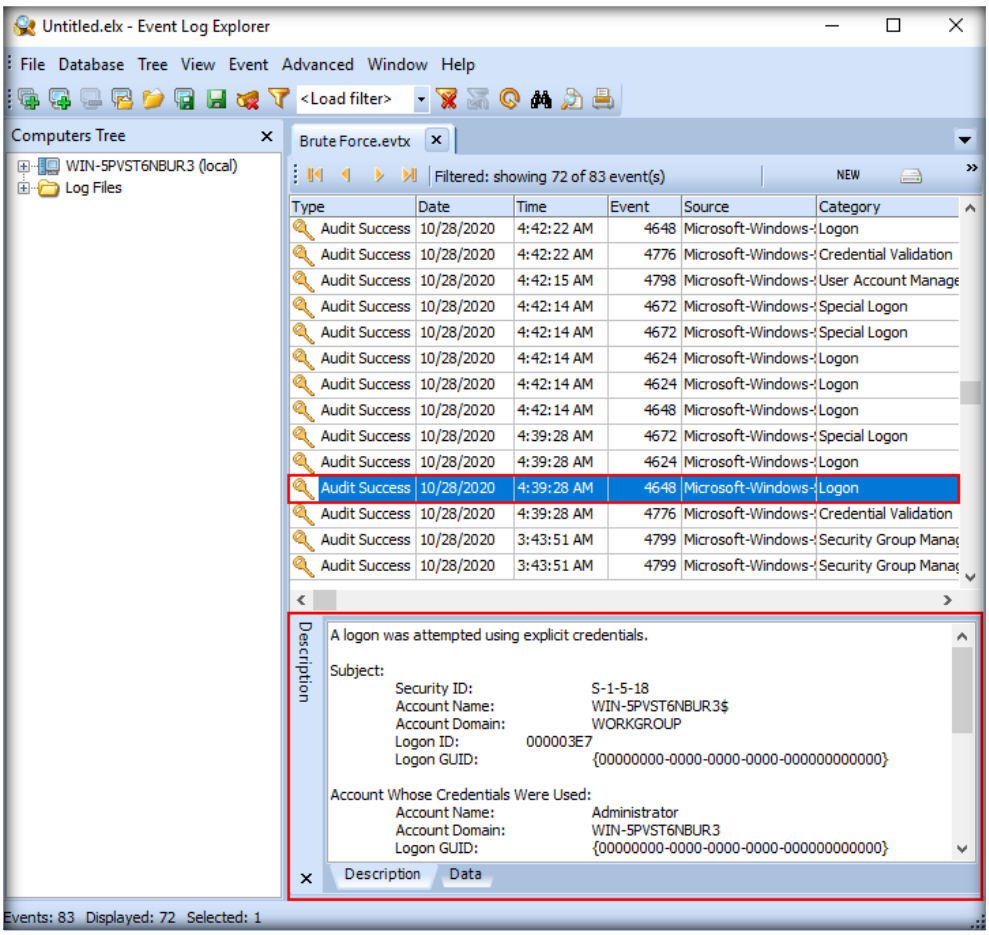


FIGURE 6.17: A logon attempt using explicit credentials detected

26. Now, select the **Audit Success** event that is above the currently selected event and holds the previously mentioned timestamp. Its **Description** pane reveals that an account was successfully logged onto. Further scrolling down in the **Description** pane reveals the account name (here, it is **Administrator**) on which the successful login occurred, as indicated through the screenshots below:

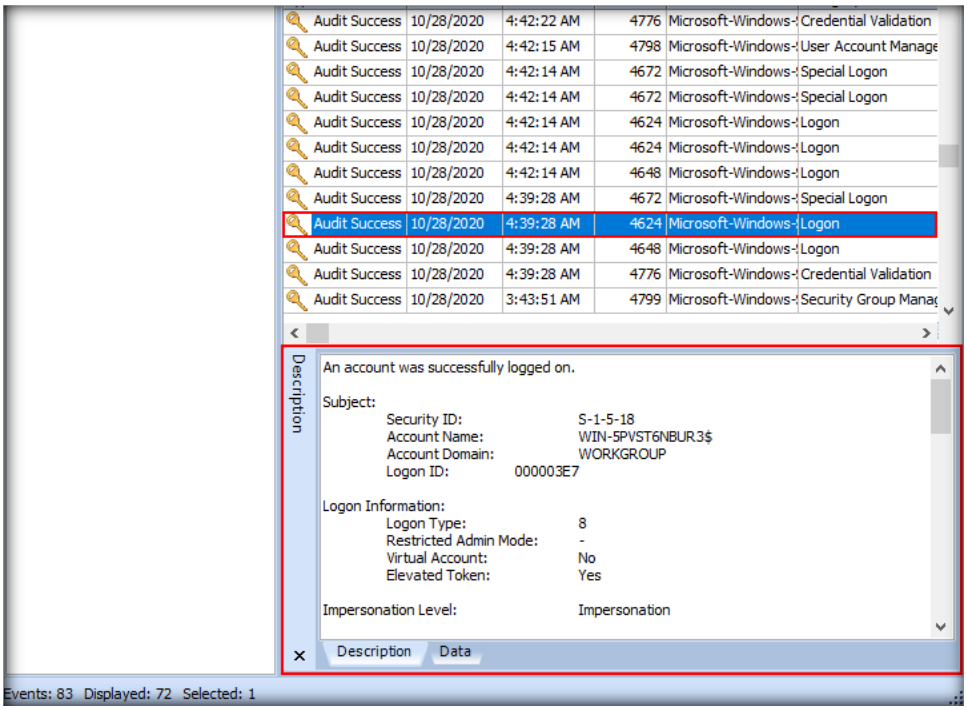


FIGURE 6.18: Successful Logon to an account in an Audit Success event

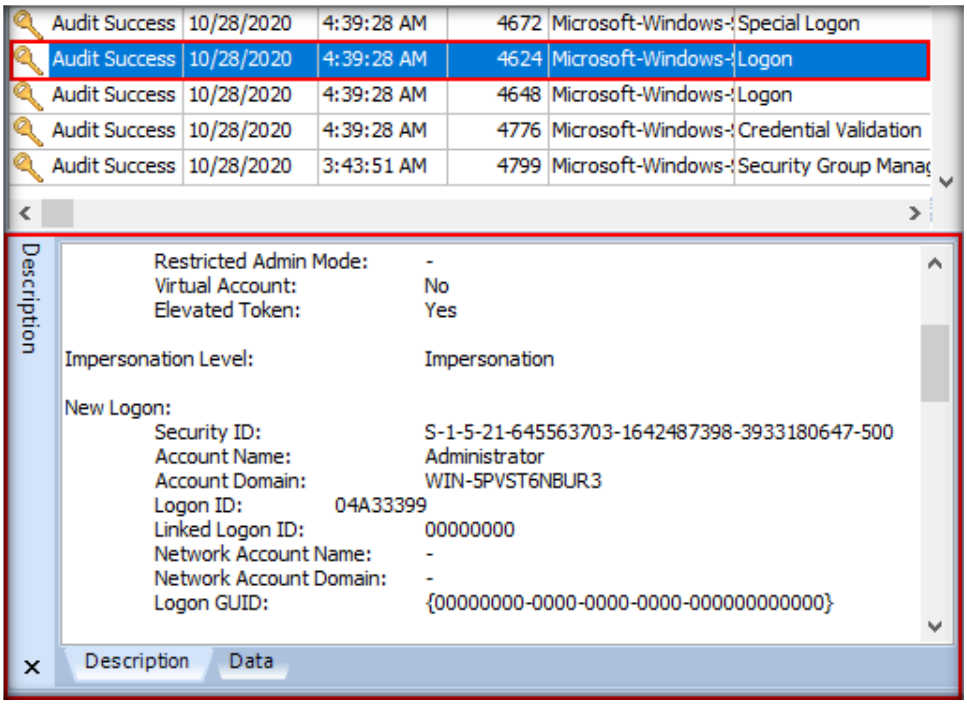


FIGURE 6.19: Account name (Administrator) for which the logon occurred

27. Now, select the **Audit Success** event that is above the currently selected event and holds the previously mentioned timestamp. Its **Category** field mentions **Special Logon**. The **Description** pane for this event, at the very top, reveals that special privileges have been assigned to the new logon, as indicated in the screenshot below. This implies that the suspicious user who had logged in might have attained elevated privileges on the local machine.

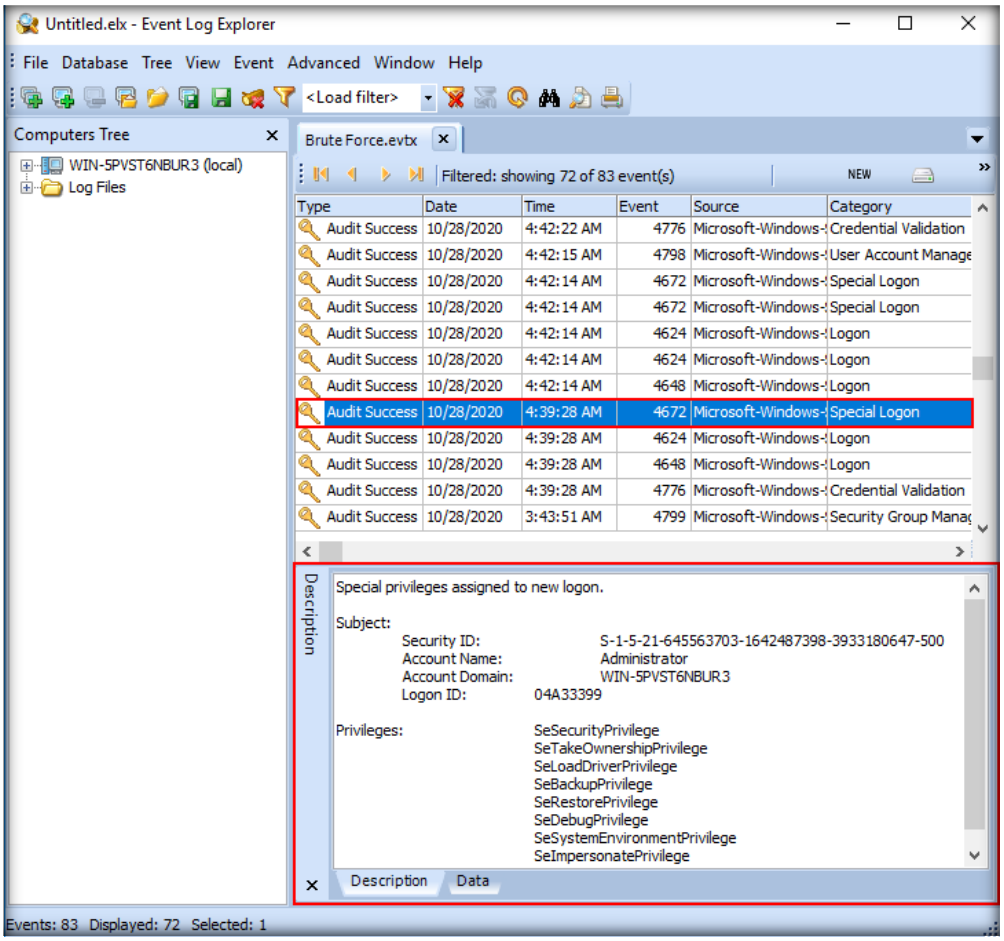


FIGURE 6.20: Audit Success event showing special privileges assigned to a new logon

28. All the above findings reveal that a successful login attempt occurred on the local machine after multiple, repeated failed login attempts. Both the failed and successful login events occurred in quick succession as they both reflect the same timestamps. This is strongly indicative of a brute-force attack.

29. We will now **save** the **Audit Failure** and **Audit Success** event logs that we just examined.

TASK 5

Saving the Event Logs

30. Before saving the event logs, click the **Filter** icon at the top. In the **Filter** window that appears, check only the options **Audit Success** and **Audit Failure** under the **Event types** section while leaving all other options unchecked. Then, click **OK**, as indicated in the screenshot.

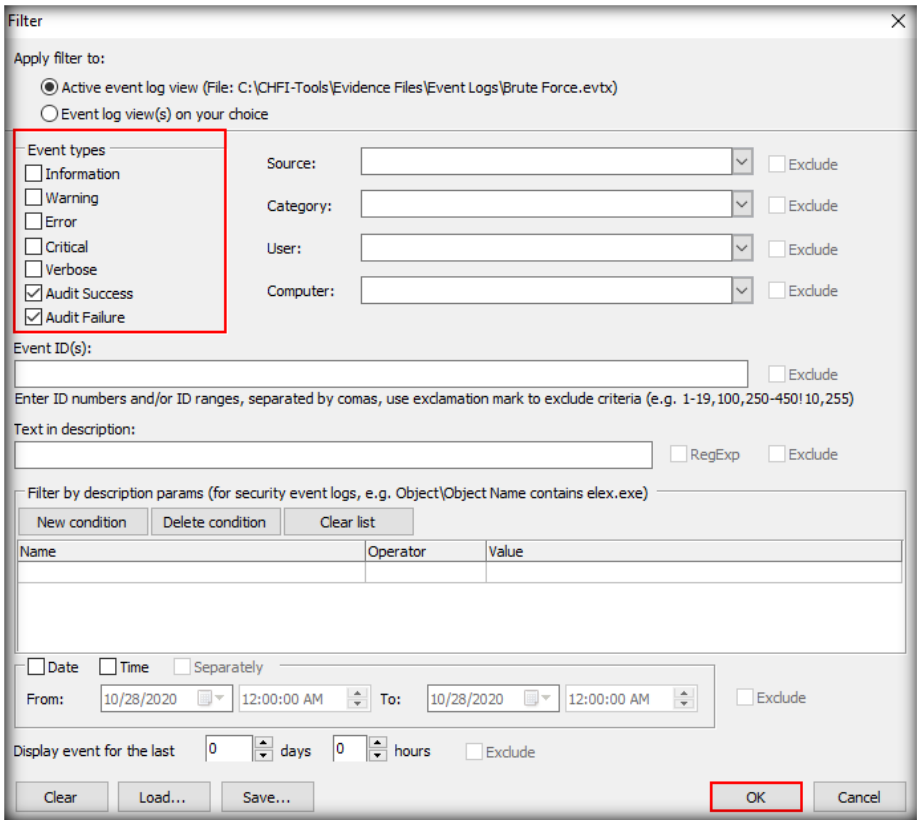


FIGURE 6.21: Applying filters to choose only Audit Failure and Audit Success events for saving

31. You will now see both the **Audit Success** and **Audit Failure** events listed in the right pane of the window, as indicated in the screenshot below:

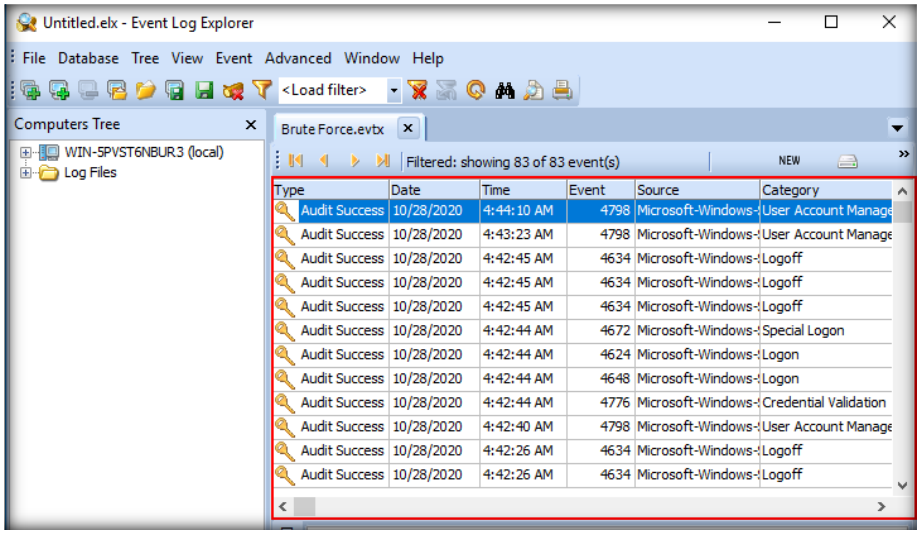


FIGURE 6.22: Audit Success and Audit Failure events listed in the right pane

32. Now, to save the event log file, go to **File →Save Log As...→Save Event Log (Backup)...**

📖 Event Log Explorer allows bookmarking. Bookmarking is a handy method to navigate between events in the logs view.

📖 Event Log Explorer's log view control toolbar displays a log view status message (e.g. Loading, Filtering, Showing events), event list navigator buttons (First, Previous, Next, and Last), and different status indicators.

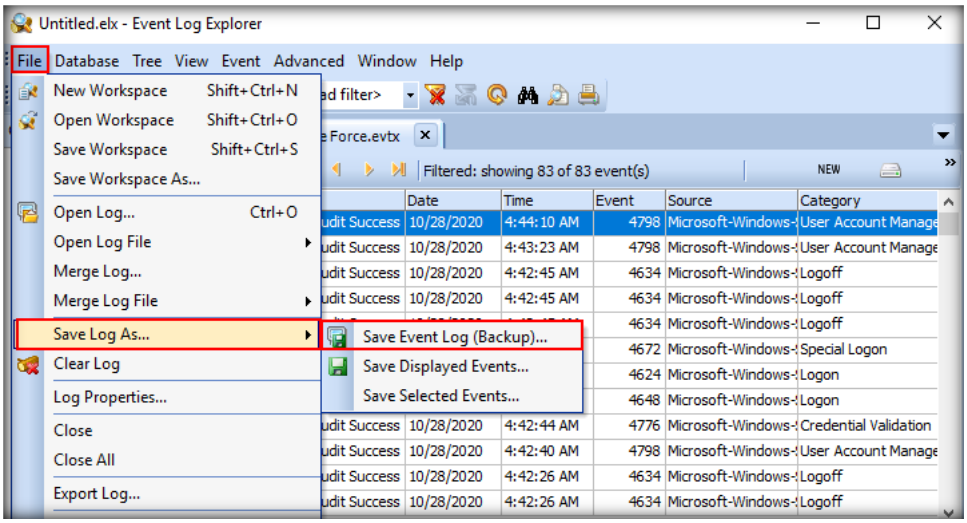


FIGURE 6.23: Saving Event logs

33. A **Save** window will appear. Navigate to the location where you wish to save the log file. Here, we are saving the file on **Desktop**. Name the log file (here, we are naming it as **Security Event Logs**) and click **Save**.

📖 Event Log Explorer allows sorting a list by a specified column. To sort a list, click the column header. Click a second time to reverse the sort order.

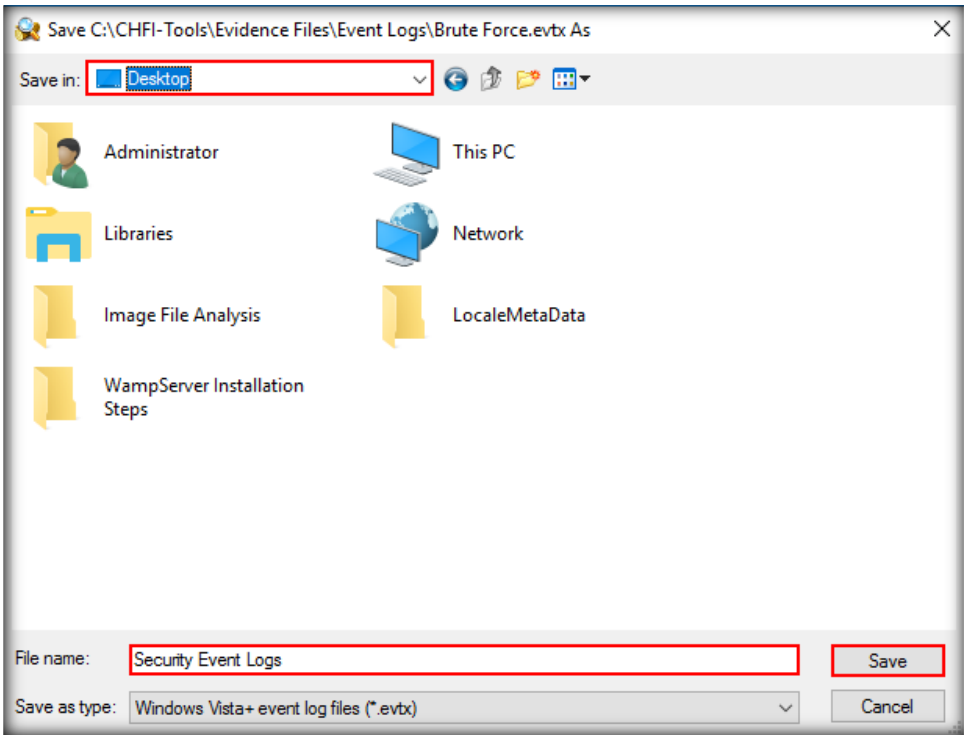


FIGURE 6.24: Saving event logs to Desktop

34. To view the saved security event log file, you can directly open it in Event Viewer from the location where you have saved it. Alternatively, you can access it from the Event Explorer tool window by going to **File → Open Log File → Standard (legacy)**.

35. Similarly, you can also examine application event logs and system event logs by using log files that have recorded those events. In this lab, we have used only the **Brute Force.evtx** file, which shows security events.
36. In this manner, you can examine **Windows** events through **Event Log Explorer**.

Lab Analysis

Analyze the security, application, system, and other logs of the computer, and document the results related to the lab exercise. Give your opinion on the target computer's security situation and exposure.

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

Lab

7

Performing Digital Forensic Investigation on a Computer

Digital forensic investigation involves uncovering and extracting forensic evidence from a target system that could be useful for an investigation.

ICON KEY

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

Lab Scenario

When a cyber-security breach occurs in an organization, network security administrators should be able to aptly respond to the situation and take the necessary steps to avoid further damage. In case there are no professional incident response personnel to respond to the situation, the company would hire an investigator for incident response. To be an expert computer forensic investigator, you must have thorough knowledge on incident response methodologies. This lab discusses how an investigator can respond to incidents by using the Helix tool.

Lab Objectives

The objective of this lab is to help you learn how to investigate cyber-crimes using the Helix tool.

Lab Environment

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics

To carry out the lab you need:

- A computer running **Windows Server 2016** virtual machine
- A computer running **Windows 10** virtual machine
- Administrative privileges to execute the commands
- A web browser with internet access
- **Helix** installer located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\Helix**

Note: You can download the latest version of the Helix tool from the link http://www.e-fense.com/h3-enterprise.php

If you are using the latest version of software for this lab, then the steps and screenshots demonstrated in the lab might differ.

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 30 minutes

Overview of the Lab

Based on different tasks, this lab helps you understand how to perform forensic investigation on a Windows system with the help of the Helix tool.

Note: This lab is based on the free version of Helix.

Lab Tasks

1. Login to the **Windows Server 2016** virtual machine.

   Note: Before beginning the lab, we will create a folder named **Helix Forensic Images** in **Documents**, which we will need to save the forensic images of a logical drive that we are going to create in the lab.

2. Navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\Helix** and double-click **helix.exe**.

3. The application opens and displays a warning message. Read it thoroughly and select your preferred language from the **Choose Your Language** drop-down menu.

4. Click **Accept** to continue.

FIGURE 7.1: Warning message in Helix

5. After clicking the **Accept** button, Helix GUI appears, as shown in the following screenshot:

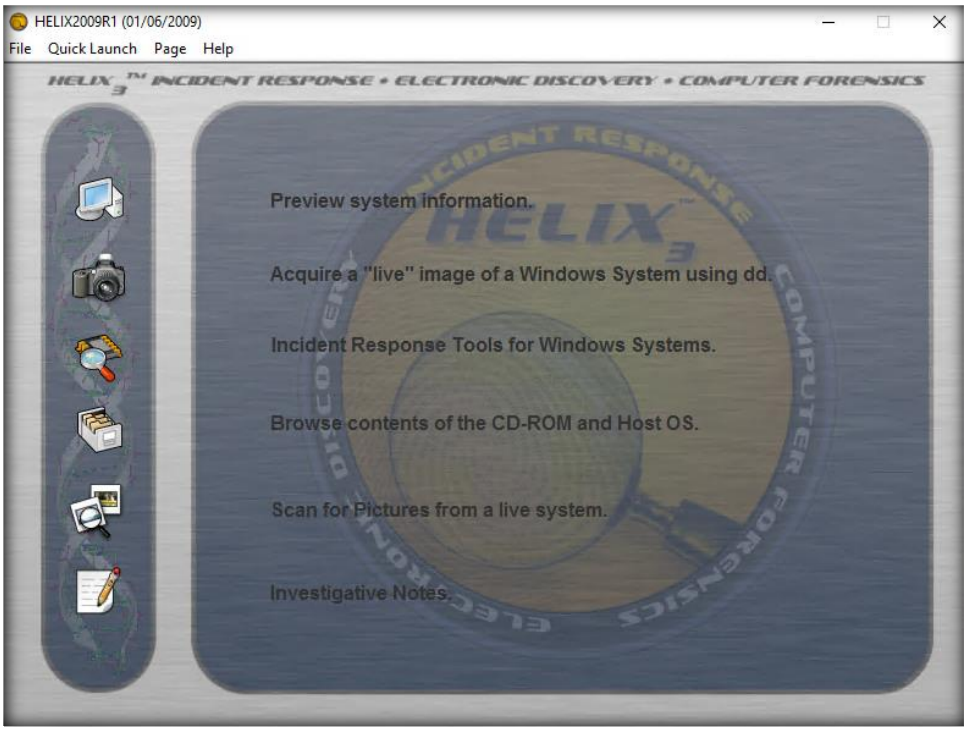


FIGURE 7.2: Main interface of Helix

Helix is a fast and powerful live CD originally based on Knoppix, which provides all the necessary tools for live forensics, incident response, and e-discovery.

TASK 1

Viewing System Information

6. Click the **System Information** icon on the left side of the window to view complete system information.

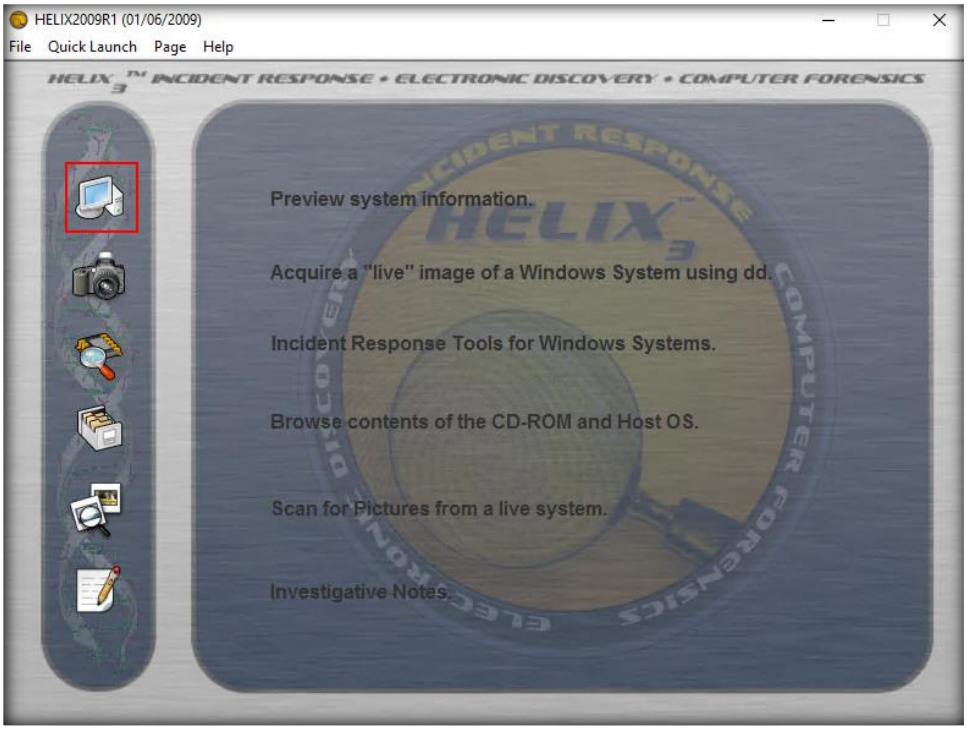


FIGURE 7.3: System information icon in Helix

Helix is a customized version of Ubuntu Linux, allowing you to boot into a Linux environment containing customized Linux kernels, hardware detection, and a large number of applications designed for incident response and forensics.

7. The first page of system information displays the OS information, owner information, network information, drives on the system, and the file types of the drives.

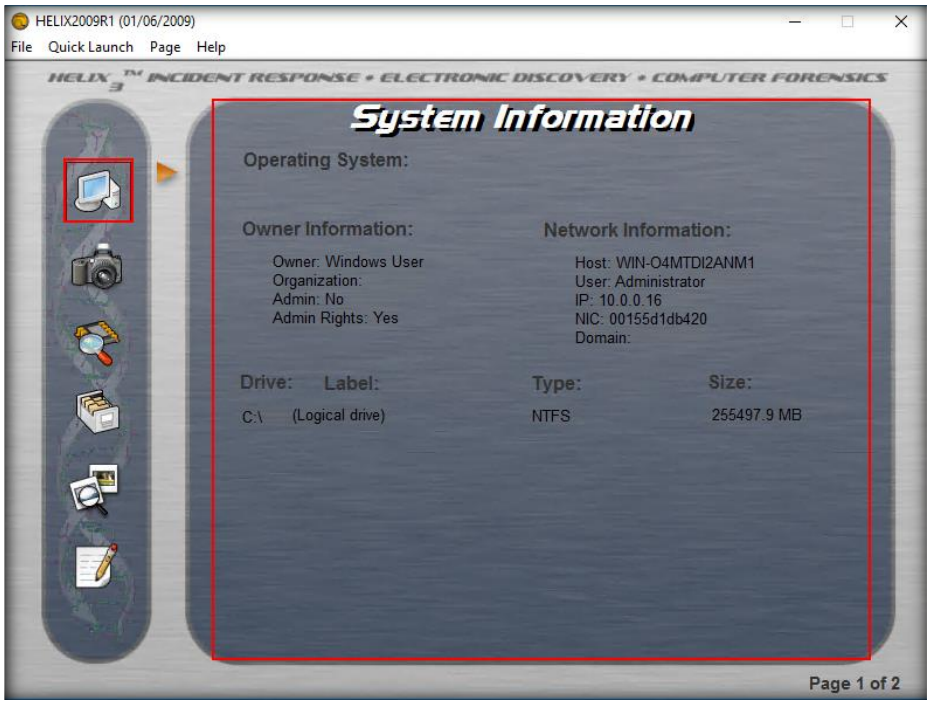


FIGURE 7.4: System Information details in Helix

Helix does not touch the host computer in any way and is forensically safe. It will not auto-mount swap space or attached devices.

Helix mainly provides incident response and forensics tools and is intended for users with solid knowledge of these techniques.

8. Click the  button to go to the second page of **System Information**. This page displays the processes running on the system.

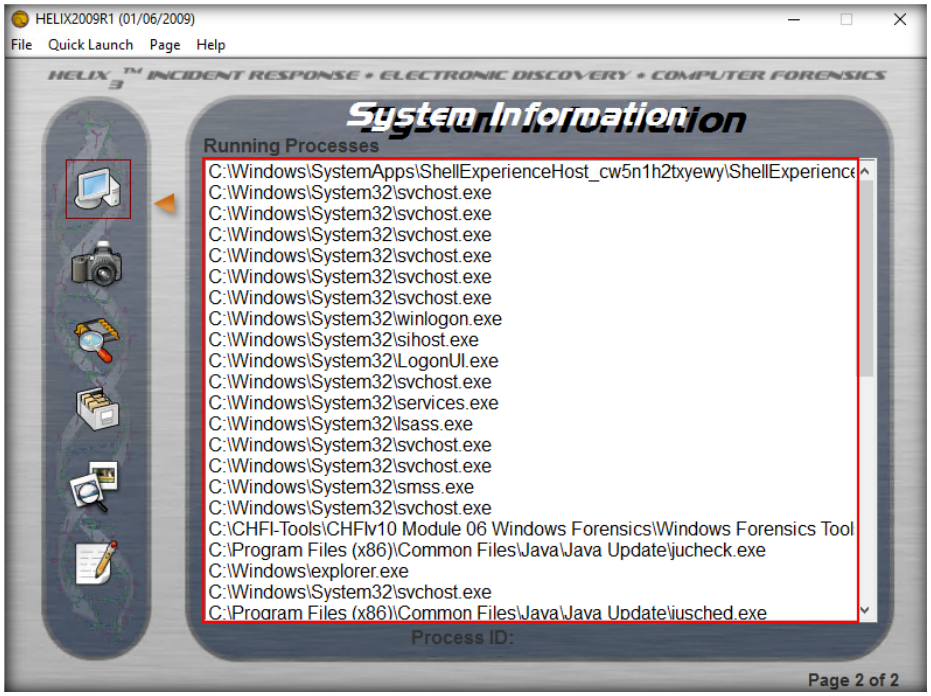


FIGURE 7.5: Helix System Information showing running processes

TASK 2

Perform Data Acquisition

9. Click the **Acquisition** icon to acquire physical memory or disk drives.

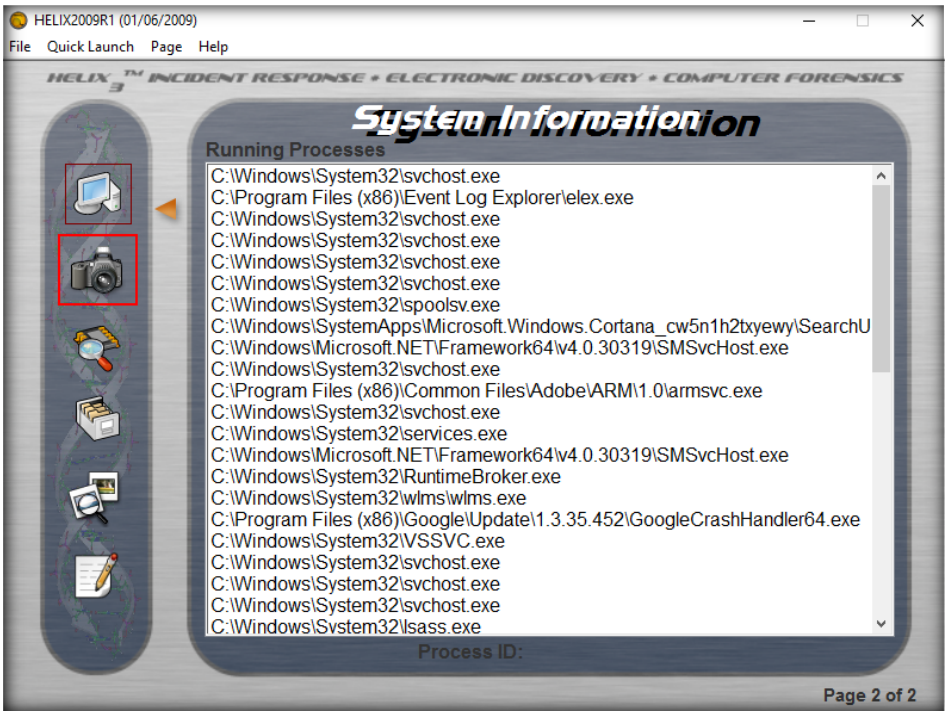


FIGURE 7.6: Helix acquisition

10. Here, we will be demonstrating the steps to create/acquire the forensic image of a logical drive (here, the **C:** Drive). Upon clicking the **Acquisition** icon, navigate to its second page by clicking on the arrow icon, as highlighted in the following screenshot:

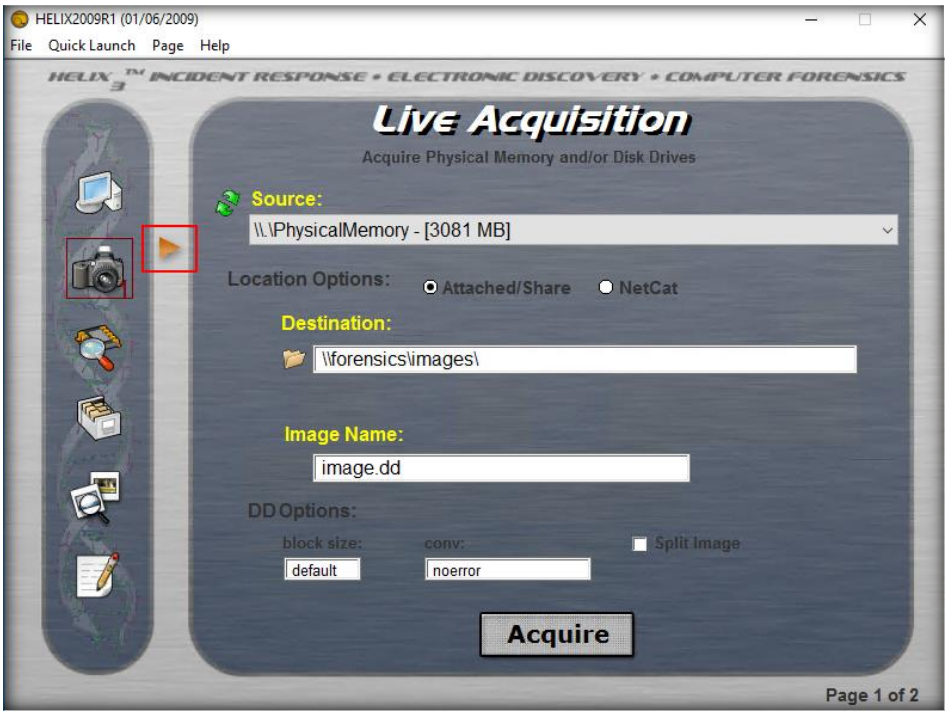


FIGURE 7.7: Step 1 of acquisition in Helix

Security firm e-fense released a new version of their popular Linux-based forensics live CD, Helix. This new version is Ubuntu-based, which seems to be a popular choice among this genre of tools.

11. The tool will now take you to the second page of **Live Acquisition**. Click on **Imager**, as shown in the following screenshot:

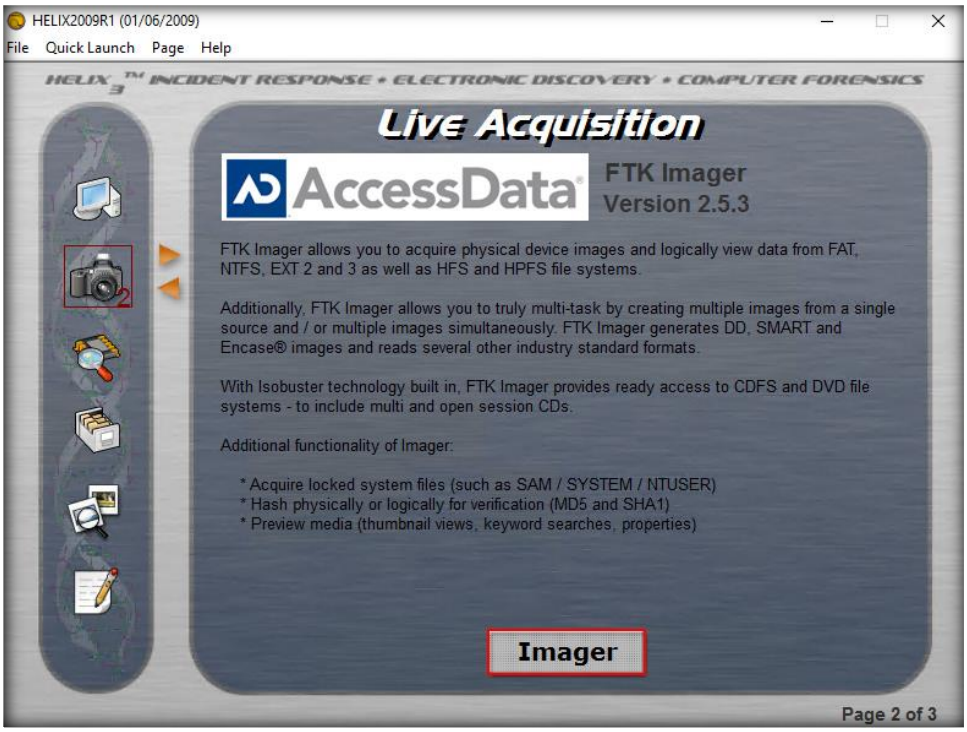


FIGURE 7.8: Step 2 of acquisition in Helix

Helix is a customized distribution of Ubuntu Linux. Helix is more than simply a bootable live CD. You can also boot into a customized Linux environment that includes customized Linux kernels, excellent hardware detection, and many applications dedicated to incident response and forensics.

TASK 3

Create Disk Image

12. Helix will now launch the Access Data FTK Imager application. Go to **File→Create Disk Image...**.

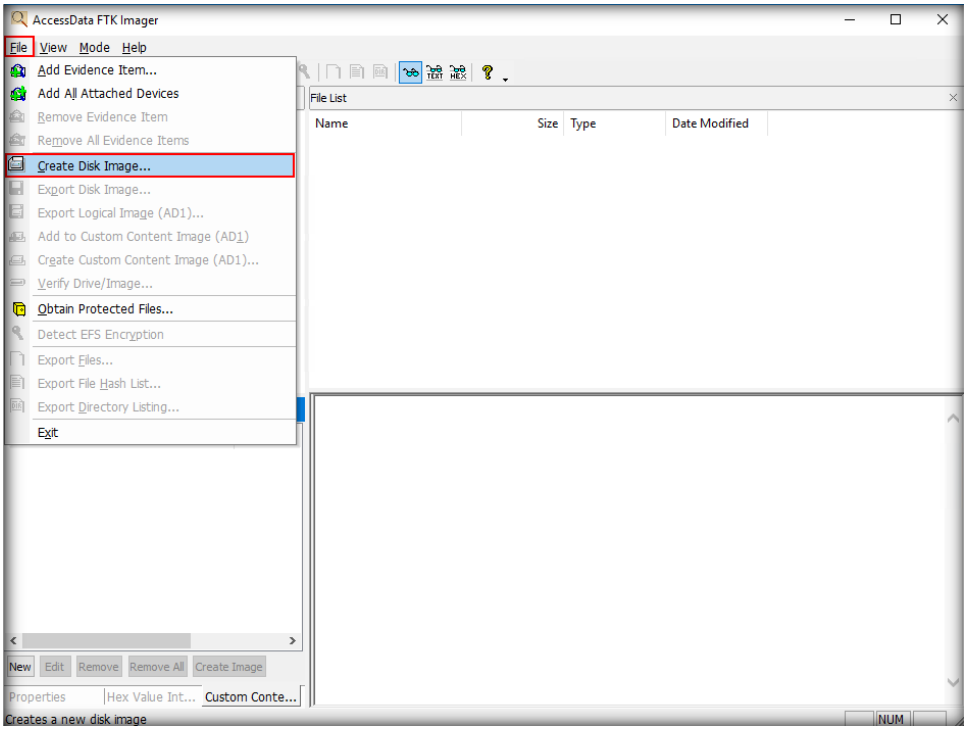


FIGURE 7.9: Helix Acquisition AccessData FTK Imager

Preview files and folders on local hard drives, network drives, floppy diskettes, zip disks, CDs, and DVDs.

13. The **Select Source** window appears. Select **Logical Drive** as the source evidence type and then click **Next** to proceed to creating an image.

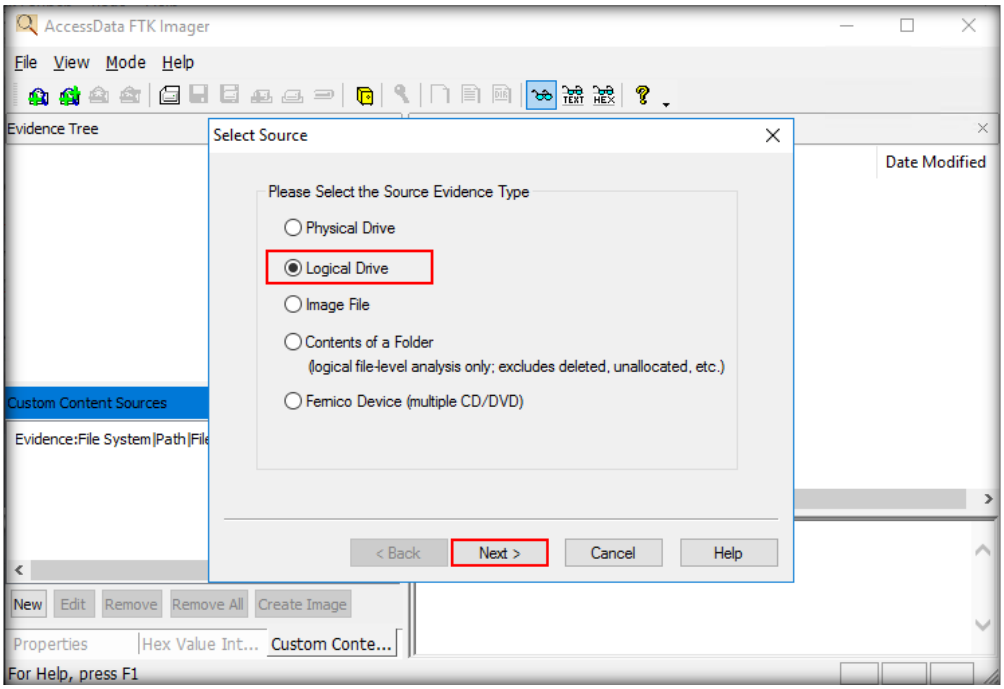


FIGURE 7.10: Select a source in Helix Acquisition AccessData FTK Imager

Preview the contents of forensic images stored on the local machine or on a network drive.

14. You will now see the **Select Drive** window, where you need to select **C:\ - [NTFS]** in the **Source Drive Selection** section. Click **Finish** to proceed.

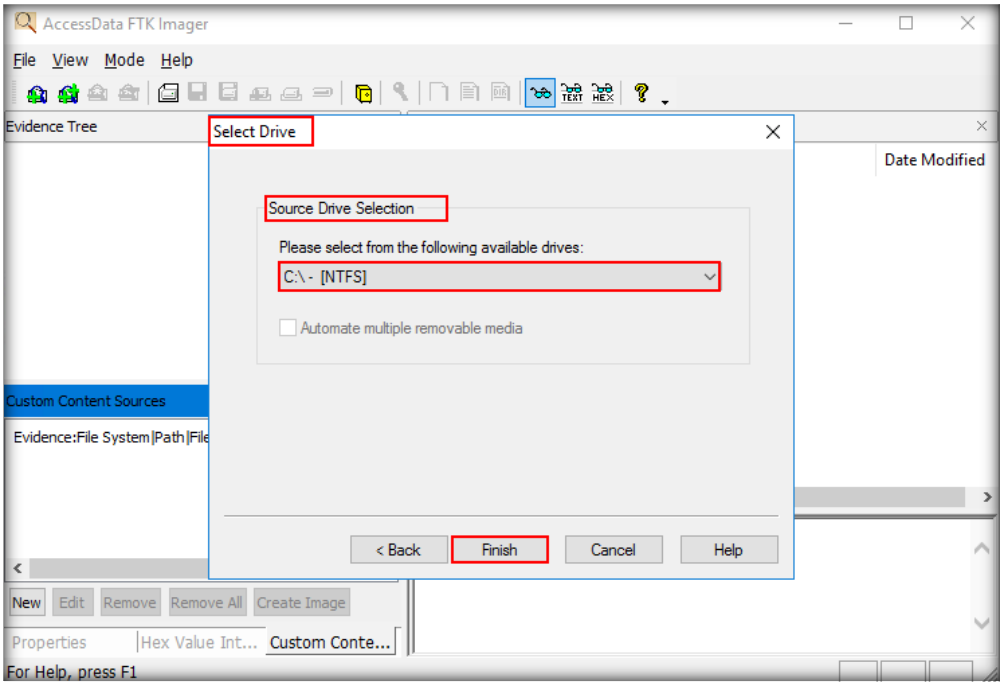


FIGURE 7.11: Source Drive Selection section in Helix Acquisition AccessData FTK Imager

Mount an image for a read-only view that leverages Windows Explorer to show the content of the image exactly as the user saw it on the original drive.

15. The **Create Image** window now appears. Click **Add**.

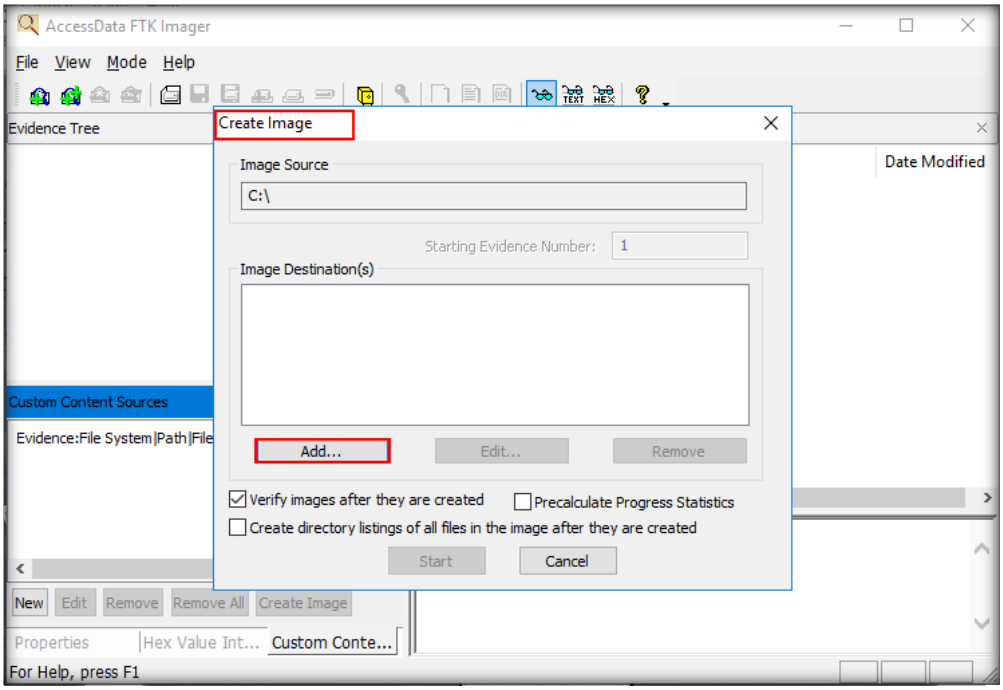


FIGURE 7.12: Specify a folder path to create an image in Helix Acquisition AccessData FTK Imager

16. Upon clicking **Add**, the **Select Image Type window** will open. In the **Please Select the Destination Image Type** section, select **Raw (dd)**, and click **Next**.

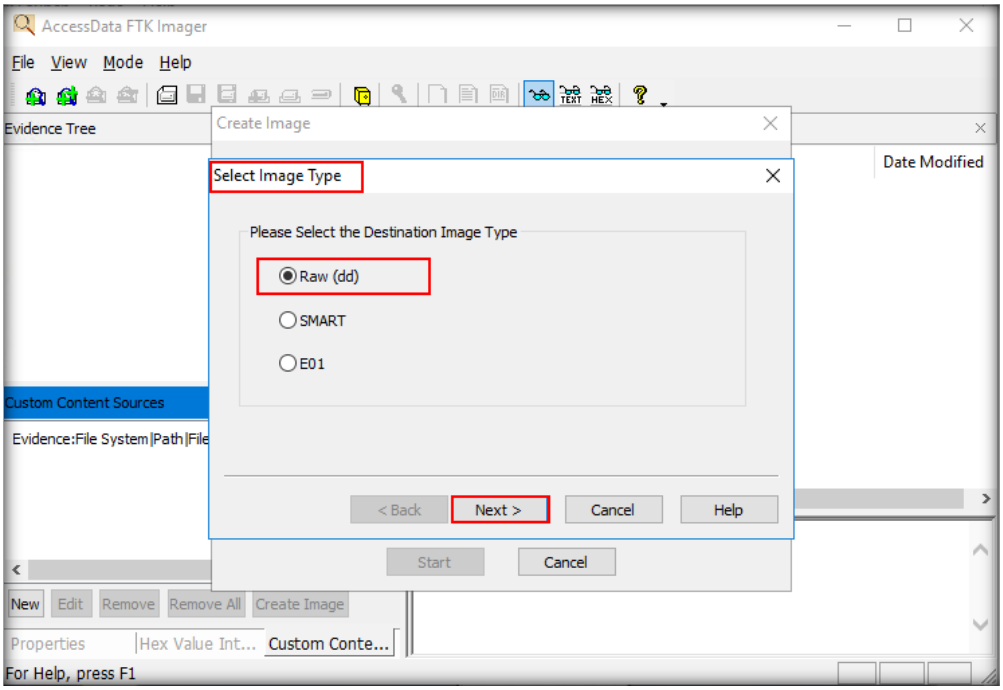


FIGURE 7.13: Select Image Type window

View and recover files that have been deleted from the Recycle Bin but have not yet been overwritten on the drive.

17. The **Evidence Item Information** window will now appear. In this window, fill out the case details and click **Next**. You may also click **Next** without filling out the case details.

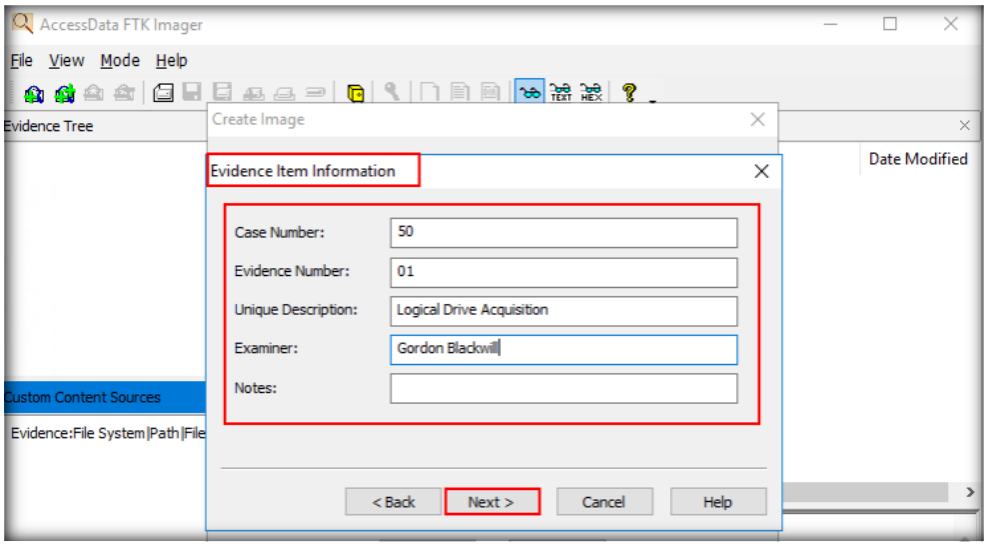


FIGURE 7.14: Evidence Item Information window in Helix Acquisition AccessData FTK Imager

Create hashes of files using either of the two hash functions available in FTK Imager: Message Digest 5 (MD5) and Secure Hash Algorithm (SHA-1).

18. The **Select Image Destination** window now appears. Click **Browse** to provide the path in the **Image Destination Folder** field (here, we will provide the **Helix Forensic Images** folder that we created in **Documents** as the image destination). In the **Image Filename (Excluding Extension)** field, provide the name for the logical drive image that you will create (here, **C Drive Image**). In the **Image Fragment Size (MB)** field, leave the value set to default (**1500**) to allow the tool to create the image in several fragments for reasons of disk space sufficiency. Click **Finish**.

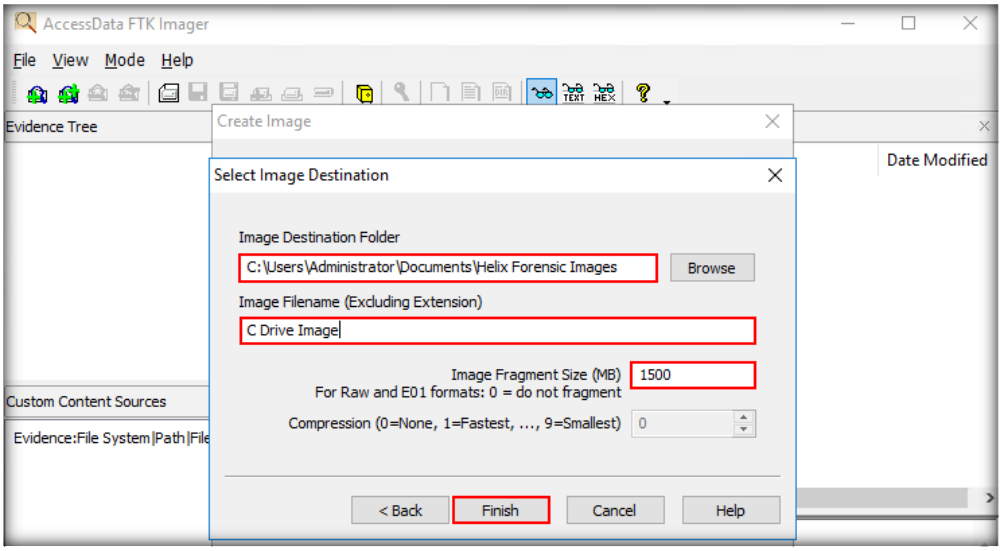


FIGURE 7.15: Select Image Destination window in Helix Acquisition AccessData FTK Imager

Generate hash reports for regular files and disk images (including files inside disk images) that you can later use as a benchmark to prove the integrity of your case evidence.

19. You will now be taken back to the **Create Image** window. Click the **Start** button to initiate the creation of the logical drive image. You may leave the other options above the **Start** button set to default.

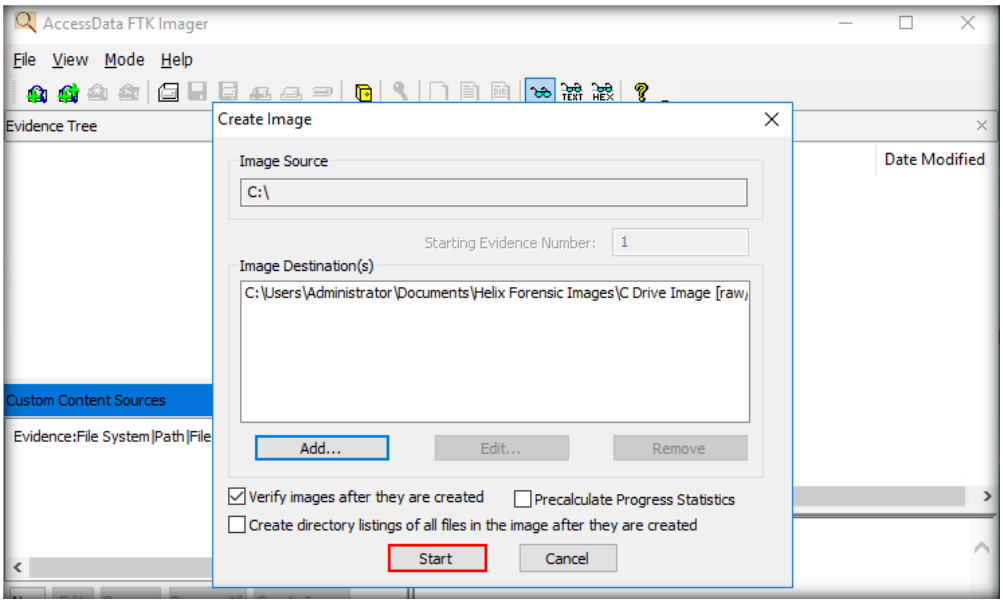


FIGURE 7.16: Starting image creation in Helix Acquisition AccessData FTK Imager

Helix is also unique in that it includes a Win32-based autorun toolset, providing users quick access to common forensic tools not found in Linux.

20. The tool will now start creating the image. When the imaging is complete, the application will save the logical drive image fragments to the destination folder, i.e., **Helix Forensic Images** in the **Documents** directory. You can monitor the progress of the image creation in the **Progress** section of the **Creating Image...** window, as shown in the following screenshot:

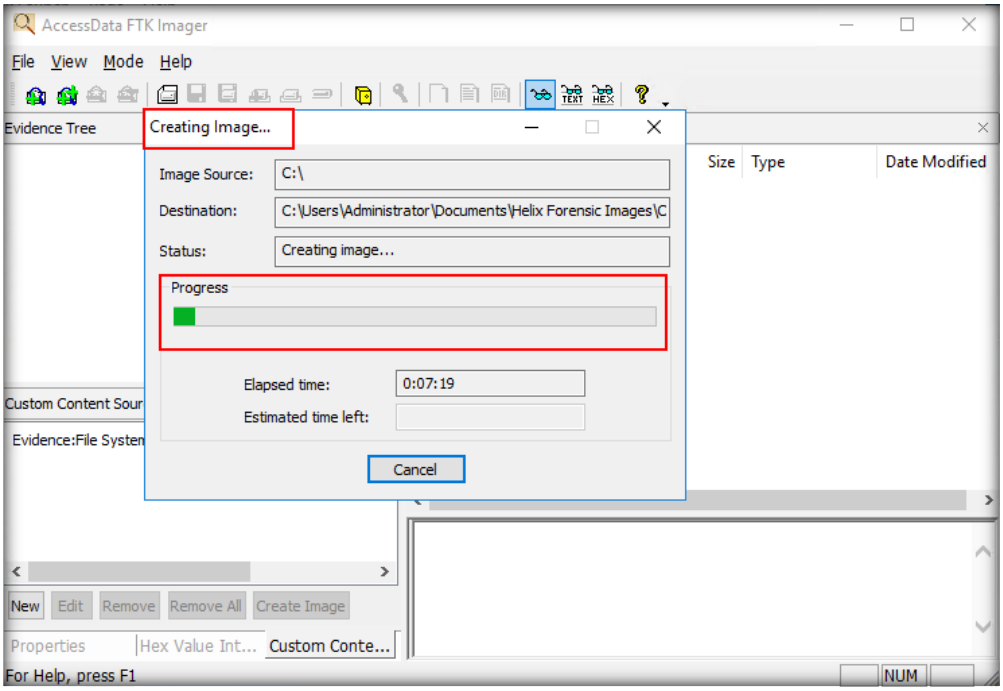


FIGURE 7.17: Image creation in Helix Acquisition AccessData FTK Imager

Helix is a customized distribution of the Knoppix Live Linux CD. Helix is more than simply a bootable live CD.

21. The application will take a long time to complete the imaging process. For the purpose of this task, as we are focusing on demonstrating the steps to create a logical drive image, we will abort the imaging process by clicking **Cancel**.

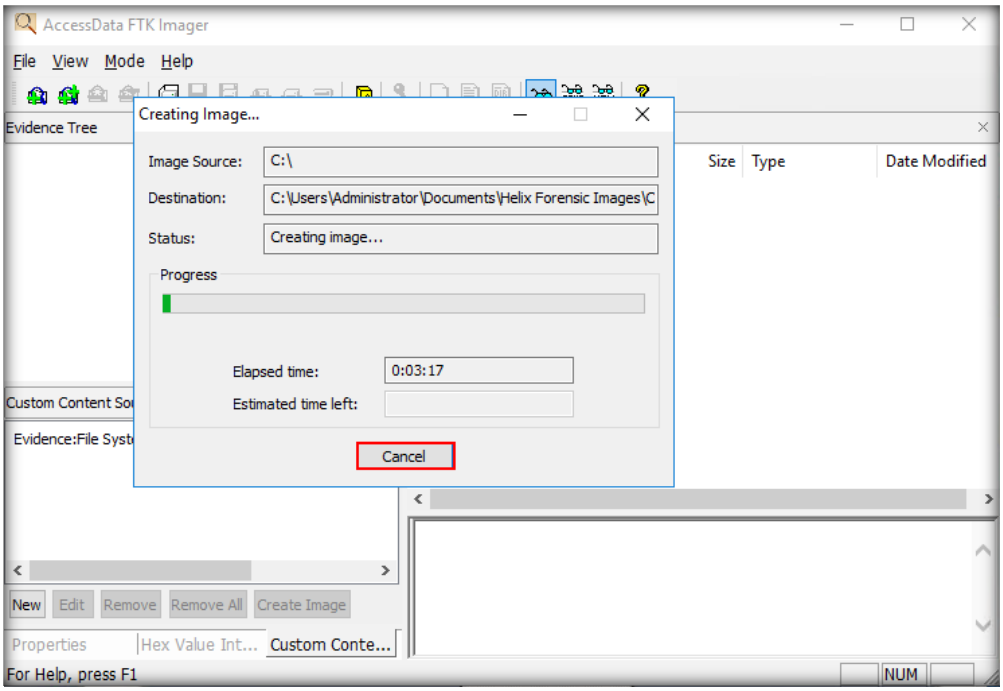


FIGURE 7.18: Canceling of the imaging process

22. Upon clicking **Cancel**, an **FTK Imager** pop-up prompts you to confirm the canceling of the imaging process. Click **Yes**.

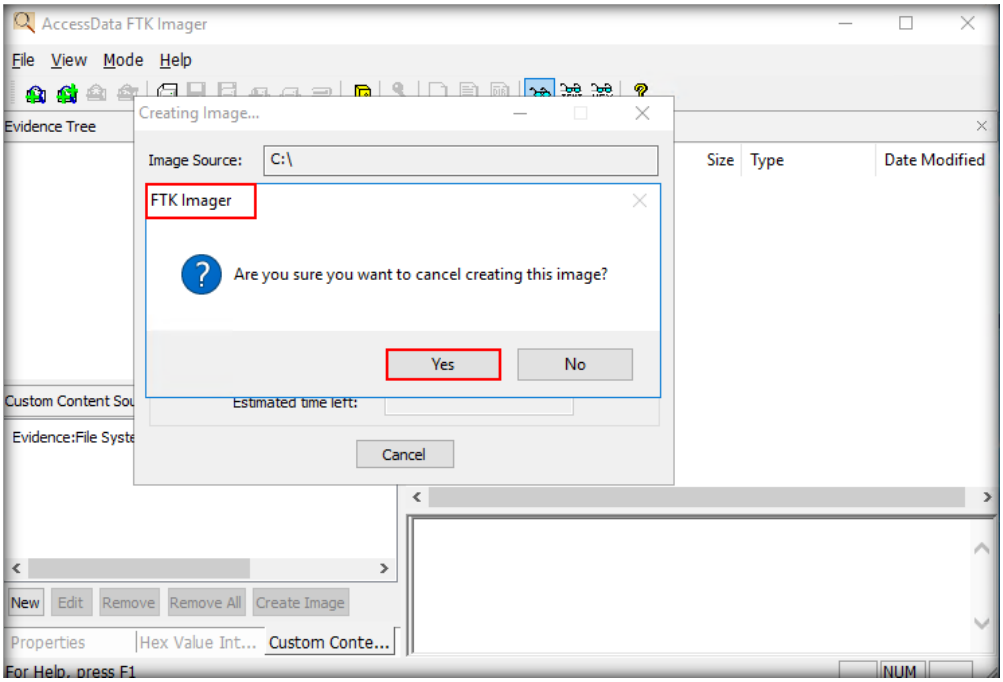


FIGURE 7.19: FTK Imager pop-up

23. The application will now abort the imaging process. Close the **Creating Image…** window by clicking **Close**.

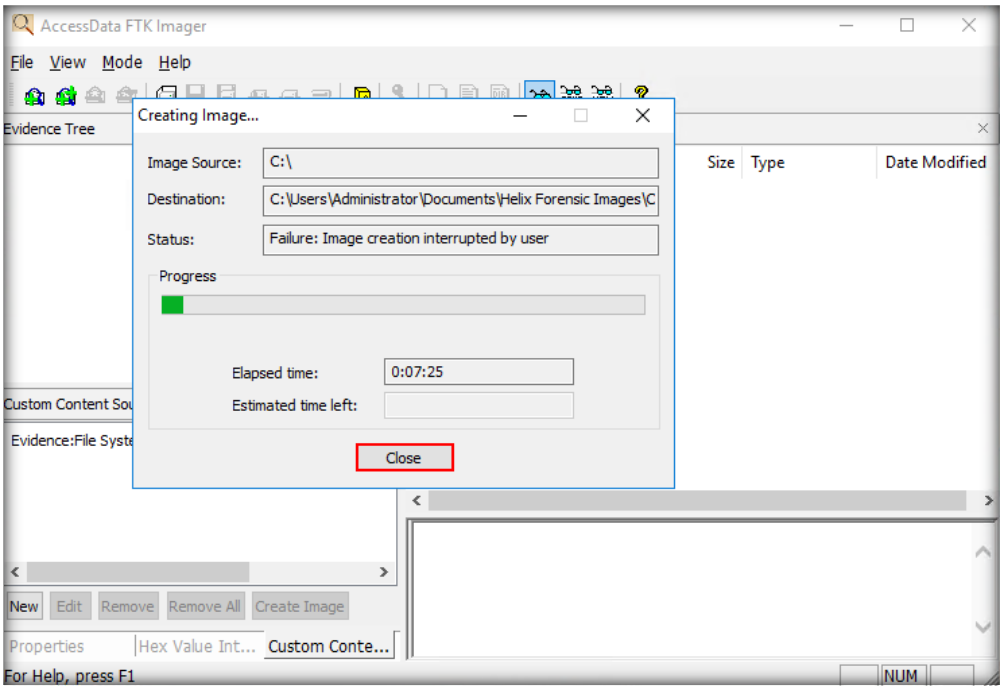


FIGURE 7.20: Close the Creating Image… window

24. Similarly, using the **AccessData FTK Imager**, you can image a **physical drive**, **specific folder**, or **Fernico Device (multiple CDs/DVDs)**. You can also convert an image of one type to another.

Mount an image for a read-only view that leverages Windows Explorer to show the content of the image exactly as the user saw it on the original drive.

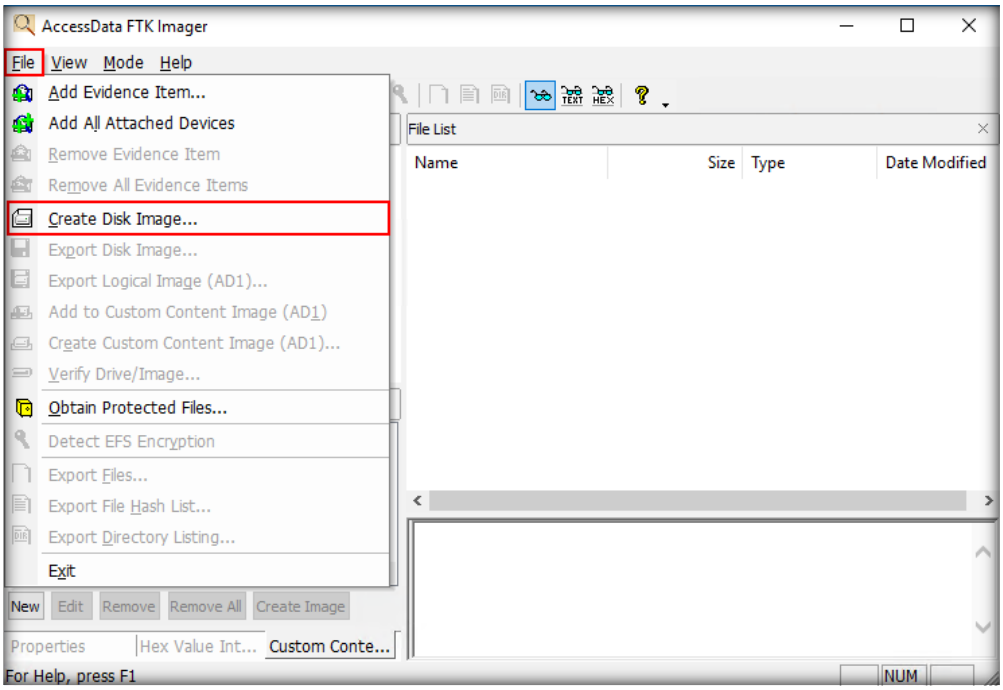


FIGURE 7.21: Create Disk Image… option under the File menu in Helix Acquisition AccessData FTK Imager

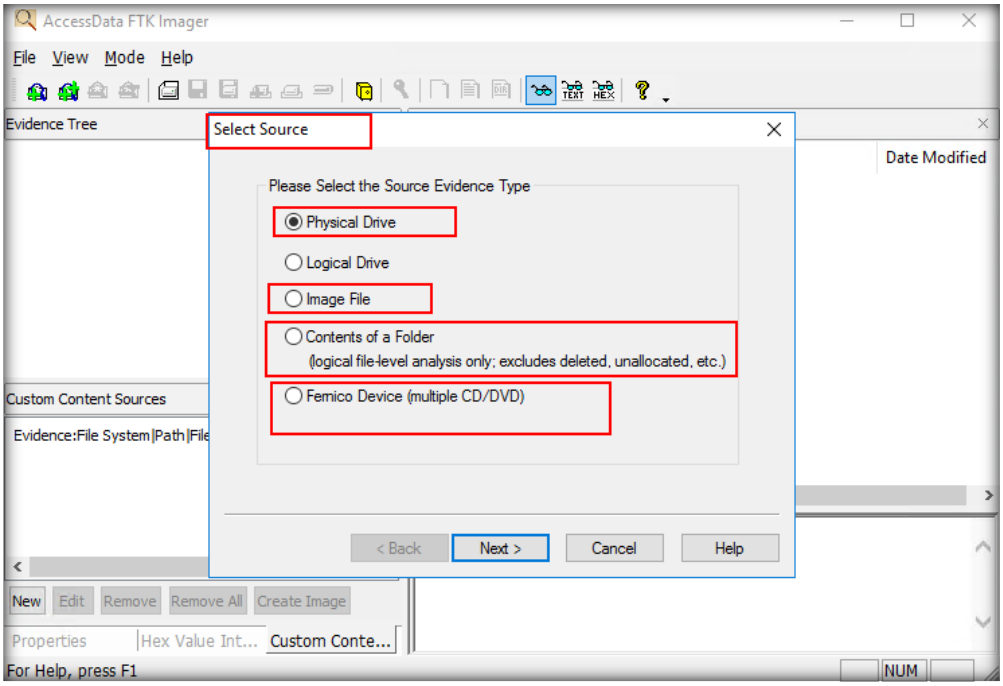


FIGURE 7.22: Select Source window

25. We will now proceed to investigating a forensic image file using **AccessData FTK Imager**. We will investigate the **Windows_Evidence_001.dd** image file located at **C:\CHFI-Tools\Evidence Files\Forensic Images**.

26. Go to **File → Add Evidence Item...** from the menu bar to add the evidence file for investigation.

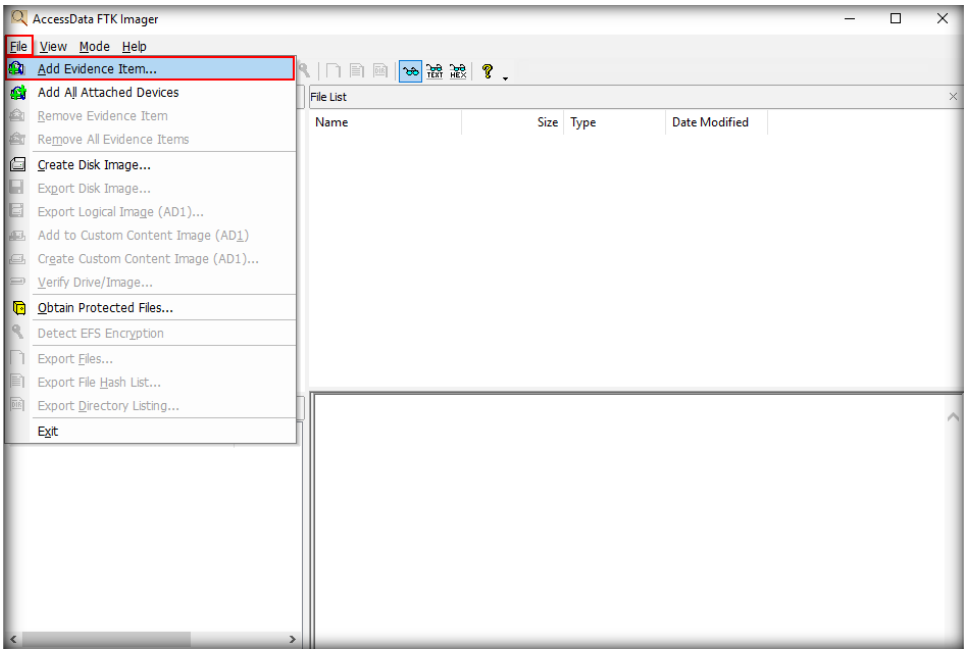


FIGURE 7.23: Add an evidence item in Helix Acquisition AccessData FTK Imager

To prevent accidental or intentional manipulation of the original evidence, FTK Imager makes a bit-for-bit duplicate image of the media. The forensic image is identical in every manner to the original, including file slack and unallocated space or drive free space.

27. A **Select Source** window appears. In this lab, we are going to analyze an image file; so, select the **Image file** radio button and click **Next**.

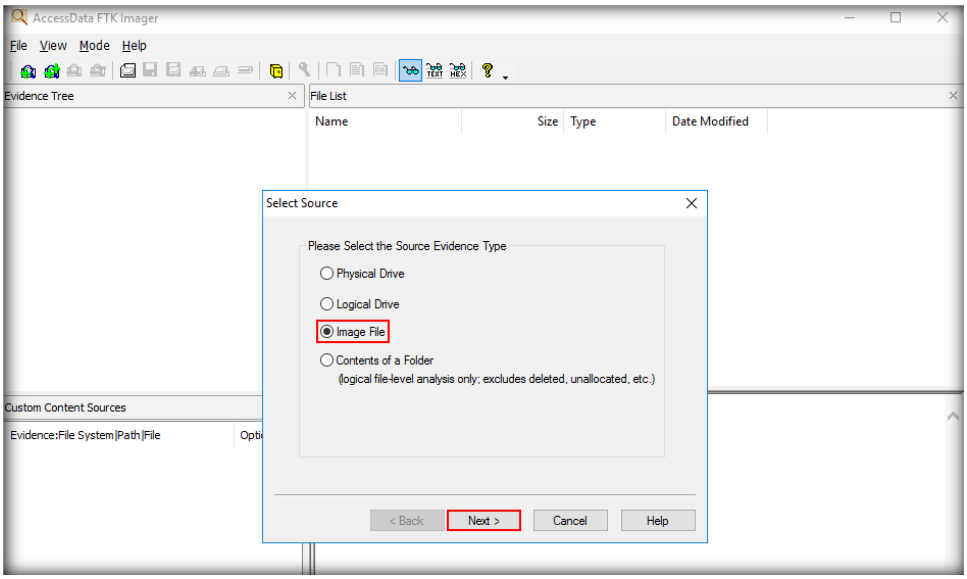


FIGURE 7.24: Select Source window in Helix Acquisition AccessData FTK Imager

FTK Imager can be installed on the computer where it will be used or run from a portable device such as a USB thumb drive connected to a machine in the field; therefore, it need not be installed on a suspect's computer to capture the image.

28. A **Select File** window appears, where you need to specify the path of the image file. In this lab, we will be analyzing **Windows_Evidence_001.dd**; so, specify the source as **C:\CHFI-Tools\Evidence Files\Forensic Images\Windows_Evidence_001.dd** and then click **Finish**.

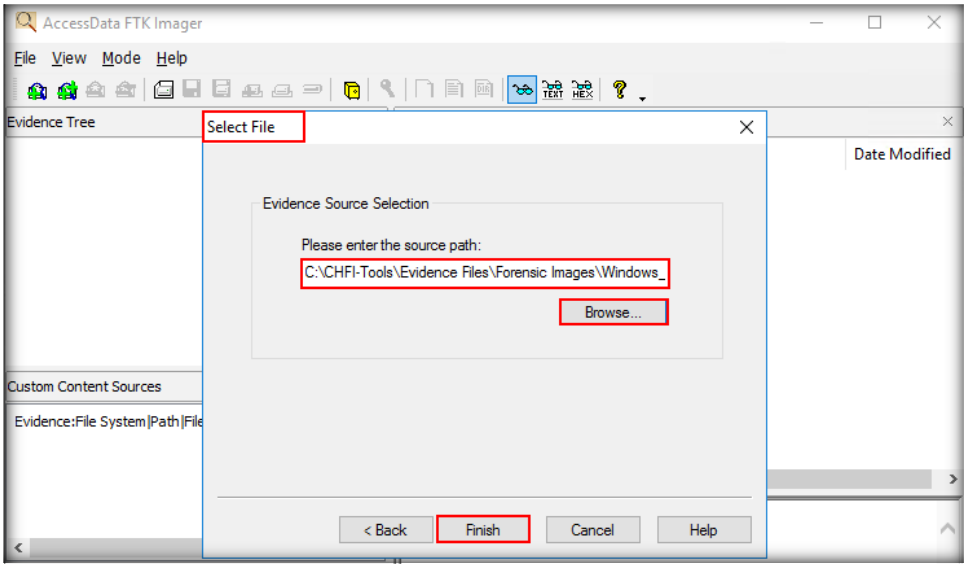


FIGURE 7.25: Select an evidence source in Helix Acquisition AccessData FTK Imager

/CreateDirListing= creates a directory listing file in the folder you run FTK Imager.exe from. /VerifyImage= verifies an image when you specify the image path and filename. /EnableDebugLog= enables logging to the FTKImageDebug.log file created in the folder you run FTK Imager.exe from.

29. The image file is displayed in the **left pane** of the tool window. Now, expand **Windows_Evidence_001.dd → Evidence [NTFS] → [root]** and click on the **Images** folder. The contents of the folder will be displayed in the right pane. Select an image to view it in the lower section of the application window.

Note: The application may not be able to open other file formats

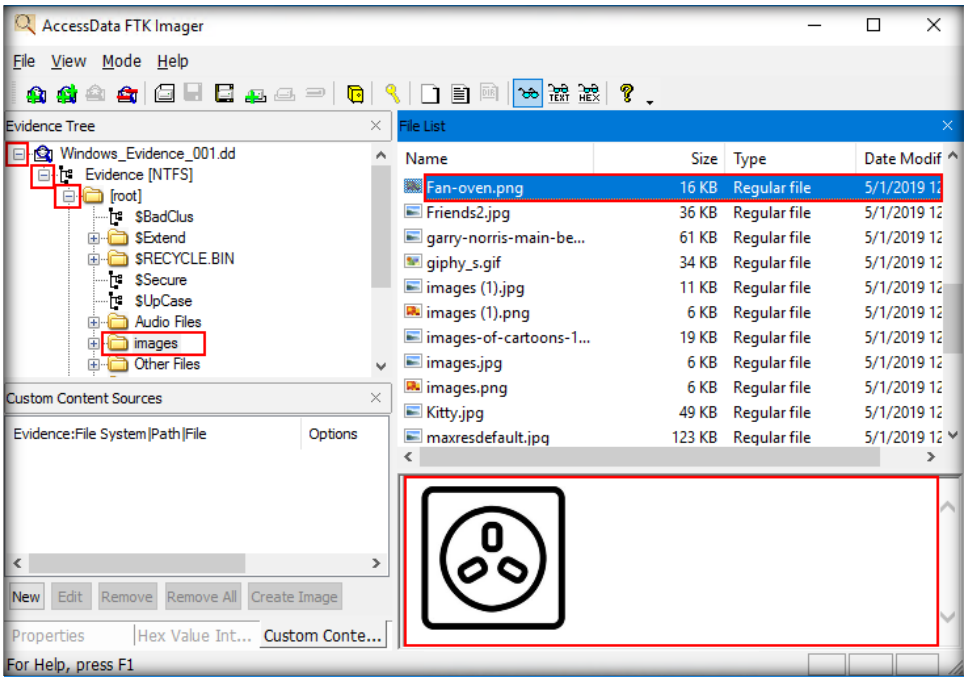


FIGURE 7.26: Evidence content in Helix Acquisition AccessData FTK Imager

30. To view the properties of the image file, click the **Properties** tab in the lower section of the left pane in the tool window. Similarly, you can view the **Hex Value Integer** and **Custom Content Sources** tabs, which are located next to the **Properties tab**, by clicking on them, as shown in the screenshot.

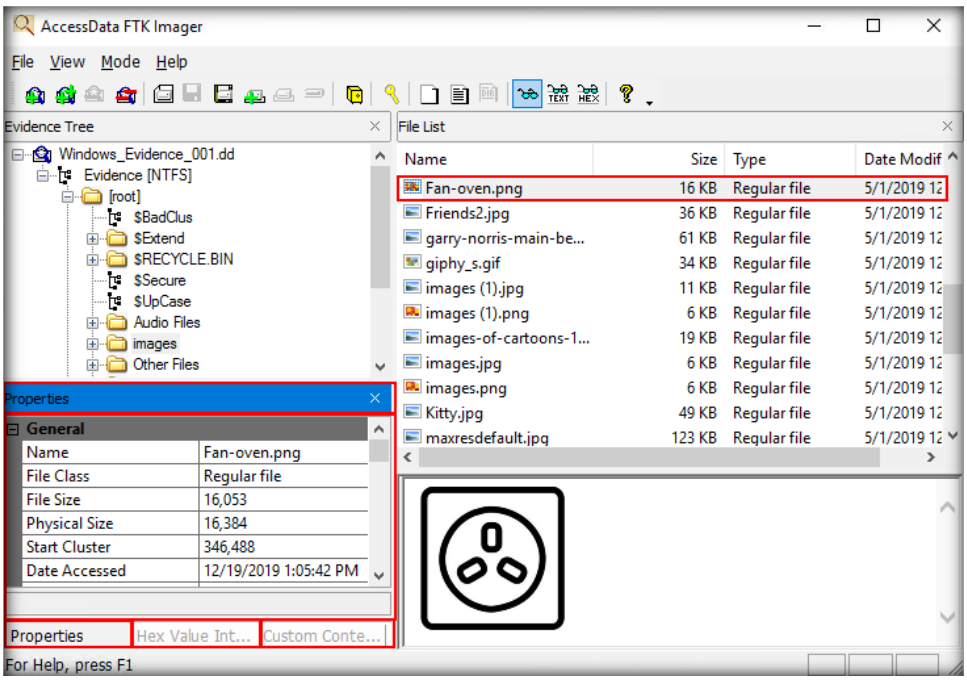


FIGURE 7.27: Properties tab in Helix Acquisition AccessData FTK Imager

If you fail to specify an image when using the /CreateDirListing= or /VerifyImage= options, an error message appears, indicating no image was found.

The **Evidence Tree** pane (upper-left pane) displays added evidence items in a hierarchical tree. At the root of the tree are the selected evidence sources. Listed below each source are the folders and files it contains.

TASK 4

Use Nigilant32 to Preview the Hard Disk

31. In this manner, you can create or view images using the AccessData FTK Imager application. Now, close the application.
32. In the left pane of the **Helix** window, click the **Incident Response** icon.
33. The **Incident Response** section appears. Click the **Agile Risk Management's Nigilant32** icon on **Page 1** of this section.

FIGURE 7.28: Incident Response in Helix

Nigilant32 is an incident response tool designed to capture as much information as possible from a running system with the smallest potential impact.

34. You will be prompted with a **Notice** pop-up. Click **Yes**.

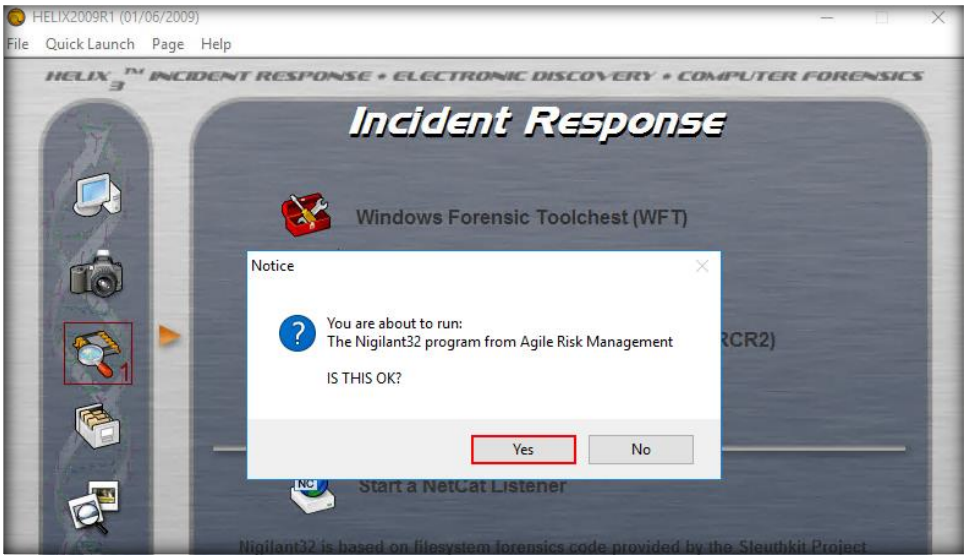


FIGURE 7.29: Nigilant32 notice in Helix Incident Response

Nigilant32 For First Responders: By using the SnapShot feature of Nigilant32, we can review and save a report of the running system that includes processes, services, user accounts, scheduled tasks, and network ports.

35. The **Nigilant 32** pop-up appears. Click on the pop-up.

FIGURE 7.30: Helix Incident Response Nigilant 32 pop-up

Nigilant32 For First Responders: By using the filesystem review feature of Nigilant32, we can explore the filesystem and possibly locate hidden files, folders, or recently deleted content or extract files for offline analysis with limited risk of contamination.

36. The **Nigilant32 - Windows Afterdark Forensic** window appears, as shown in the screenshot:

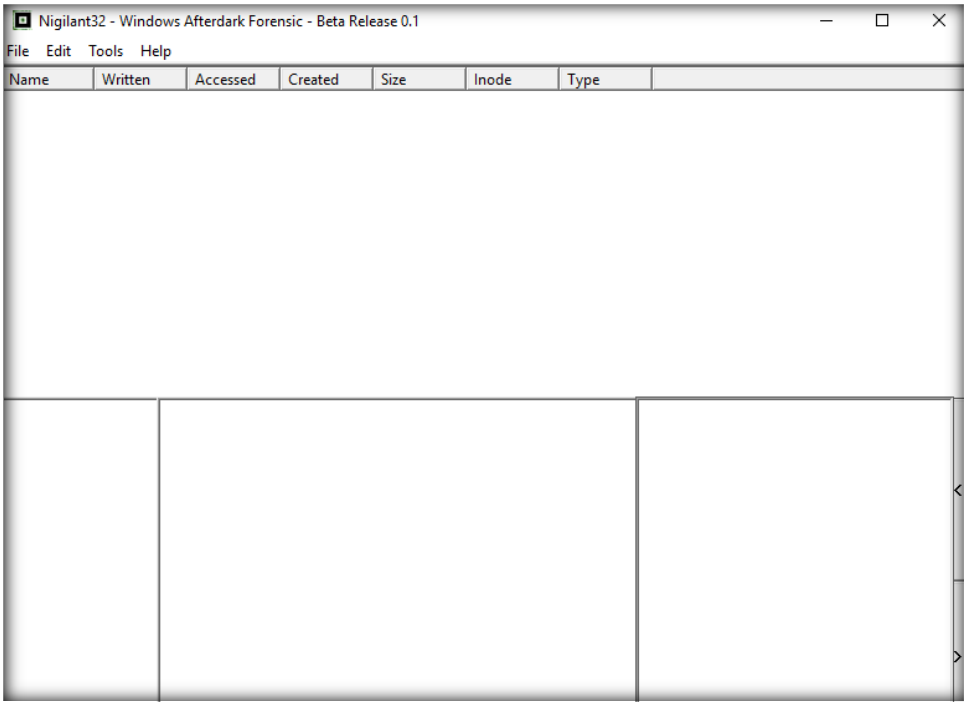


FIGURE 7.31: Main screen of Helix Incident Response Nigilant32

37. Choose **File → Preview Disk...** to see a preview of the hard drives

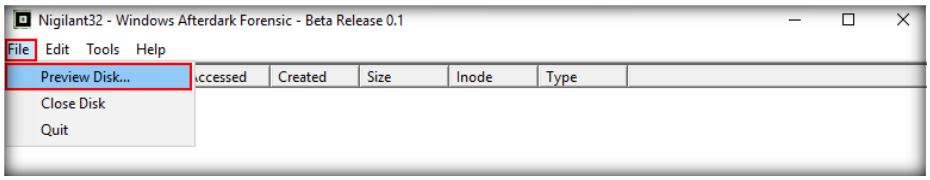


FIGURE 7.32: Preview Disk feature in Helix Incident Response Nigilant32

38. Select the drive to preview and then click the **Apply** button. Here, we are selecting the first drive.

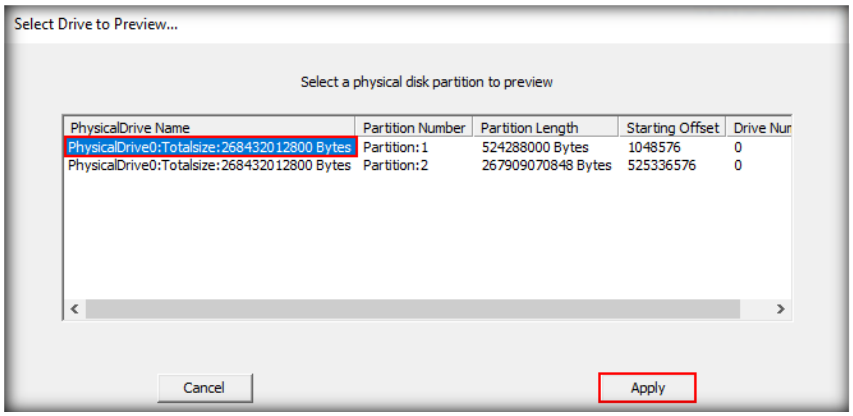


FIGURE 7.33: Selecting a drive to preview in Helix Incident Response Nigilant32

39. The tool now displays the files and folders pertaining to the partition. Double-click a file to view its hex-equivalent data in the lower pane of the window. Here, we have double-clicked on **$MFTMirr** to view its contents.

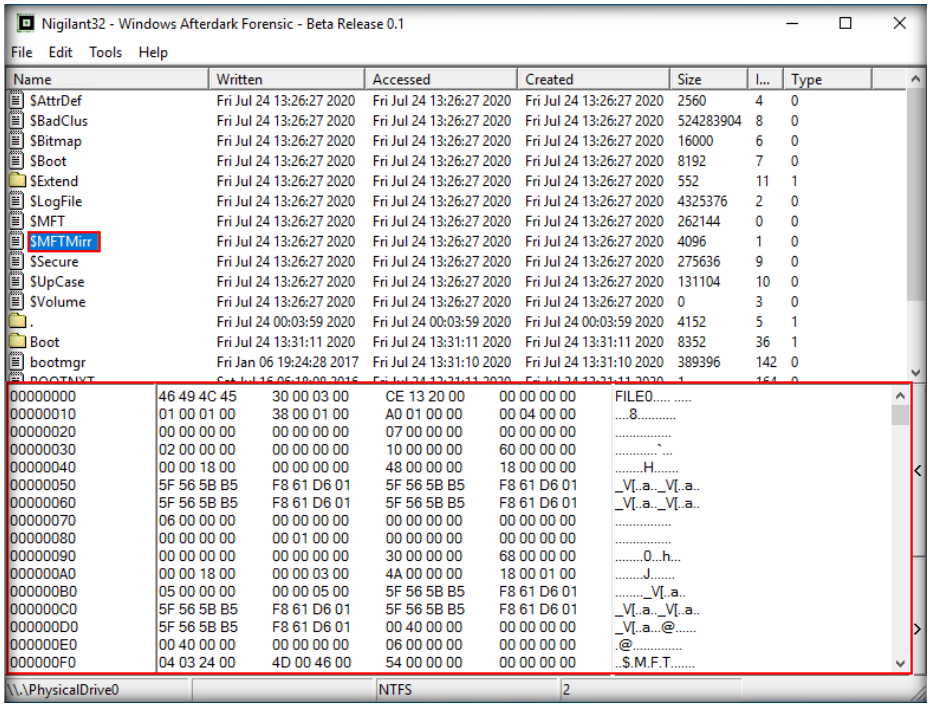


FIGURE 7.34: Preview of the $MFTMirr file in Helix Incident Response Nigilant32

Nigilant32 For First Responders: By using the active memory imaging feature of Nigilant32, we can image the active physical memory (RAM) of the suspect workstation or server to secure portable media.

40. To take a snapshot of the computer, choose **Tools → Snapshot Computer**.

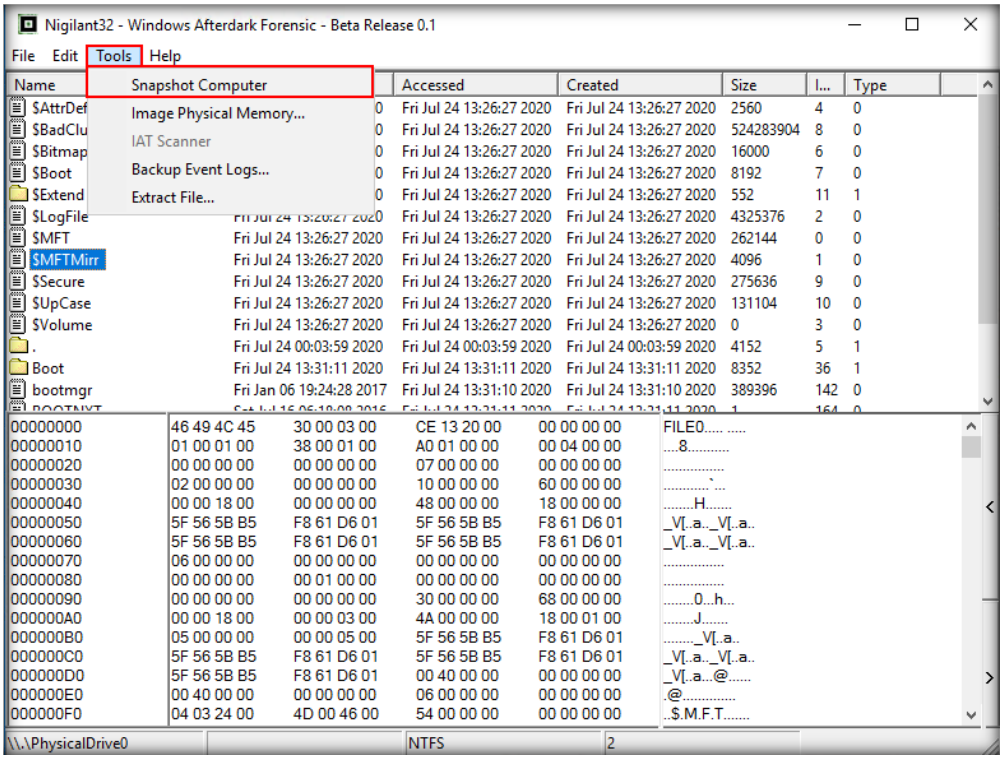


FIGURE 7.35: Take a snapshot of the computer in Helix Incident Response Nigilant32

Helix also has a special Windows autorun side for incident response and forensics.

41. The tool will display the **Live Machine Snapshot** window. If you wish to save the snapshot, click the **Save** button.

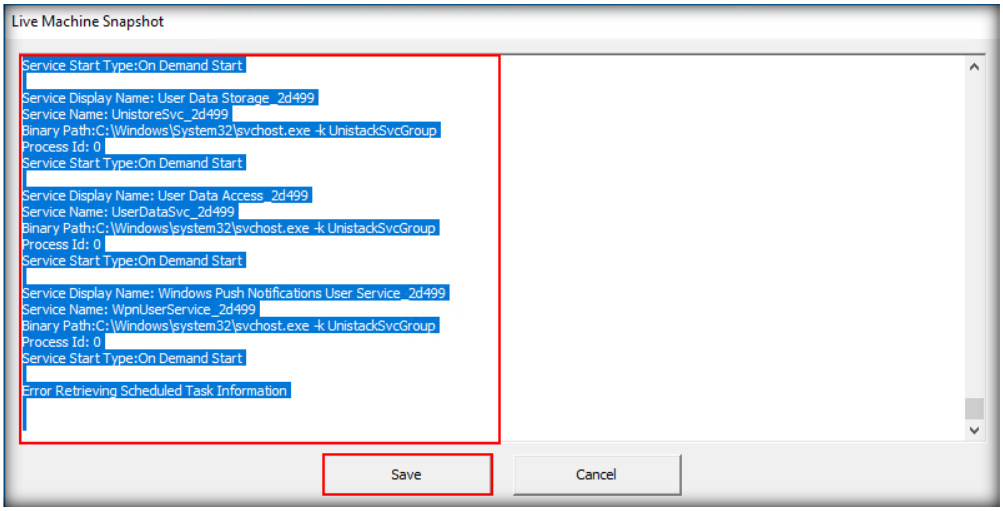


FIGURE 7.36: Live Machine Snapshot window in Helix Incident Response Nigilant32

Helix focuses on incident response and forensics tools. It is meant to be used by individuals who have a sound understanding of these techniques.

42. The **Save As** window appears. Select the location where you wish to save the snapshot, specify the filename (here, **Live Machine Snapshot**) in the **File name** field, and click **Save**. The file is saved as a **.txt** file. Here, we are saving the file to **Desktop**.

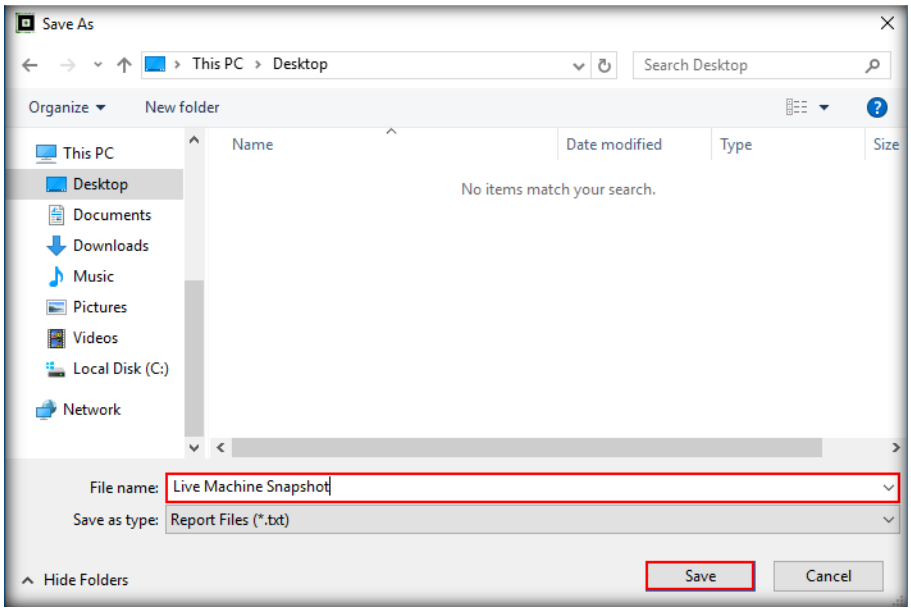


FIGURE 7.37: Save As window in Helix Incident Response Nigilant32

43. Now, close the **Nigilant32 - Windows Afterdark Forensic** window and return to the main window of Helix. Click the **arrow** icon against the **Incident Response** section to move to its second page, as highlighted in the screenshot:

FIGURE 7.38: Navigate to Page 2 of Helix Incident Response

Nigilant32, , a freeware Windows GUI, is a recently released incident response tool.

TASK 5

Examine a File

44. The second page of the **Incident Response** section contains the **FILE** and **MD5** fields.

45. In this task, we are going to check a file's hash and its legitimacy. We will first calculate the file's hash and then search for that file or its hash in the VirusTotal database.

46. Click the **ellipsis** button against the **FILE** field to provide an evidence file as input. A **Select File** window will appear. Navigate to **C:\CHFI-Tools\Evidence Files**, select the file **Infected.docx**, and then click **Open**. Subsequently, click on the **Hash** button to generate the **MD5** hash value of the **Infected.docx** file. The MD5 hash value of **Infected.docx** will be generated as shown in the screenshot.

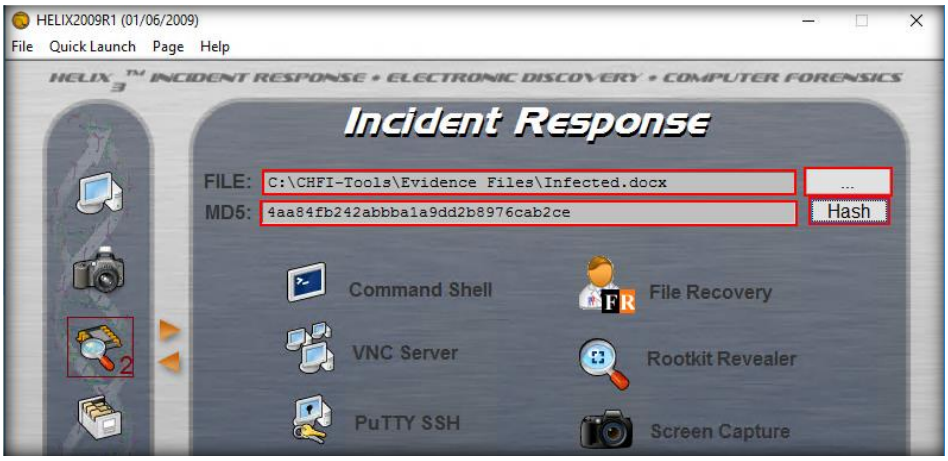


FIGURE 7.39: MD5 calculation in Page 2 of Helix Incident Response

The Message-Digest 5 (MD5) algorithm is a widely used cryptographic hash function that produces a 128-bit (16-byte) hash value. It is specified in RFC 1321.

MD5 has been employed in a wide variety of security applications and is commonly used to check data integrity.

47. Now, we will look up the VirusTotal database to determine whether the file is safe or malicious. Launch a browser and open the URL https://www.virustotal.com/gui/home/upload. The VirusTotal homepage appears. In this lab, we will upload the file to VirusTotal. Click **Choose File** as shown in the screenshot.

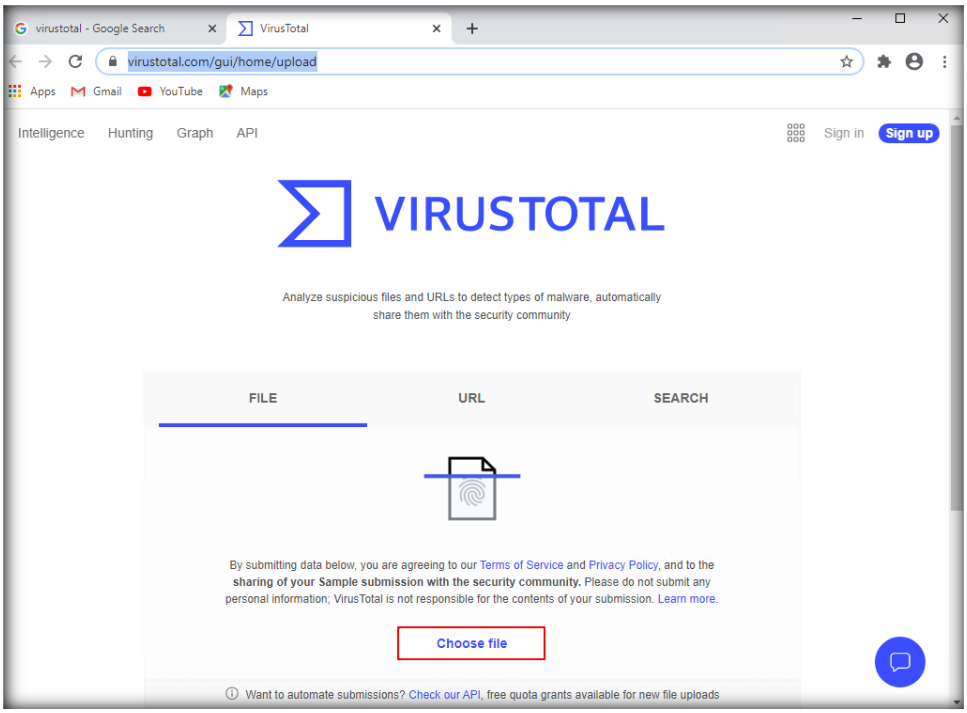


FIGURE 7.40: Check the file integrity in the VirusTotal database

48. An **Open** window will appear. Navigate to **C:\CHFI-Tools\Evidence Files**, select the **Infected.docx** file, and click **Open**.

49. Upon uploading the file, the **VirusTotal** website indicates that the file has been detected as malicious by several anti-virus engines, as shown in the screenshot:

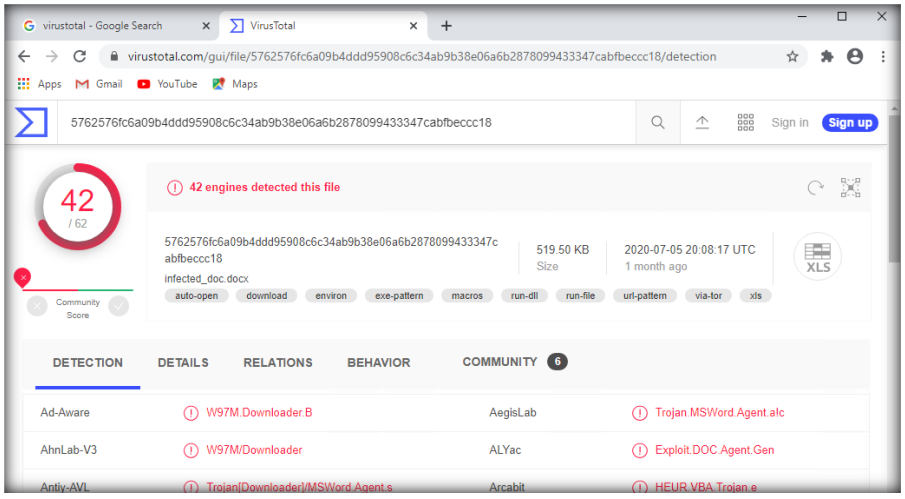


FIGURE 7.41: Verify file integrity on the VirusTotal website

Note: For demonstrative purposes of this lab, the malicious file has been represented by the name **Infected.docx**. In real-time, a malicious file may appear as a normal, harmless file. Alternatively, instead of uploading the file to check for its malicious nature, you may copy and paste its MD5 hash value that we calculated using Helix in the **SEARCH** section of the VirusTotal website. This operation will also reveal whether the file is malicious.

50. Similarly, you can follow the above procedure for other files as well to check if they are malicious.

TASK 6

Recovering Deleted Files

51. We will now proceed to the task of **recovering deleted files**.

52. To perform the task of recovering deleted files, click the **File Recovery** option in the **Incident Response** section of the **Helix** window, as shown in the screenshot:

FIGURE 7.42: File Recovery in Page 2 of Helix Incident Response

53. You will be prompted with a **Notice** pop-up. Click **Yes** to run **Filerecovery.exe**.

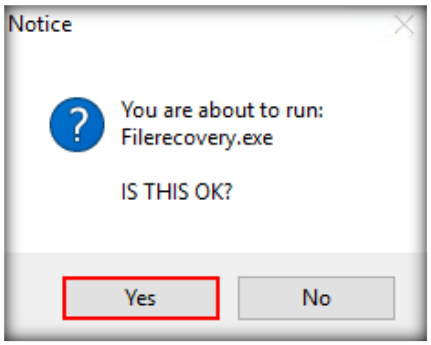


FIGURE 7.43: File Recovery notice in Helix Incident Response

54. Select a language and click the **tick mark** icon in the **Choose Language** wizard

FIGURE 7.44: Language selection in File Recovery of Helix Incident Response

55. Now, click the **cross mark** icon to close the **Welcome to PC Inspector File Recovery** window

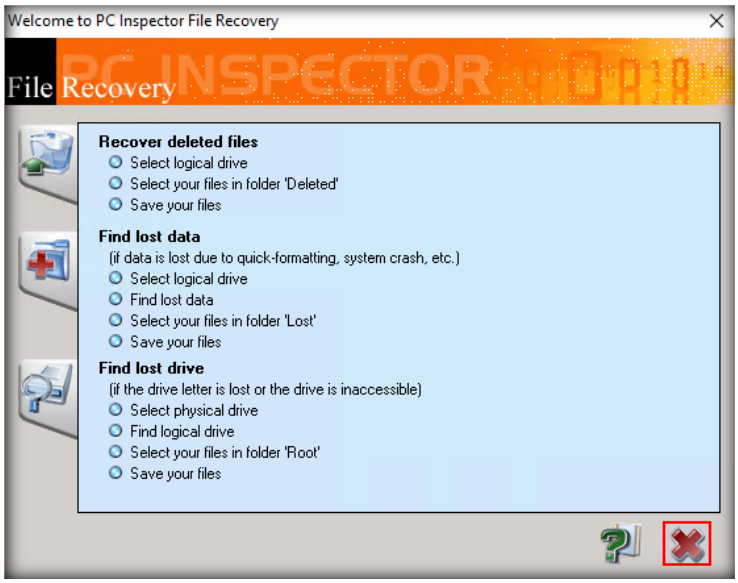


FIGURE 7.45: Welcome window of the File Recovery wizard in Helix Incident Response

Supported file systems: FAT 12/16/32, NTFS (used by hard disks, disks, Smart media™, Compact Flash™, Memory Stick™, and others)

Select your file (for several files, use the **Ctrl** or **Shift** keys) and save it to another drive by selecting **Save to** from the **Object** menu.

Moreover, you can search for a deleted or lost file (or for several files) by selecting **Find** from the **Object** menu.

Select **Find lost data** from the **Tools** menu. The dialog windows that appear allow you to delimit the search (e.g., if you know that the file/directory is situated further behind, move the track bar Start sector to the right).

56. The **PC Inspector File Recovery** window will now appear. Navigate to **Object → Drive**.

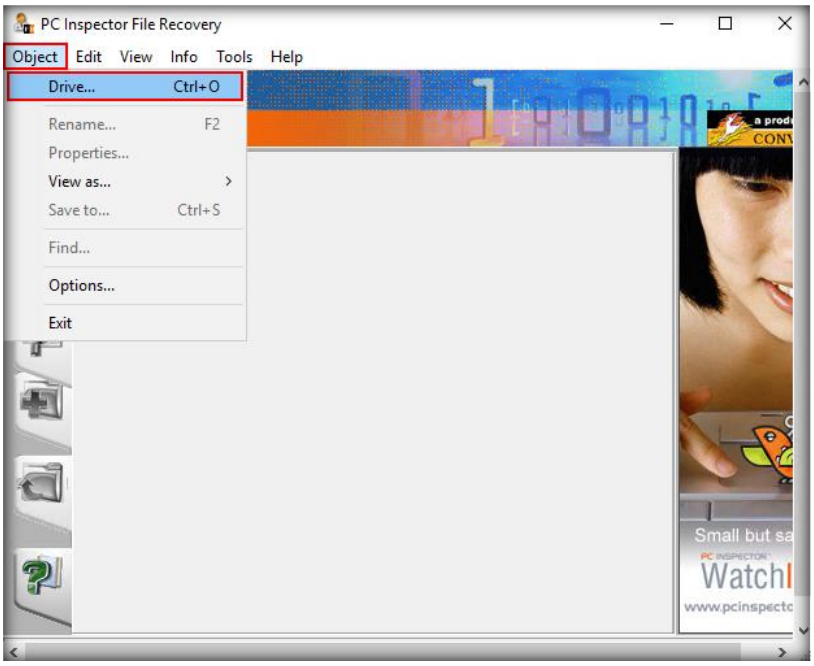


FIGURE 7.46: Object menu of File Recovery in Helix Incident Response

57. Now, select a drive from the **Logical drive** or **Physical drive** tabs. Here, we are recovering files from a logical drive.

58. Select the **C:** drive from the **Logical drive** tab and click the **tick mark** icon

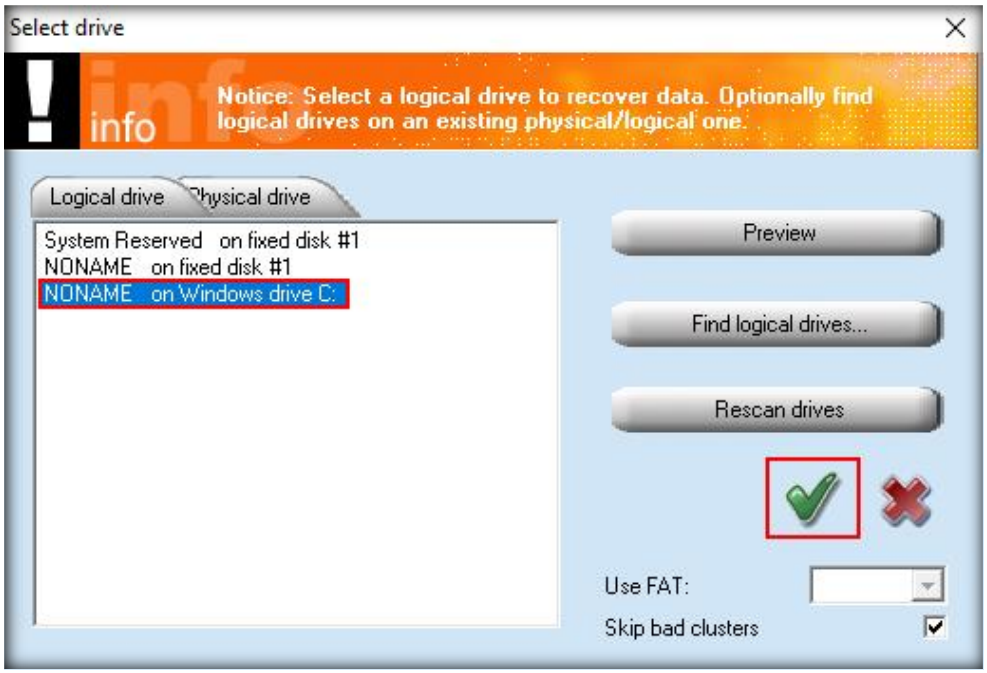


FIGURE 7.47: Selecting a drive in File Recovery in Helix Incident Response

Important: If a file is found with an unknown name and size (e.g., cluster40.jpg), you can correct the name and size by choosing **Properties** from the **Object** menu. If the file is too small, simply change the file size to a higher value.

59. The tool will take some time to retrieve the files and folders of the drive.

60. In the **left pane** of the window, the tool will display a tree structure under the **Folders** section. In this section, click the **Deleted** items option. All the deleted contents of **C:** drive will now be listed in the right pane of the tool window, as shown in the screenshot:

Note: The output result might vary in your lab environment.

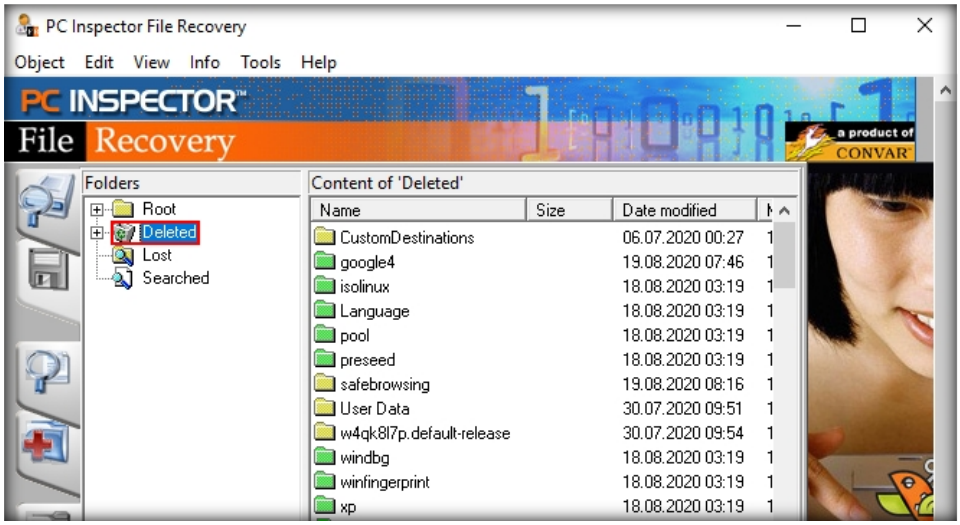


FIGURE 7.48: Deleted items node

61. To recover a deleted a file, select the desired file and then click the **Save** icon on the left of the tool window, as indicated in the screenshot below:

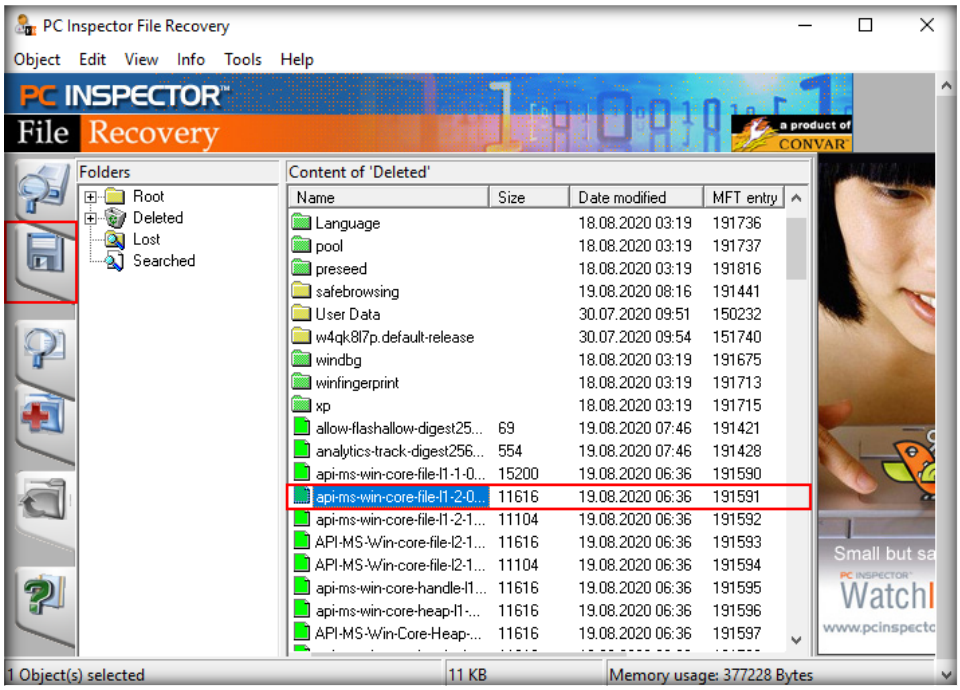


FIGURE 7.49: Recovering deleted contents

In addition to the cluster scan, you can search for a deleted/lost file (or for several files) by selecting **Find** from the **Object** menu.

Important: To open the drive and to save your data, choose the option **no FAT (consecutive)** since the FAT has been deleted by quick formatting.

62. Upon clicking the **Save** icon, a **Select Directory...** window will appear. Choose the directory to which you wish to save the selected deleted file. Here, we are saving the file to the default location, i.e., **Desktop**. Click the **tick mark** icon after choosing the location for saving the file, as shown in the screenshot below. The selected file will be saved to the chosen destination (in this case, **Desktop**).

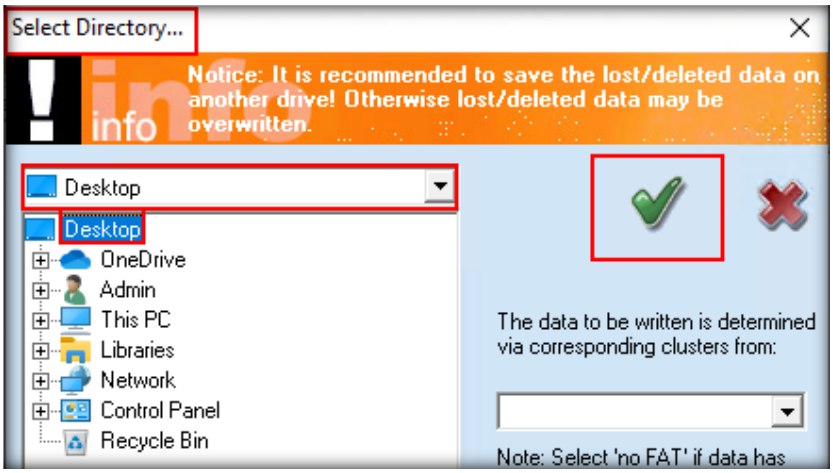


FIGURE 7.50: Select a location for file recovery

63. We will now proceed to the task of browsing contents on a hard drive.
64. Close the **PC Inspector File Recovery** window.
65. To find the contents of the system's hard drive, return to the main window of Helix and click the **Browse** icon in the left pane. This operation lists all the drives of the system in the middle pane of the window.

TASK 7

Browsing Contents on a Hard Drive

After the search has been completed and a logical drive has been found, select the drive under the logical drive if the format seems to be correct. It is not exceptional to find more drives than has been installed. The reason is that there are several copies of the boot sector on the hard disk (which is the unit that contains the information about a logical drive). So, first use the Preview function to obtain the correct logical drive.

FIGURE 7.51: Drives on a system

66. Click on one of the listed drives (here, **C:** drive) to find its node on the left and then expand the node.
67. The tool will display all the folders stored within the selected drive.
68. Click on one of the listed folders to find its node against it on the left. Expand the folder node to list the files/contents stored within them.

Here, we are expanding the node against the folder **CHFI-Tools** and then further expanding the node against the sub-folder **Evidence Files**. Under the **Evidence Files** sub-folder, click the **Image Files** folder to see its contents in the right pane of the tool window, as shown in the following screenshot.

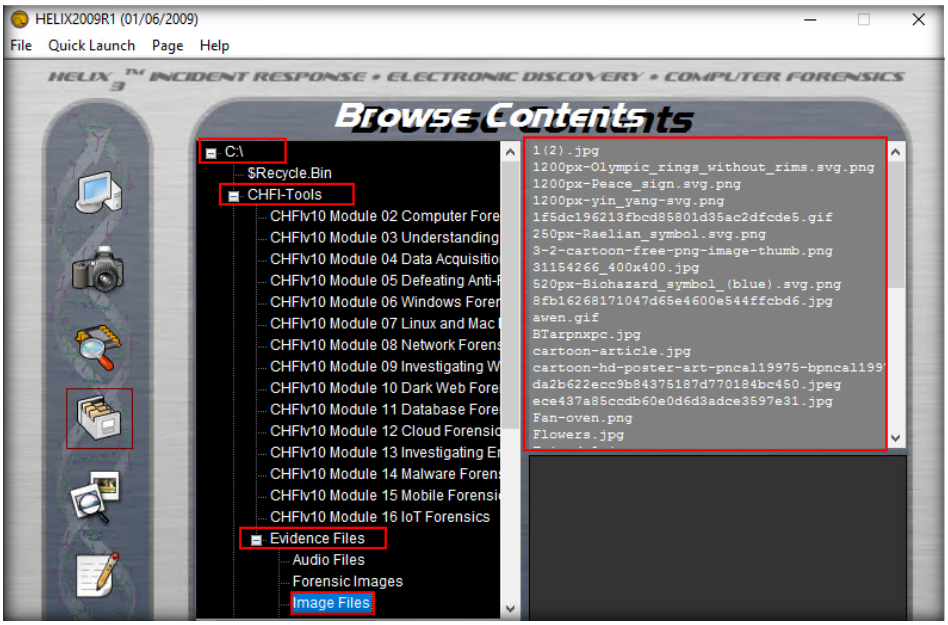


FIGURE 7.52: Expanding a drive and its folder(s)

69. Now, select an image file in the right pane. The application displays the file's properties in the lower-right section of the window, as shown in the screenshot:

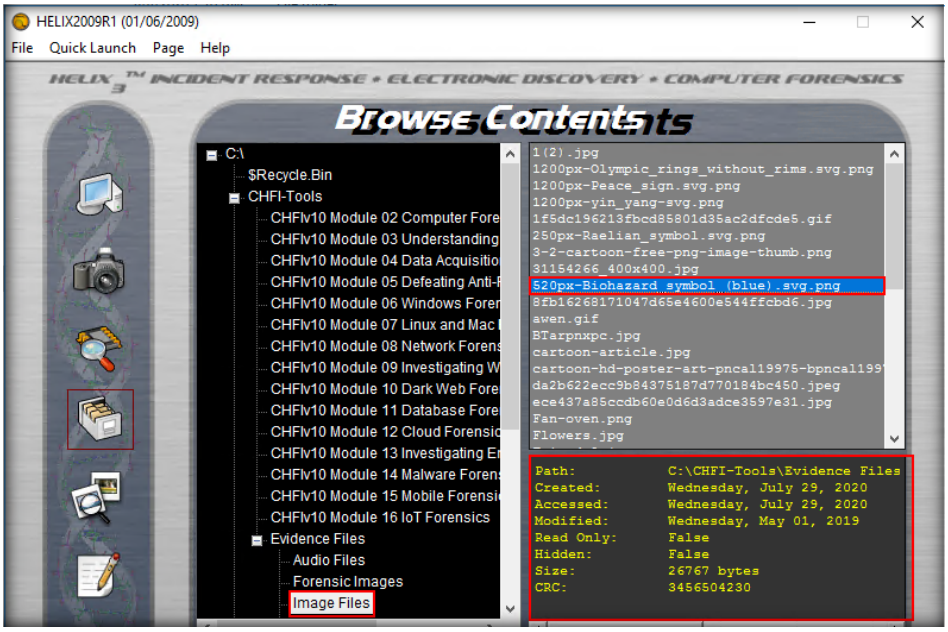


FIGURE 7.53: Browsing contents in Helix

Helix is a customized version of Ubuntu Linux, allowing you to boot into a Linux environment containing customized Linux kernels, hardware detection, and a large number of applications designed for incident response and forensics.

TASK 8

Scanning for Pictures

70. To scan images or pictures on the computer, click the **Scan for Pictures** icon and then click **Load Folder** to view the images.

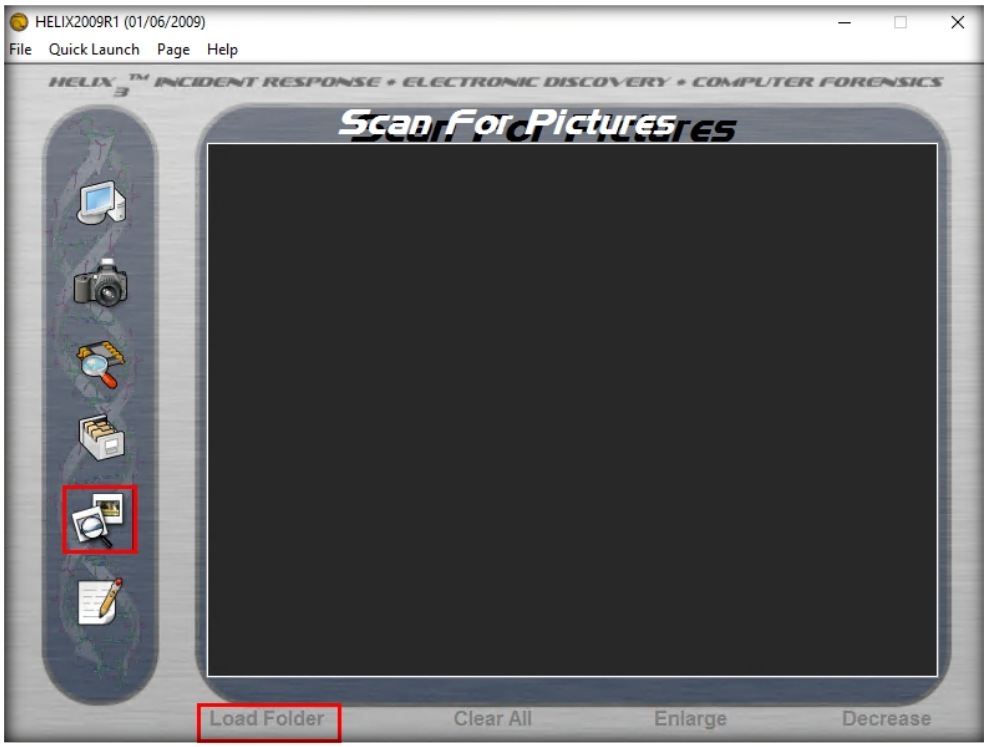


FIGURE 7.54: Scan for pictures in Helix

Helix is also unique in that it includes a Win32-based autorun toolset, providing users quick access to common forensic tools not found in Linux. Helix is a customized distribution of the Knoppix Live Linux CD. Helix is more than just a bootable live CD.

71. A **Browse For Folder** window appears. Navigate to **C:\CHFI-Tools\Evidence Files\Image Files** to load the pictures for scan and then click **OK**.

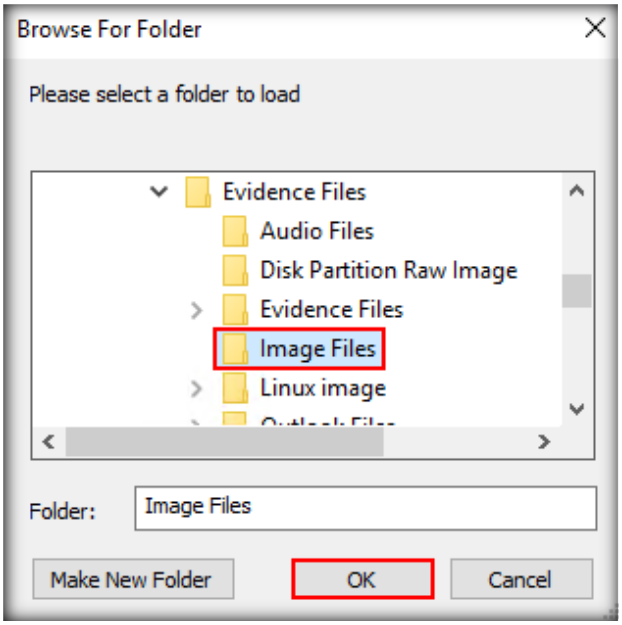


FIGURE 7.55: Select a location to scan for pictures in Helix

Helix will not auto-mount swap space or any attached devices. Helix also has a special Windows autorun side for incident response and forensics. Helix has been modified very carefully *not to* touch the host computer in any way, and it is forensically sound.

72. A **Notice** pop-up appears, asking you to be patient. Click **OK**.

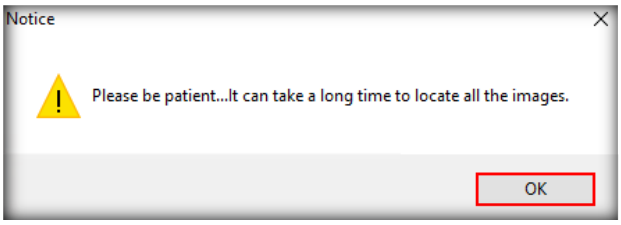


FIGURE 7.56: Notice appearing when scanning for pictures in Helix

73. The tool will load the previews of all images within the folder and display them as shown in the screenshot:

FIGURE 7.57: Preview of scanned images in Helix

74. In this manner, you can create images, retrieve deleted files, and scan for image files on a computer by using Helix.

Lab Analysis

Analyze drive image creation and extraction, file recovery, and document the respective details.

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

Lab

8

Collecting and Parsing Forensic Artifacts on a Live Windows Machine

In some situations, investigators cannot seize the suspect machine to start the forensic investigation. In such cases, they need to use appropriate forensics tools that allow them to collect and process data from a live system and accelerate the pace of investigation.

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

Lab Scenario

Robert, a forensic investigator, was called by a German finance company to examine the system of an employee responsible for important payroll-related processes. The employee complained that some confidential files were missing from his system. Robert enquired further and learnt that most of the sensitive files were stored in the system drive of the machine.

Shutting down and seizing the machine for forensic analysis was not a viable option as it was being used for mission-critical purposes. Therefore, Robert decided to gather data regarding the file system and other artifacts relevant to the case and parse them quickly by using the KAPE tool, even before the full system image was taken, so that he could start the forensic analysis at the earliest.

Lab Objectives

This lab will help you understand how to identify and collect forensically useful artifacts from a Windows machine and parse them simultaneously to expedite the investigation.

Lab Environment

This lab requires:

- A computer running **Windows Server 2016** virtual machine
- Administrative privileges to run tools
- A web browser with internet access

- **KAPE** tool located at **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\Windows Forensics Tools\KAPE**

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics

Note: You can download the latest version of **KAPE** tool from the link https://www.kroll.com/en/services/cyber-risk/incident-response-litigation-support/kroll-artifact-parser-extractor-kape

If you are using the latest version of the tool for this lab, then the steps and screenshots demonstrated in the lab might differ.

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 40 minutes

Overview of the Lab

This lab familiarizes you with **Kroll Artifact Parser and Extractor** or **KAPE**, an open-source triage application that has both forensic data collection and parsing capabilities. In this lab, you will learn how to collect files of evidentiary value via the **KAPE** tool and then process the collected data for forensic analysis.

Note: It is not possible to acquire the image of a full drive via the **KAPE** tool. It can collect one or more files as per the **Target options** and then parse them via one or more **Modules**. It thus allows investigators to **collect** and **process artifacts** most critical to their case before starting the imaging process. Artifacts obtained using **KAPE** can be used gain insight into events and build a timeline, thus accelerating the forensic investigation process.

Lab Tasks

TASK 1

Open Kape and Explore Its Components

1. In this lab, we will demonstrate how to perform live collection and processing of artifacts via the **KAPE** tool.
2. Before beginning this lab, ensure that you have **Microsoft Office** application installed on your machine.
3. Login to the **Windows Server 2016** virtual machine.
4. Open **File Explorer**, navigate to **C:\CHFI-Tools\CHFIv10 Module 06 Windows Forensics\ Windows Forensics Tools\KAPE**, and double-click **gkape.exe**.

5. The **KAPE GUI** appears, as shown in the screenshot below:

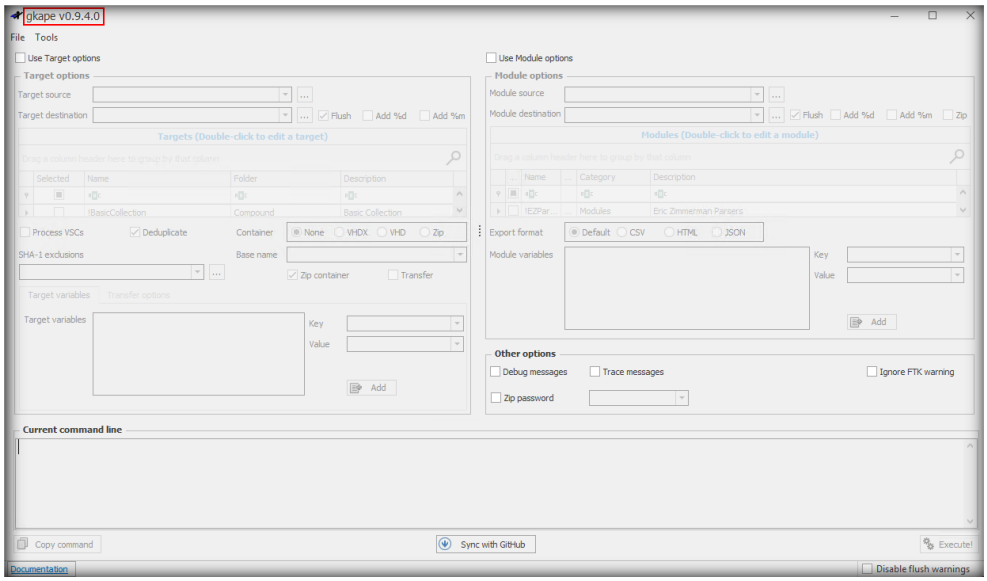


FIGURE 8.1: KAPE GUI

6. Before proceeding with the lab, it is important to note different components of the tool. The top-left side has an entry named **Use Target options**. When that entry is checked, it enables all the **Target options** embedded into the tool, as shown in the screenshot below:

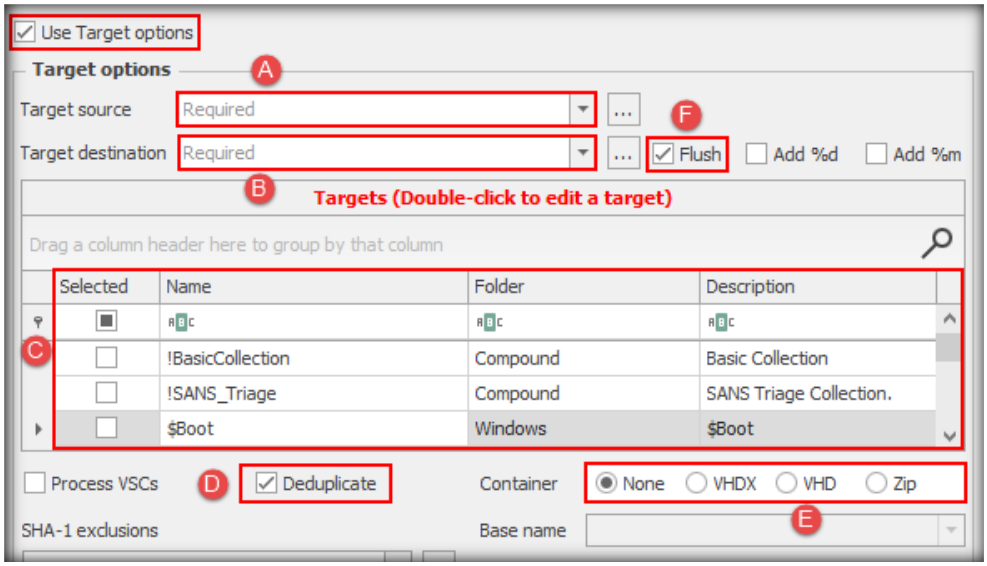


FIGURE 8.2: View all the Target options on the KAPE tool

7. On checking the **Use Target options** entry, we see the following **fields** and **options** getting enabled on the tool:

   A. **Target source**: Here, you can specify the drive letter you wish to collect the files from. In this lab, we will be selecting the **C:** or system drive.

B. **Target destination**: This is the location where the collected files will be stored. For live collection, it should usually include the path to an **external drive**. However, as it is a lab scenario, we will be saving the tool outputs in the **C:** drive itself.

C. **Targets**: This is the location from which you can select the types of files that the tool will collect. It includes an extensive list of **Targets** that you can select and edit as per the case requirements.

D. **Deduplicate**: With this option checked, **KAPE** will remove any duplicate file by matching their SHA 1 hash values. This option is already **checked** by default.

E. **Container**: This field determines the format in which files will be collected. Available formats for use are **VDHX**, **VHD**, and **Zip**.

F. **Flush**: This option flushes or removes the contents of the **Target destination** before collecting any data. This option is checked by default.

8. Immediately next to the **Target options** is an entry named **Use Module options**. Upon enabling this entry, you will see all **Module options** embedded in the tool, as shown in the screenshot below:

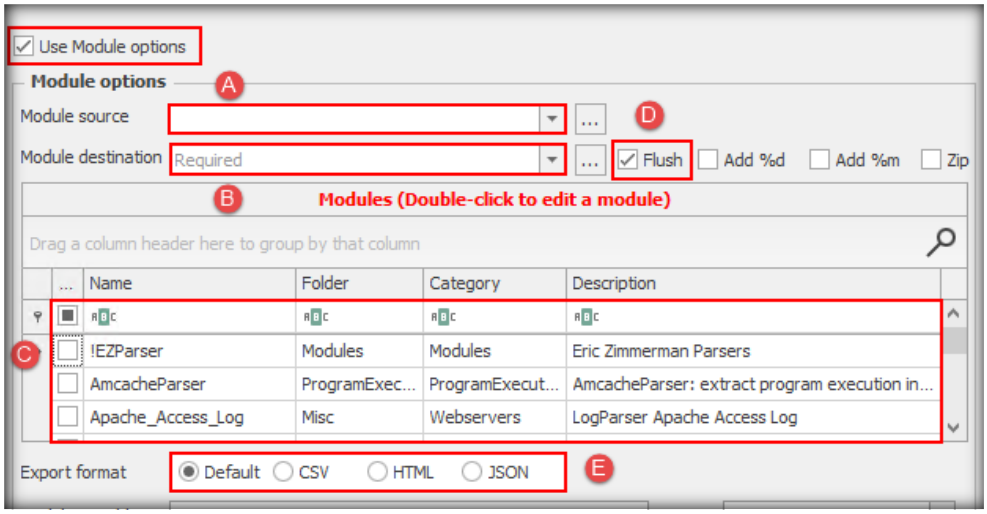


FIGURE 8.3: View the Module Options on the KAPE tool

9. On checking the **Use Module options** entry, we see the following **Module sections** and **options** getting enabled on the tool:

A. **Module source**: Here, the path of the **Target destination** where you copied the files should be mentioned.

B. **Module destination**: Here, the folder path is to be specified where the parsed copies of the **Target files** will be saved.

C. **Modules**: This section contains the names of the **Modules** belonging to specific **executables**, using which the selected **Target files** will be parsed. These **executables** are stored in the **bin folder** of the **KAPE tool**.

D. **Flush**: The **Flush** option is **checked** by default. It flushes the contents of the **Target/Module destinations** before **storing/parsing** any data.

E. **Export Format**: This section specifies the format in which the parsed data will be generated. The default option is the **CSV** format, along with **HTML** and **JSON**.

10. Immediately below the **Target options** and **Module Options**, you will see the **Current command line** section and other buttons:

A. **Current command line**: This is the **command line section** of the tool. Any **commands** provided in the tool is recorded in this section.

B. **Copy Command**: This button helps in copying the command from the **Current command line** and run it on **Command Prompt**.

C. **Sync with GitHub**: Clicking this button draws updated **Targets** and **Modules** from the **GitHub** repository. However, the required **executables** need to be downloaded and placed in the **KAPE/bin** folder manually.

D. **Execute!**: On selecting all the desired **Targets/Modules** and setting the folder path for **Target** and **Module sources** and **destinations**, you can click on the **Execute!** button to executes the command.

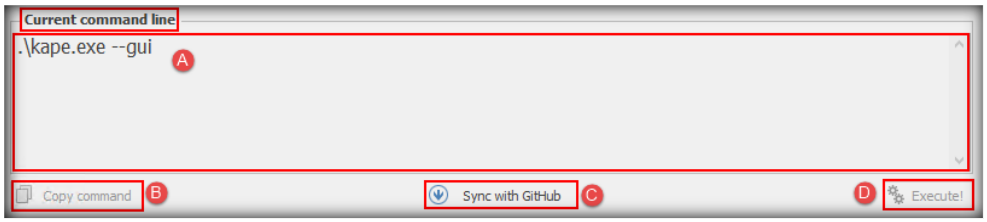


FIGURE 8.4: Current command line and other buttons on KAPE

TASK 2

Create Target and Module Destination Folders and Acquire Target Files

11. Now, we need to create the **Target** and **Module destination** folders, which will be required while running the tool.

12. Minimize the **KAPE GUI** window. Open **File Explorer**, navigate to **This PC → Local Drive (C:)**, and create a new folder named **KAPE Output**.

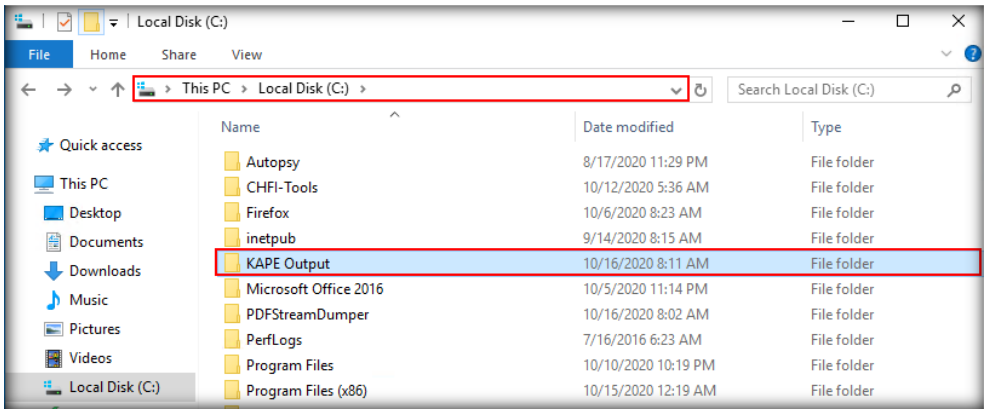


FIGURE 8.5: Create a folder named KAPE Output in C drive

13. Now, go inside the **KAPE Output** folder. Create two folders named **KAPE Target Output** (this folder will be used as the **Target destination**) and **KAPE Module Output** (this folder will be used as the **Module destination**).

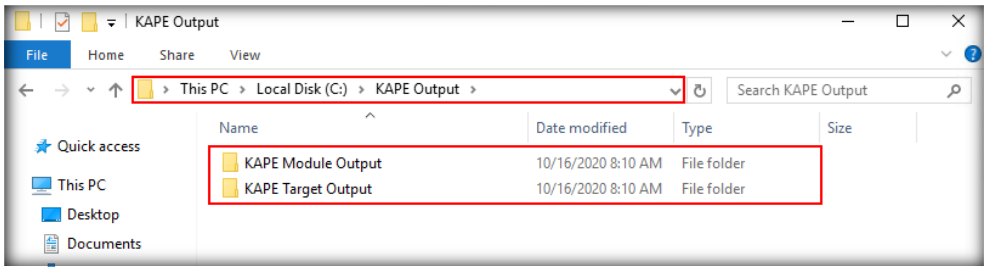


FIGURE 8.6: Create folders named KAPE Target Output and KAPE Module Output

14. Now, we can create **Target files** by copying files from the machine. To do so, return to the **KAPE GUI** window.

15. Ensure that the **Use Target options** entry is checked. To select the **Target source** folder, click on the **Ellipsis (...)** beside it.

Note: Ensure that you uncheck **Use Module options** in case it is checked

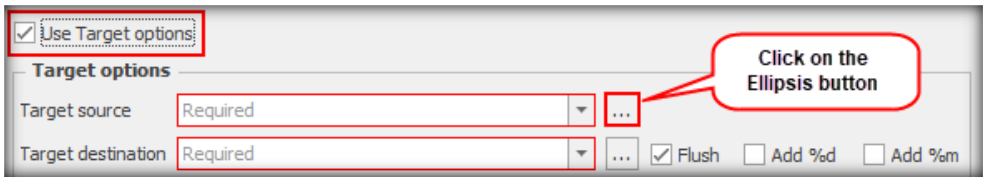


FIGURE 8.7: Click the Ellipsis button to select a Target source

16. The **Browse for Folder** window appears. Select **Local Disk (C:)** and Click **OK**.

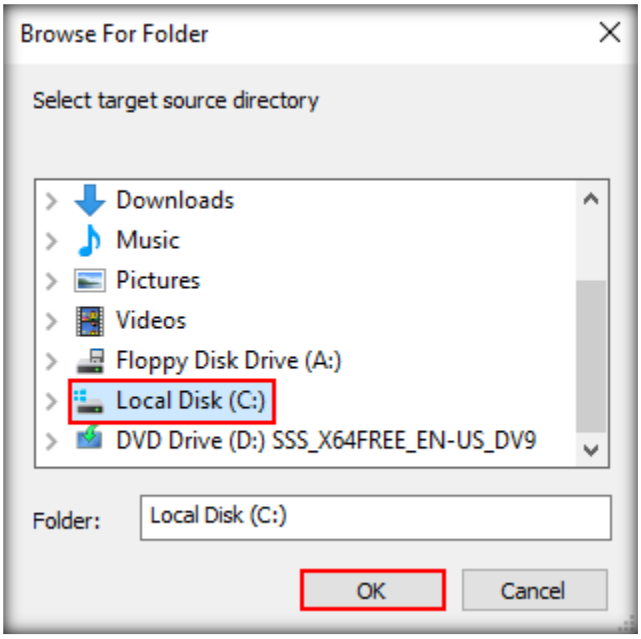


FIGURE 8.8: Select Local Disk (C:) as the target source

17. Similarly, to set the **Target destination** folder, click on the **Ellipsis (...)** button beside it. When the **Browse For Folder** window appears, navigate to **Local Disk (C:)** → **KAPE Output** → **KAPE Target Output**. Once done, click **OK**.

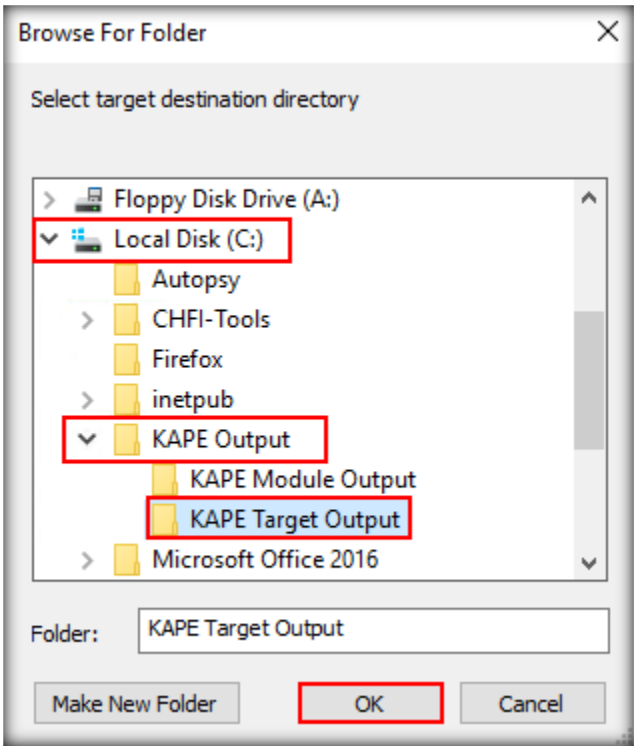


FIGURE 8.9: Select the folder KAPE Target Output as the target destination

18. In the **Targets** section, we will select **Targets** related to **Filesystem**, **LNK files**, and **Jump Lists**. To find **Filesystem** related **Targets**, place the cursor below the **Name** field beside the letters **RBC** and type **FileSystem**. Once the **Target entry** named **FileSystem** is found, **select** the checkbox associated with it.

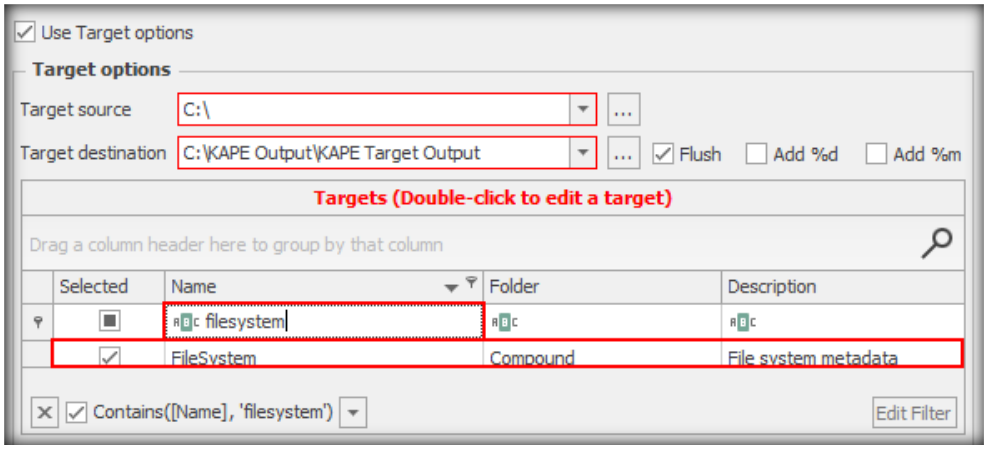


FIGURE 8.10: Select FileSystem as one of the Targets

19. Now, **double-click** on the **Target entry** named **FileSystem** to view its contents or edit it.

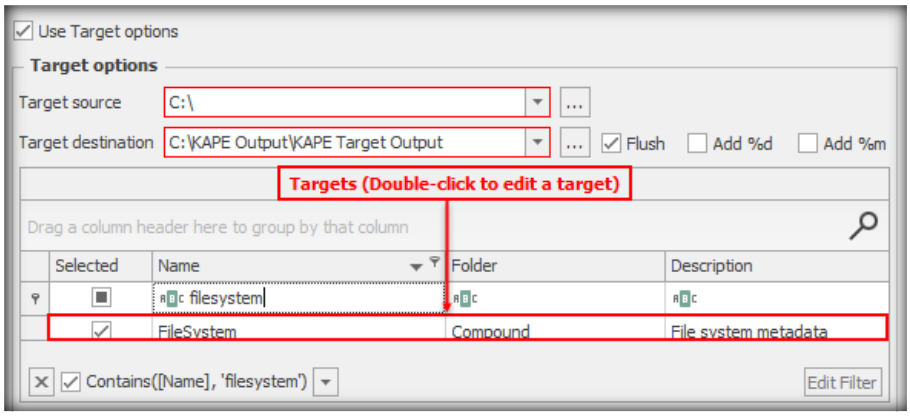


FIGURE 8.11: Double-click the FileSystem Target entry

20. An **Editor: FileSystem** window opens, as shown in the screenshot below:

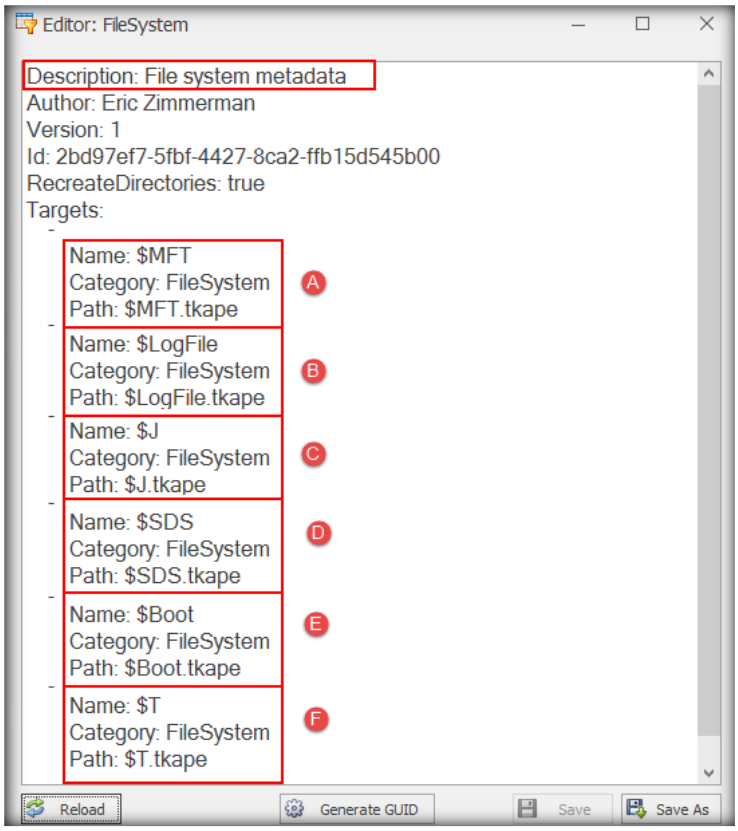


FIGURE 8.12: View information on the Editor: FileSystem window

21. Here, you can see the files in the **NTFS filesystem** that the selected **Target** would copy:

    A. **$MFT**: Also known as **Master File Table**, this file contains a series of attributes for each file in the specified file system and is of utmost forensic value.

    B. **$LogFile**: This file tracks changes made in the **NTFS filesystem**.

C. **$J**: This file is a part of the $UsnJrnl file, which includes the contents of the change journal.

D. **$SDS**: Also known as **Security Descriptor Stream**, this file stores a list of all **Security Descriptors** in the **NTFS** volume.

E. **$Boot**: This is the file copy of the boot sector of the specified **NTFS** volume.

F. **$T**: This file belongs to **TxF Old Page Stream (TOPS)** file data. It includes the **$T** data stream, which contains data about files that are partially overwritten.

22. Close the **Editor: FileSystem** window. Now, In the **Target section**, clear the RBC cell, type **LnkFilesAndJumpLists** and select the checkbox associated with **LnkFilesAndJumpLists**. This **Target entry** would copy **LNK files** and **Jump Lists** from the specified drive.

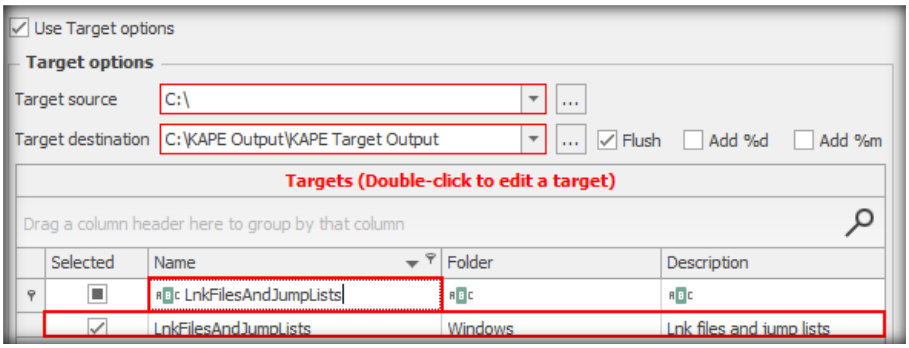


FIGURE 8.13: Select LnkFilesAndJumpLists as one of the Targets

23. Ensure that the **Flush** and **Deduplicate** options are checked, and keep the **Container** selected as **None**.

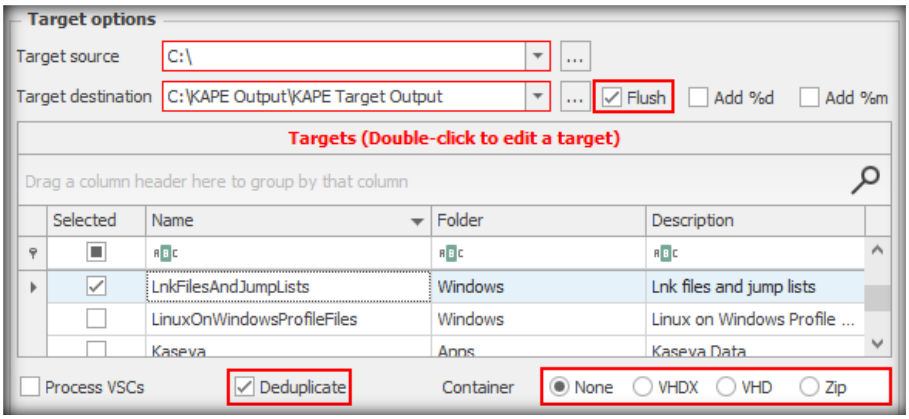


FIGURE 8.14: Keep Flush and Deduplicate options checked

24. Once both the **Target entries** are selected, the **Current command line** section would appear as shown in the following screenshot:

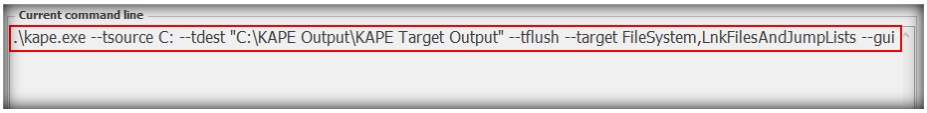


FIGURE 8.15: Current command line after selecting the Targets

25. In the highlighted section of the screenshot above, the **--tsource** parameter specifies the **Target source** folder, and the **--tdest** parameter defines the path of the **Target destination** folder. The **--tflush** parameter indicates that the **Flush** option is checked, the **--target** parameter shows the selected **Target entries**, and the **--gui** parameter indicates that the command will to be executed using the GUI of the tool.

26. Once the **Target options** are all configured, click on the **Execute!** button

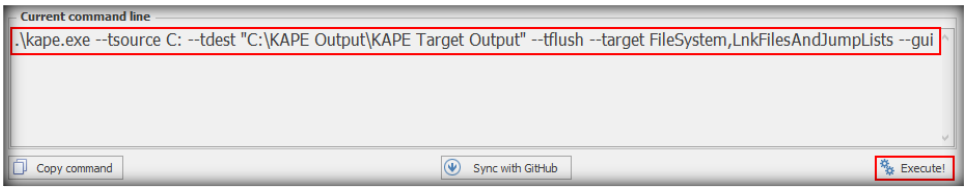


FIGURE 8.16: Click the Execute! button

Note: If a **DATA DESTRUCTION WARNING!** window appears, click **OK** to continue.

27. The tool will now open a command-line window showing the progress of the **Target files** being copied. Once all the selected **Target files** are copied to the **Target destination** folder, the tool will prompt you to **press any key to exit**.

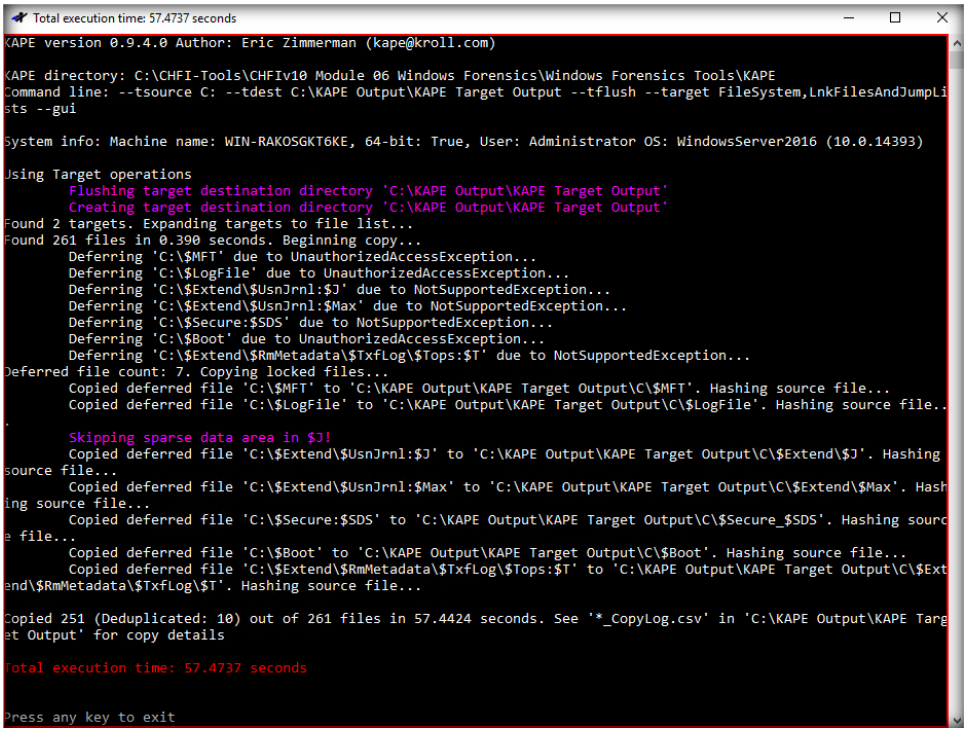


FIGURE 8.17: Command-line window generated by KAPE tool while copying the Target files

Note: The tool takes some time to copy the selected **Target files**

28. Press any key to exit the command-line window. Minimize the **KAPE** application window.

TASK 3

Check the Acquired Target Files in Target Destination Folder

29. Now, open **File Explorer** and navigate to **C:\KAPE Output\KAPE Target Output**. You will see a folder called **C**, along with **three files**, as shown in the screenshot below:

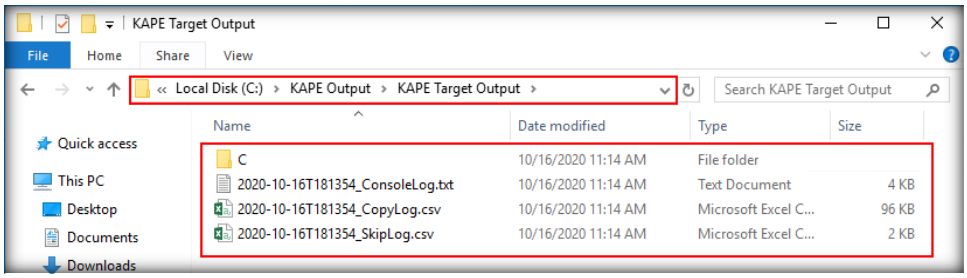


FIGURE 8.18: View Target files in the folder KAPE Target Output

30. All the files in the highlighted section of the above screenshot start with the **date** when the **Target files** were copied from the machine, followed by some **alphanumerical characters**. The file **2020-10-16T181354_ConsoleLog.txt** bears the same information as what was displayed in the command-line window, the file **2020-10-16T181354_CopyLog.csv** includes information about the **Target files** that were copied, and the file **2020-10-16T181354_SkipLog.csv** includes information about the **Target files** that were skipped during the **Target operations** by the tool.

Note: The names of files shown above will differ in your lab environment.

31. To view the copied items, double-click the **C** folder. You will see some of the copied files, namely, **$Boot**, **$LogFile**, **$MFT**, and **$Secure_$SDS**, and two folders: **$Extend** and **users**.

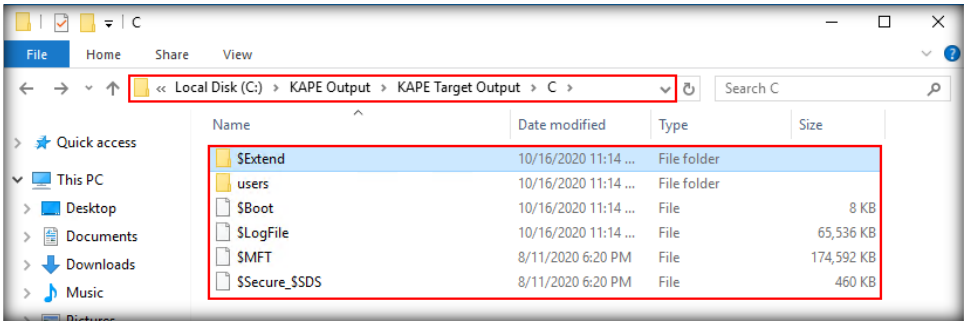


FIGURE 8.19: View the contents of the C folder in the KAPE Target Output folder

32. You will also observe the files **$J** and **$Max** as well as a folder called **$RmMetadata** upon double-clicking the **$Extend** folder.

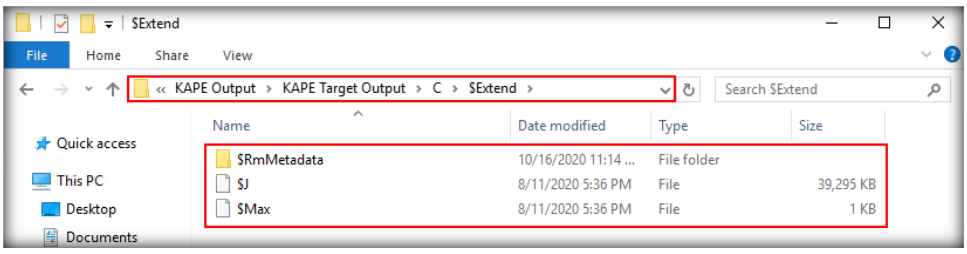


FIGURE 8.20: View contents of the $Extend folder

Note: The change journal file in **NTFS filesystem** named **$UsnJrnl** is stored in a hidden folder named **$Extend** and contains two alternate data streams named **$J** and **$Max**. Copying the **$J** file via the **KAPE** will also copy the **$Max** file, along with its full path.

33. Now, navigate to **$RmMetadata → $TxfLog**. You will see the **$T** file there.

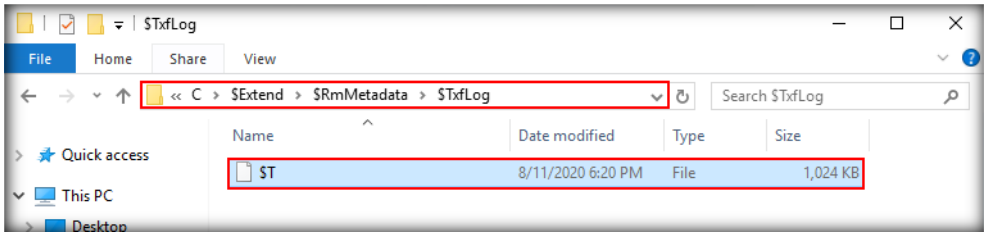


FIGURE 8.21: View contents of the $TxfLog folder

34. Return to the **C** folder inside the **KAPE Target Output** folder. Now, navigate to **users\Administrator\AppData\Roaming\Microsoft**. You will see two folders there: **Office** and **Windows**.

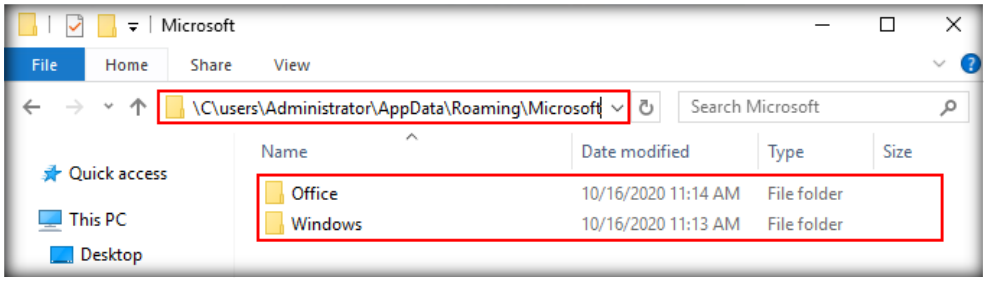


FIGURE 8.22: View contents of the Microsoft folder

Note: The **Jump List** files and **LNK** files generated in the **KAPE Target Output** folder might vary in your lab environment as these depend on the usage of files in the machine.

35. In the path **C:\KAPE Output\KAPE Target Output\C\users\Administrator\AppData\Roaming\Microsoft\Office\Recent**, you will see a list of **LNK files** that are copied from your machine.

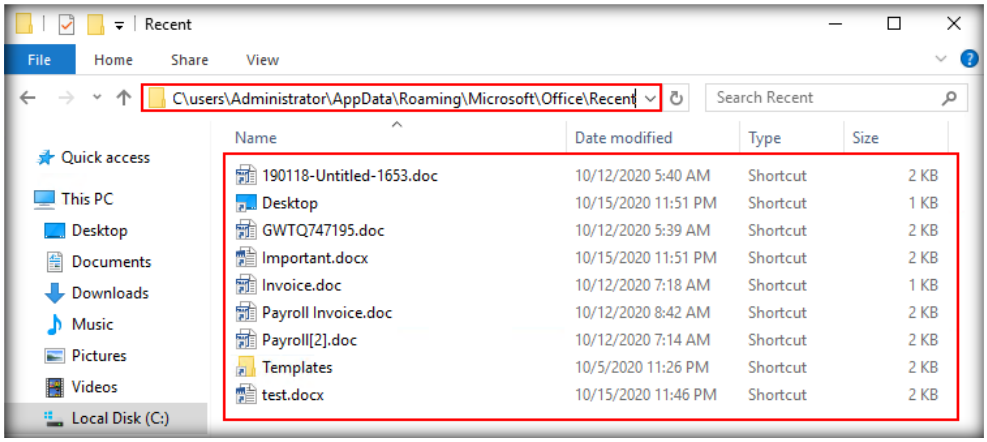


FIGURE 8.23: View the LNK files in the Microsoft/Office/Recent folder

36. Now, return to the **Microsoft** folder and navigate to **Windows\Recent**. You will see two folders named **AutomaticDestinations** and **CustomDestinations** (**A**) that contain the list of all **Jump List** files, along with a list of **LNK files** (**B**).

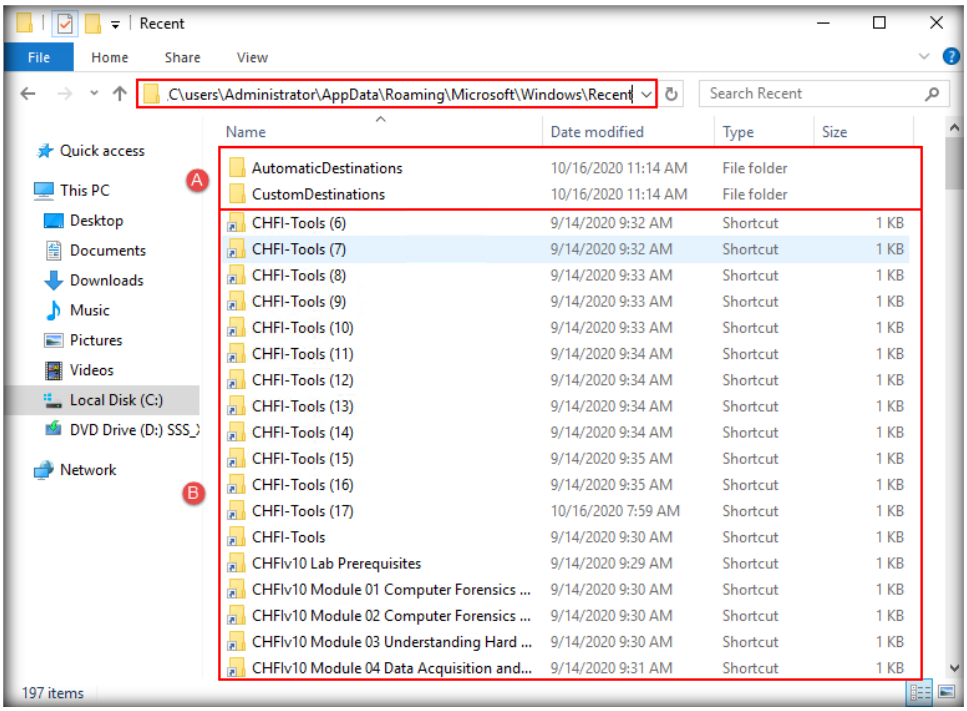


FIGURE 8.24: View contents of the Microsoft/Windows/Recent folder

37. Go to both **AutomaticDestinations** and **CustomDestinations** folders to view the lists of copied **Jump Lists** files, as shown in the following screenshots:

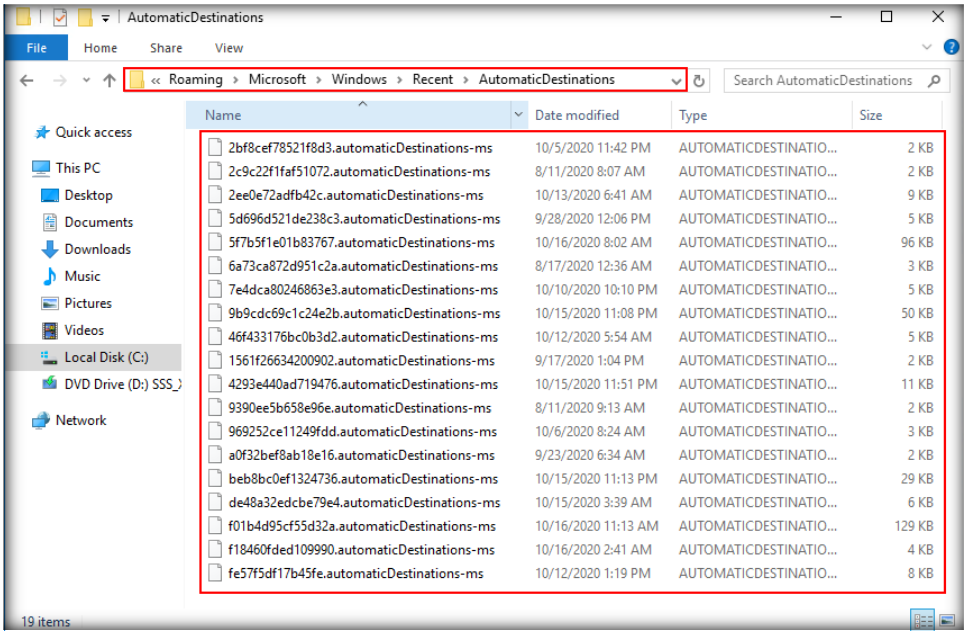


FIGURE 8.25: View the list of Jump Lists files in the AutomaticDestinations folder

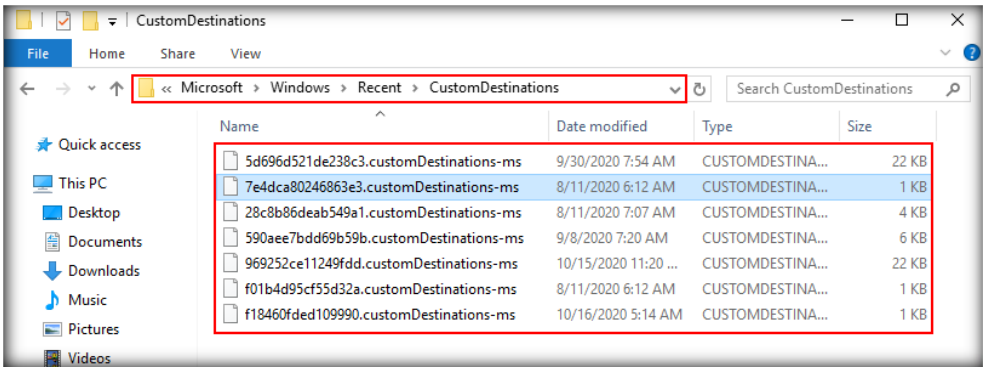


FIGURE 8.26: View the list of Jump Lists files in the CustomDestinations folder

Note: Depending on the usage, you may also locate **Target LNK files** in the **C:\KAPE Output\KAPE Target Output\C\users\Public\Desktop** folder or **C:\KAPE Output\KAPE Target Output\C\users\Administrator\Desktop** folder

TASK 4

Parse the Acquired Target Files in the Module Section in KAPE

38. As the chosen **Target files** are already acquired, we will now use the same application to **parse** these files and examine their contents. To do so, maximize the **KAPE GUI** window.

39. Ensure that you have unchecked **Use Target options** on the top-left corner of the tool. This will **disable** all the **Target options**. Now, click on **Use Module options** to enable it, as shown in the screenshot below:

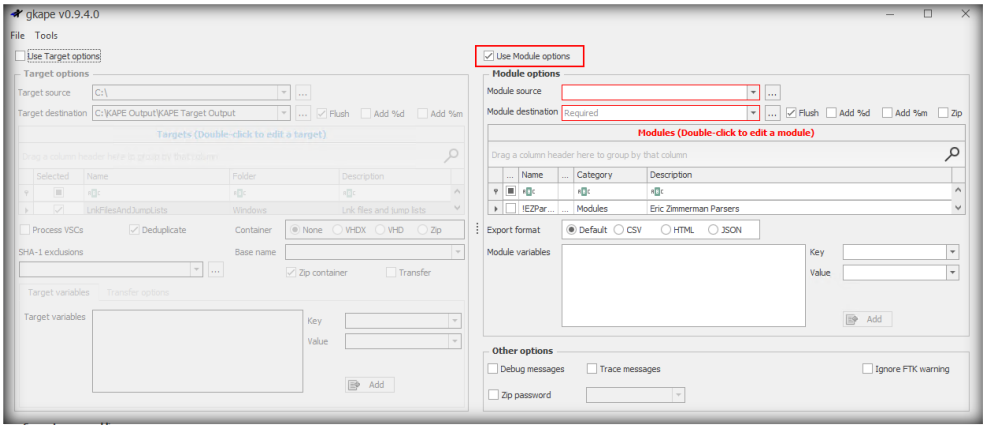


FIGURE 8.27: Unchecking Use Target options and enable Use Module options

40. Now, click on the **Ellipsis (...)** button beside **Module source** to select a source.

FIGURE 8.28: Click the Ellipsis button to add a Module source

41. The **Browse for Folder** window appears. Navigate to **Local Drive (C:)** → **KAPE Output** → **KAPE Target Output** and **select** it. This will enable you to parse the acquired **Target files**. Once configured, click **OK**.

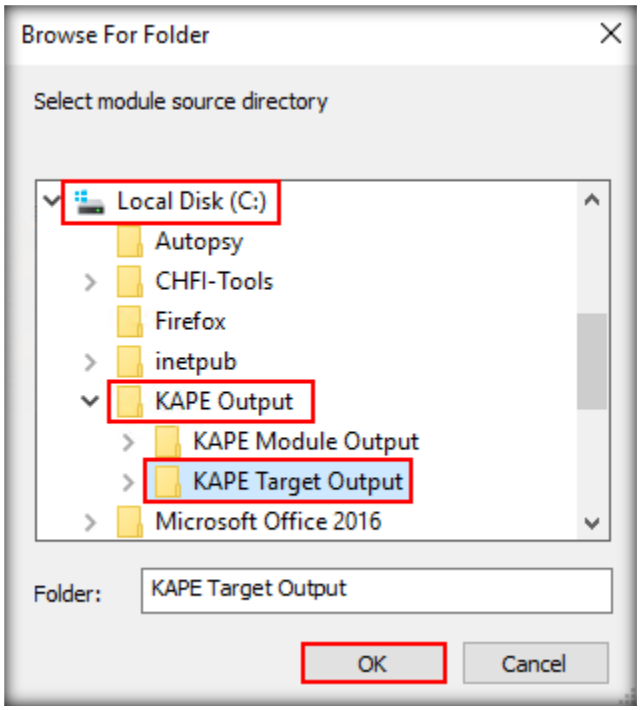


FIGURE 8.29: Select the KAPE Target Output folder as the Module Source

42. Now click on the **Ellipsis (...)** button beside **Module destination**

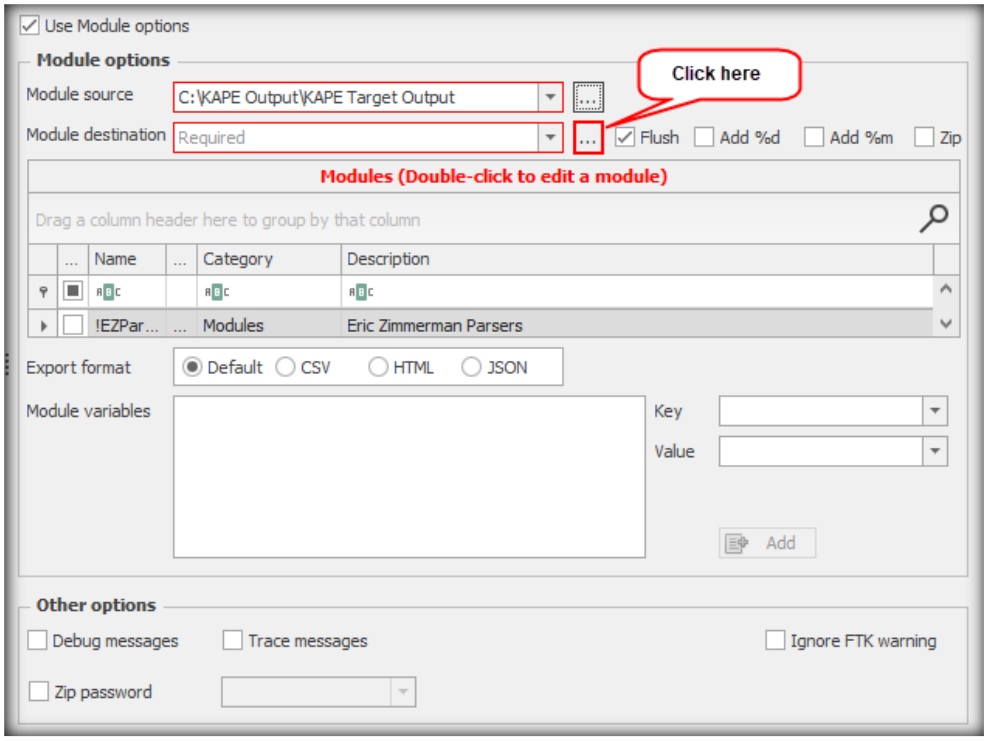


FIGURE 8.30: Click the Ellipsis button to add a Module Destination

43. On the **Browse for Folder** window that appears next, navigate to **Local Drive (C:) → KAPE Output → KAPE Module Output**, **select** it, and click **OK**

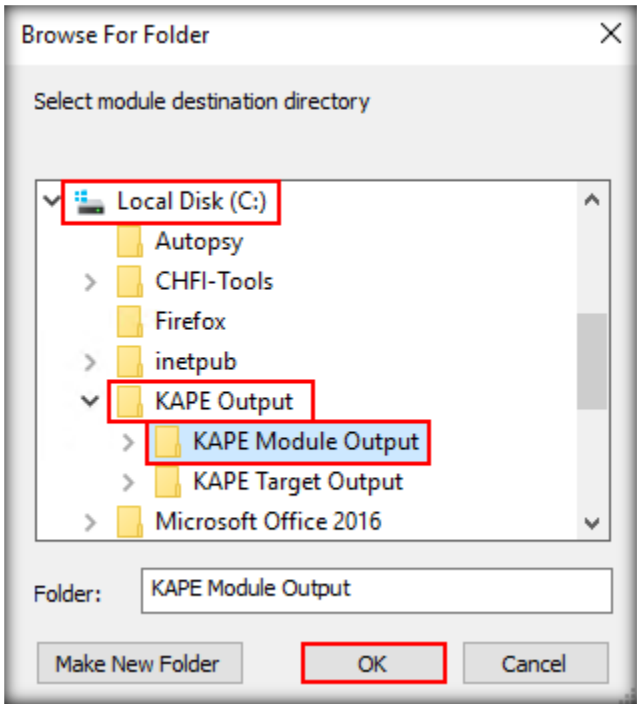


FIGURE 8.31: Select KAPE Module Output folder as the Module Destination

44. Now, we will select the **Modules** from the **Modules** section, using which we want to parse our **Target files**. In this lab, we will be demonstrating how to parse the **$MFT Target** file, along with the **LNK files** and **Jump Lists** acquired in the **KAPE Target Output** folder.

45. To parse the **$MFT Target** file, we will choose a **Module** named **MFTECmd_$MFT**. So, in the **Module** section, place the cursor below the **Name** section beside the letters **RBC** and type **MFTECmd_$MFT**. When the **Module** with the same name appears, **select** the checkbox associated with the module, as shown in the screenshot below:

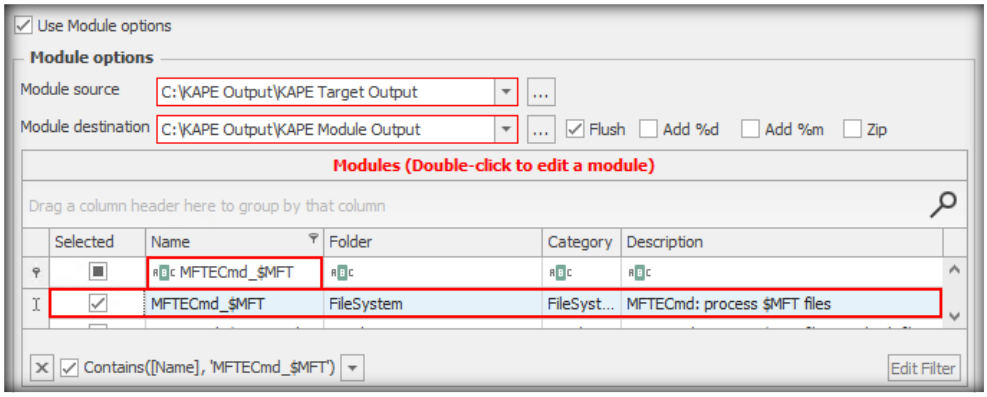


FIGURE 8.32: Select MFTECmd_$MFT as one of the Modules

46. Now **double-click** on the **MFTECmd_$MFT** module to determine what **executable** will be running via the **Module** to parse the **$MFT Target** file.

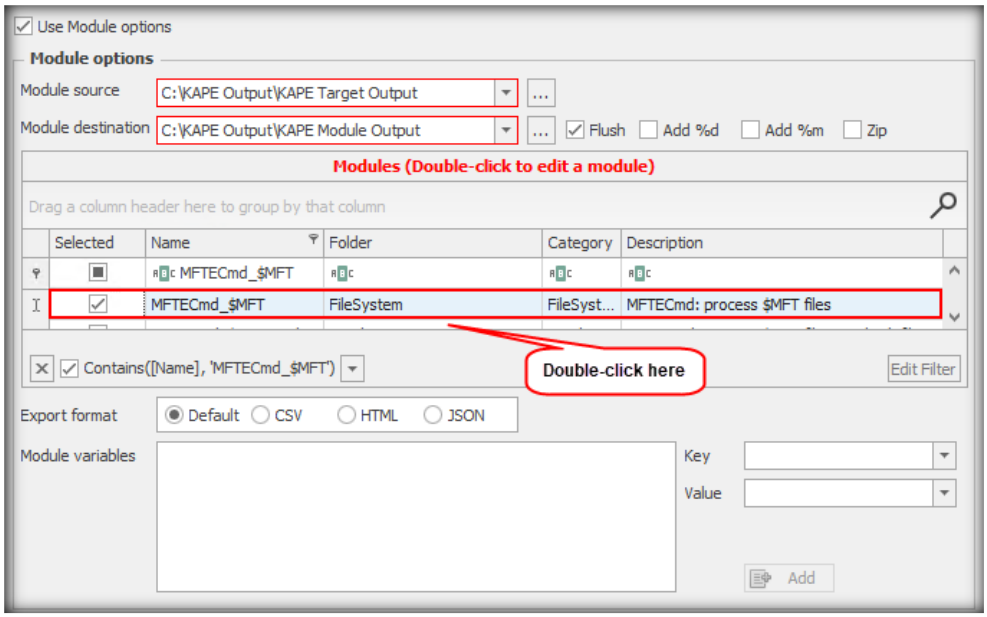


FIGURE 8.33: Double-click the MFTECmd_$MFT Module to view its contents

47. An **Editor: MFTECmd_$MFT** window opens. It shows that the selected **Module** will use the **MFTECmd.exe** file to parse the **$MFT** file, along with the **CommandLine** and **ExportFormat**.

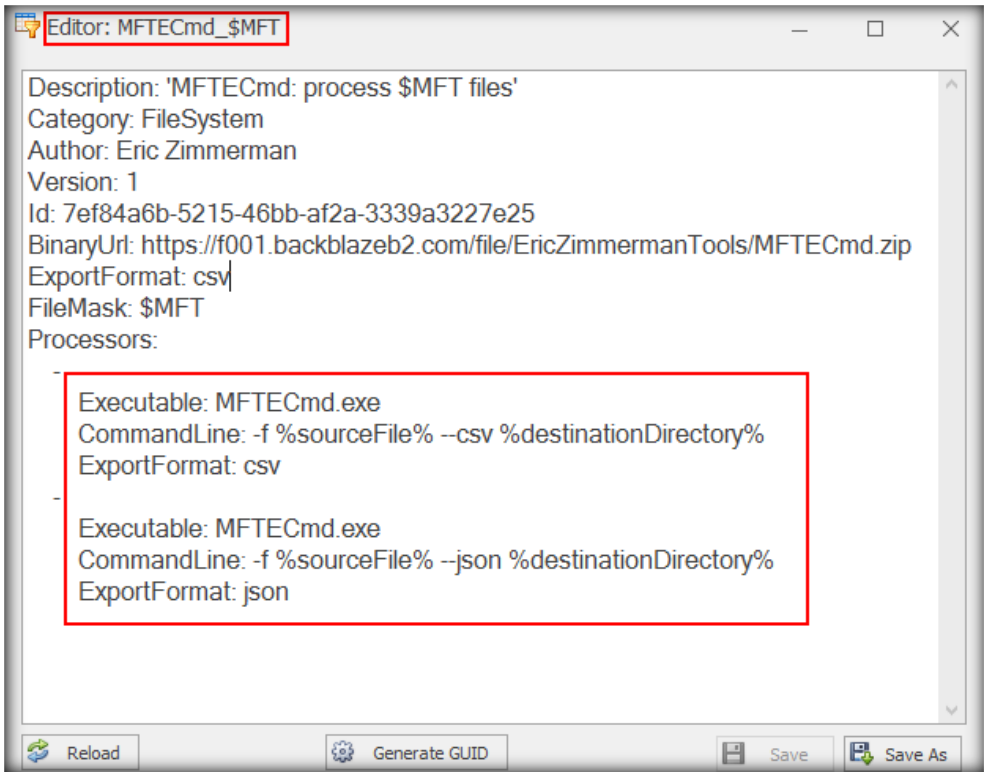


FIGURE 8.34: View information on the Editor: MFTECmd_$MFT window

48. Close the **MFTECmd_$MFT** window. Now, we will select a **Module** to parse the **Jump Lists** stored in the **KAPE Target Output** folder. For the **Jump Lists**, type **JLECmd**. A **Module entry** with the same name will appear. Select the checkbox associated with the module and then **double-click** it to view the name of the **executable** contained in that **Module**.

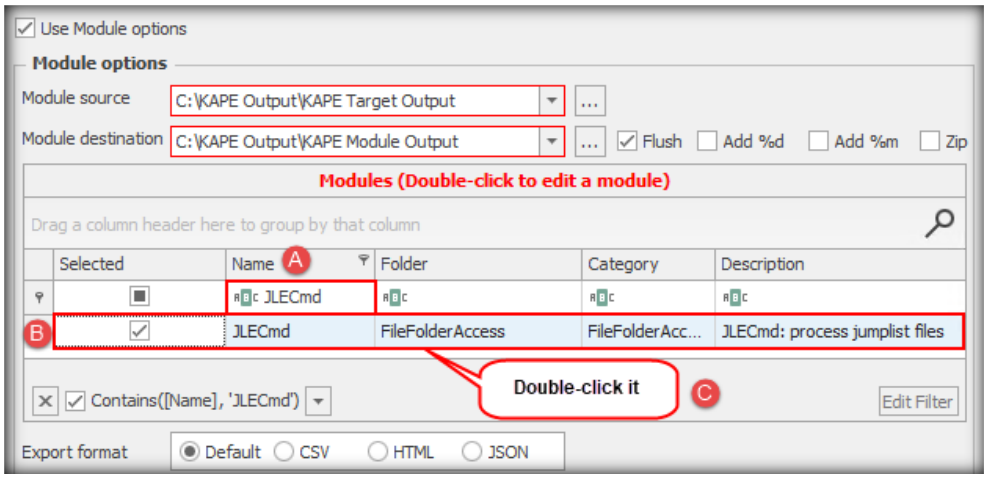


FIGURE 8.35: Selecting JLECmd as one of the Modules

49. An **Editor: JLECmd** window opens. It shows that the **executable name** to be used for parsing the **Jump List** files is **JLECmd.exe**, along with the **CommandLine** and **ExportFormat**.

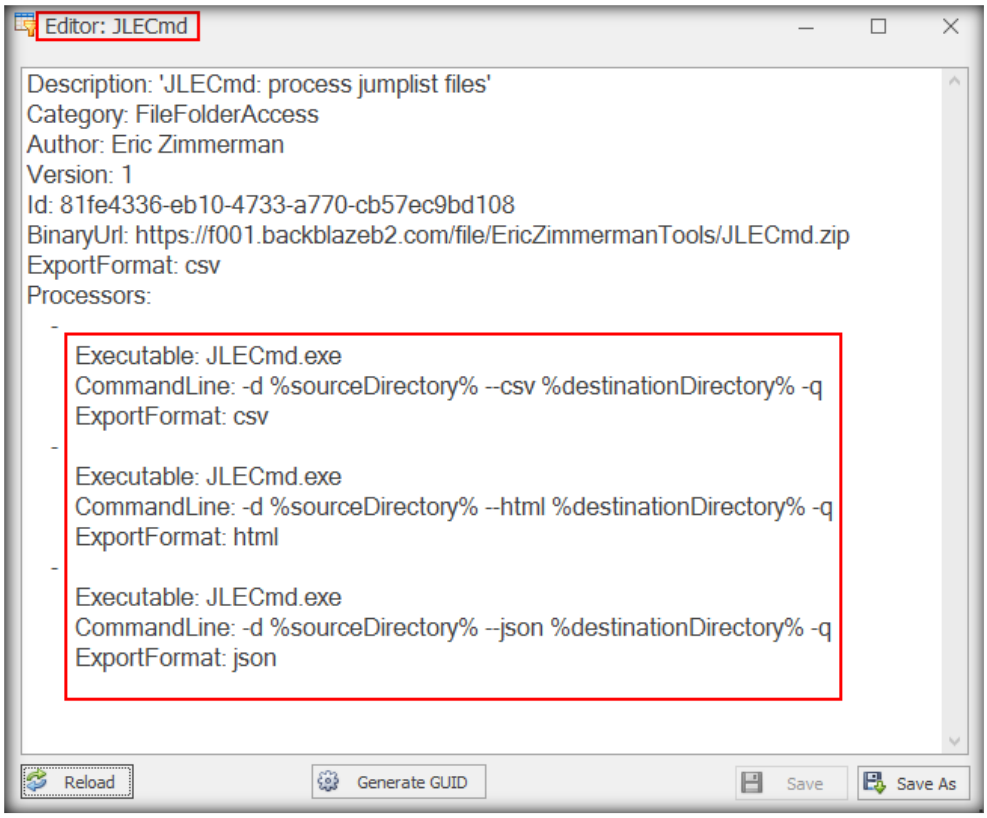


FIGURE 8.36: View information on the Editor: JLECmd window

50. Close the **Editor: JLECmd** window. Type **LECmd** and select the checkbox for the **Module entry** with the same name when it appears. Now, **double-click** on the entry. This **Module** will be used for parsing the **LNK files** in the **KAPE Target Output** folder.

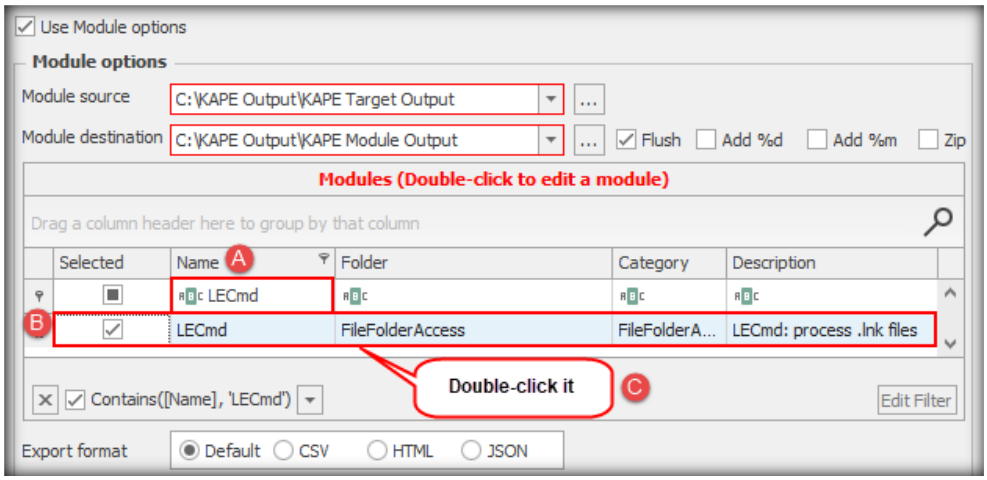


FIGURE 8.37: Select LECmd as one of the Modules

51. The **Editor: LECmd** window that appears shows the executable name as **LECmd.exe**, along with the **CommandLine** and **ExportFormat**

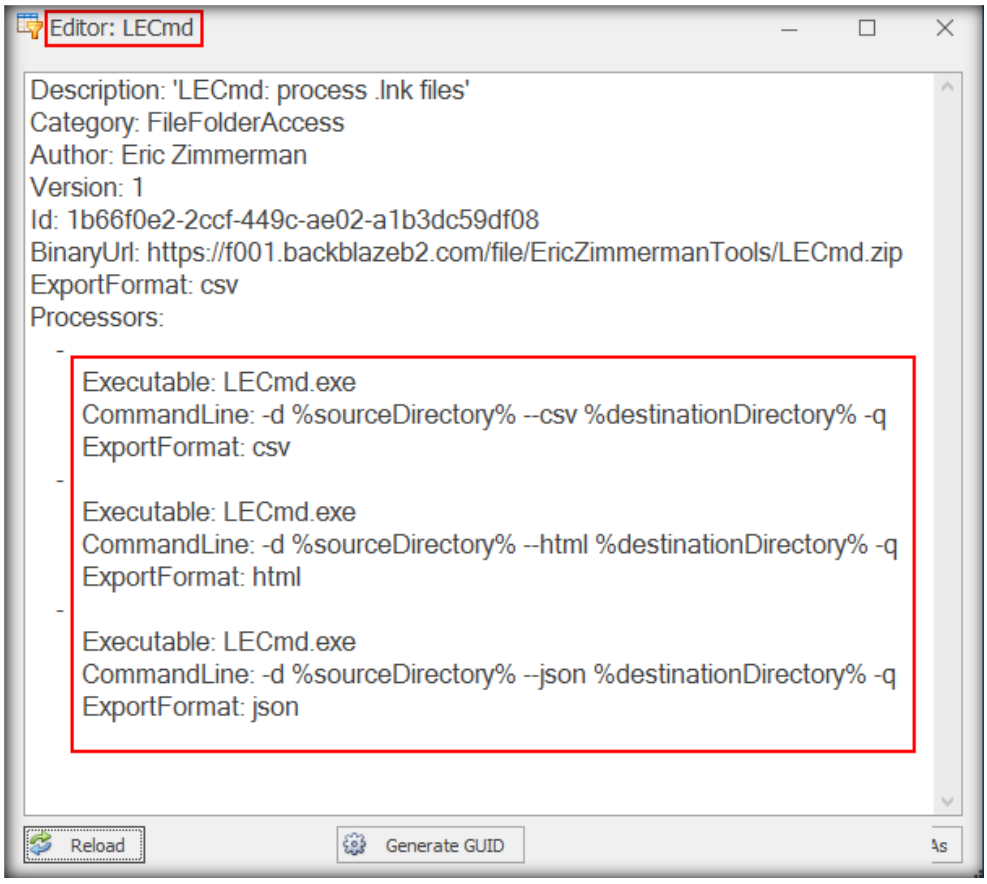


FIGURE 8.38: View information on the Editor: LECmd window

52. Close the **Editor: LECmd** window. The required **Modules** are all selected. Ensure that the **Flush** option is checked and choose **CSV** as the **Export format** as shown in the screenshot below:

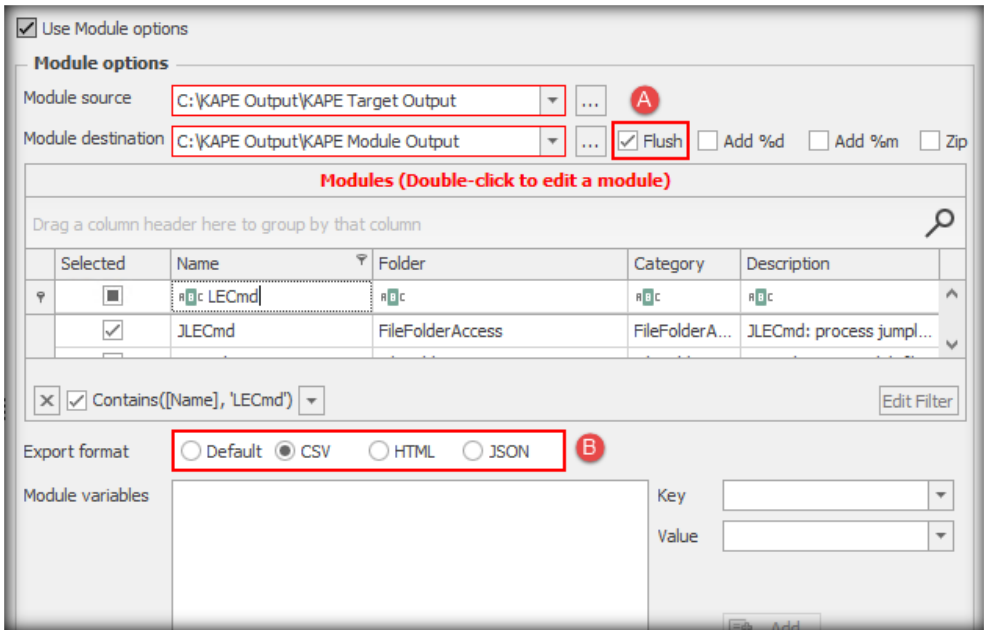


FIGURE 8.39: Keep the Flush option checked and select CSV as the Export format

53. With all configurations done in the **Modules** section, the **Current command line** will look like the following:

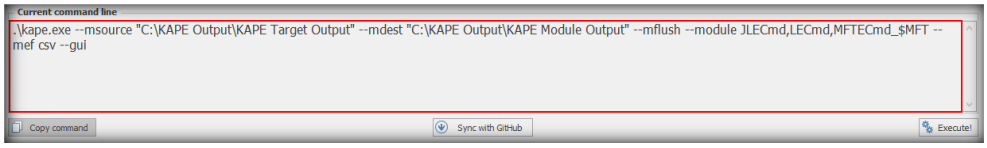


FIGURE 8.40: Current command line after selecting all Modules

54. In the highlighted section of **Current command line** as shown above, the **--msource** parameter reflects the path of the selected **Module** source folder, the **--mdest** parameter displays the path of the selected **Module destination** folder, **--mflush** shows that the contents of the selected **Module destination** folder will be flushed prior to parsing, the **--module** parameter shows the **Modules** that are selected, the **--mef** parameter shows the Export format chosen, and **--gui** shows that the command line will be executed via the **KAPE GUI**.

55. Once configured and reviewed, click on the **Execute!** button.

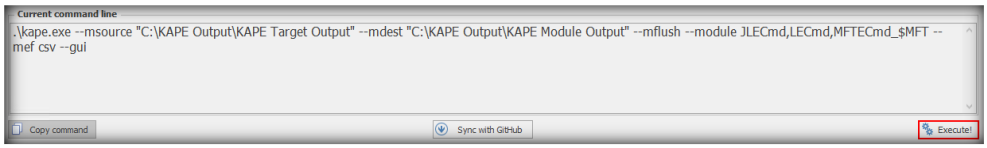


FIGURE 8.41: Click the Execute! button to parse the selected Target files

Note: If a **DATA DESTRUCTION WARNING!** window appears, click **OK** to continue.

56. The tool will now open a command-line window showing the progress of the **Modules** being run against the **Target** files stored in the **KAPE Target Output** folder. When the parsing is done, the tool will prompt you to **press any key to exit** the command-line window.

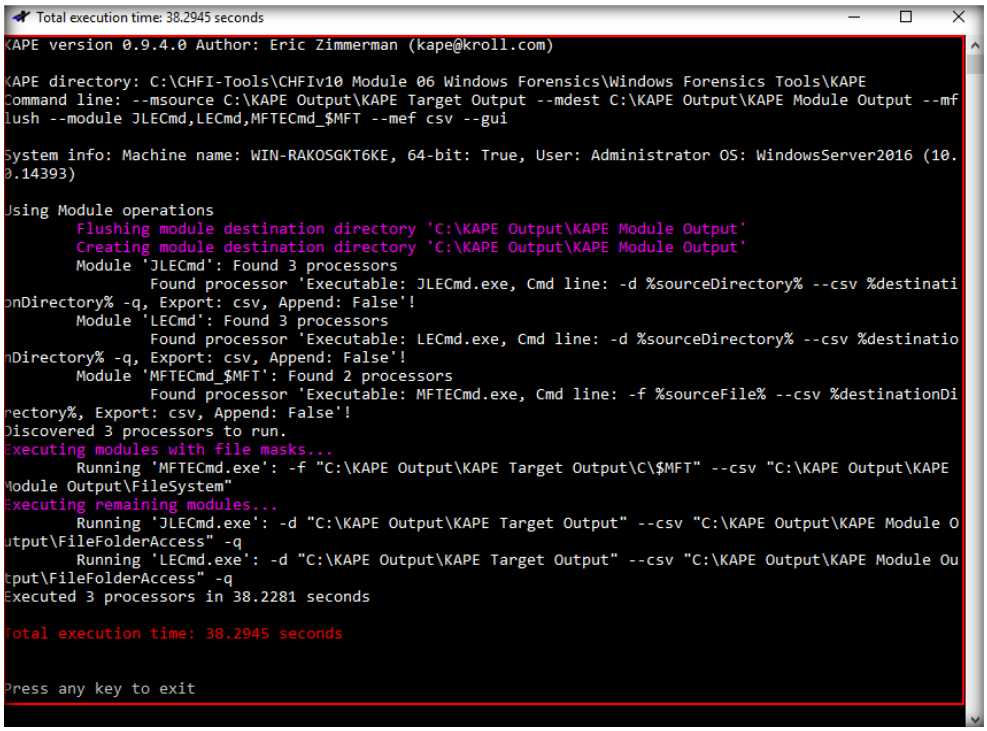


FIGURE 8.42: Command-line window generated by KAPE

Note: The tool takes some time to parse the **Target** files.

57. Press any **key** and hit **Enter** to exit the command-line window. Minimize the **KAPE GUI** window.

TASK 5

Examine the Parsed Files Stored in the Module Destination Folder

58. Now, open **File Explorer** and navigate to **C:\KAPE Output\KAPE Module Output**. You will see two folders: **FileFolderAccess** and **FileSystem**, along with one text file named **2020-10-16T182016_ConsoleLog.txt**, as shown in the screenshot below:

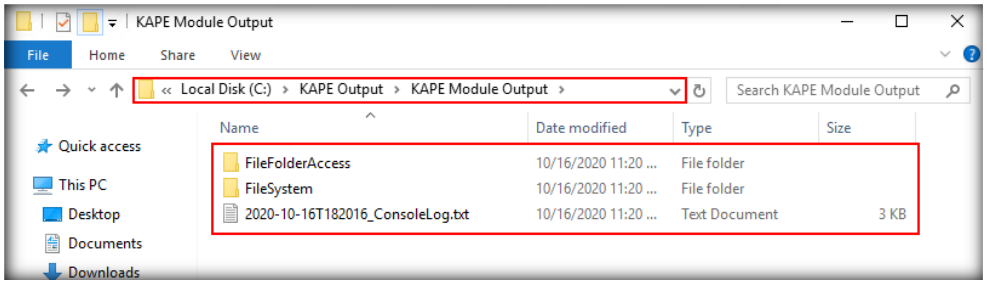


FIGURE 8.43: View contents of the KAPE Module Output folder

Note: The name of the file shown above may be different in your lab environment.

59. Double-click the **FileSystem** folder and you will see a file named **20201016182035_MFTECmd_$MFT_Output.csv**. This is the **CSV** file that was generated after parsing the **$MFT Target** file.

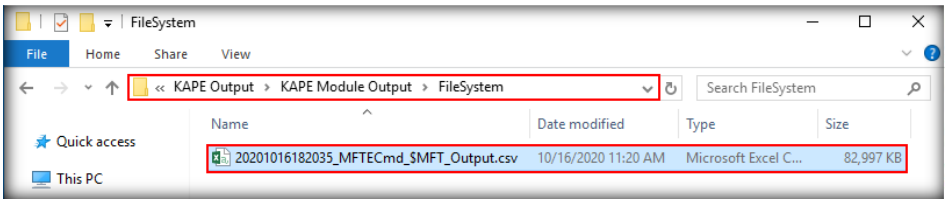


FIGURE 8.44: Viewing the 20201016182035_MFTECmd_$MFT_Output.csv file

Note: The first 8 numbers in the filename highlighted above shows the date on which it was parsed in the **yyyymmdd** format. These numbers may differ in your lab environment.

Note: You might need to sign in and activate **Microsoft Office** to in order to perform all the subsequent steps in this lab that involve **Microsoft Office Excel** application.

60. Double-click on the **CSV** file to open and view its contents.

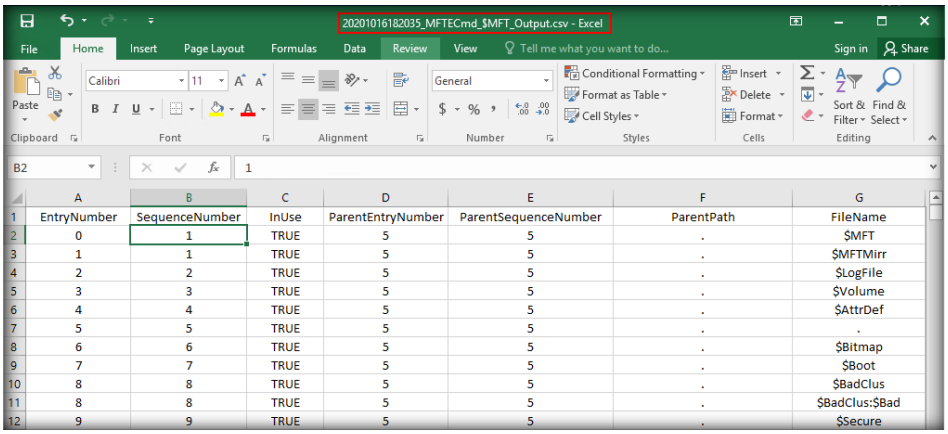


FIGURE 8.45: View the contents of 20201016182035_MFTECmd_$MFT_Output.csv file

Note: You may need to adjust the **size** and **width** of **rows** and **columns** for a better view.

61. Select the entire top row, go to the **Views** tab, click on **Freeze Panes**, and select **Freeze Top Row** from the drop-down menu.

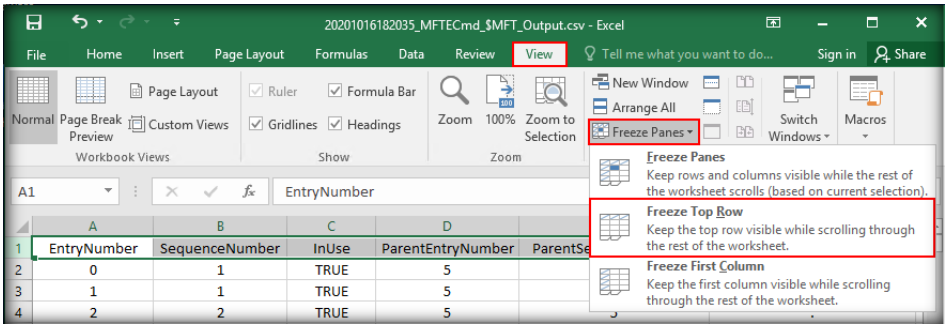
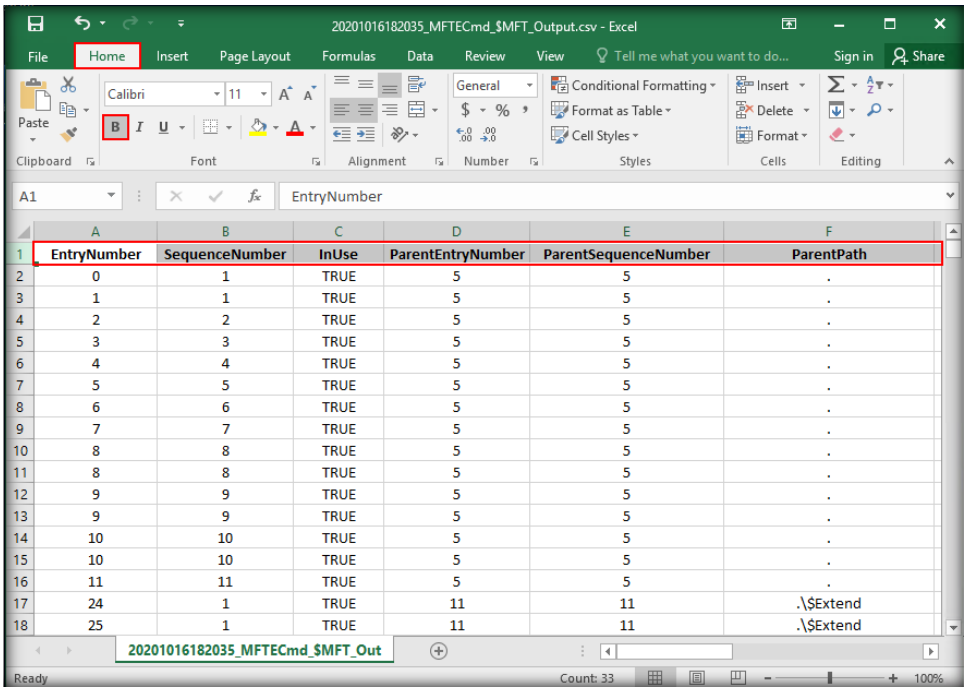


FIGURE 8.46: Go to Views → Freeze Panes→ Freeze Top Row

62. Ensure that the top row is selected. Now, return to the **Home** Tab. Click on the **B** icon to view the top row in **bold**.

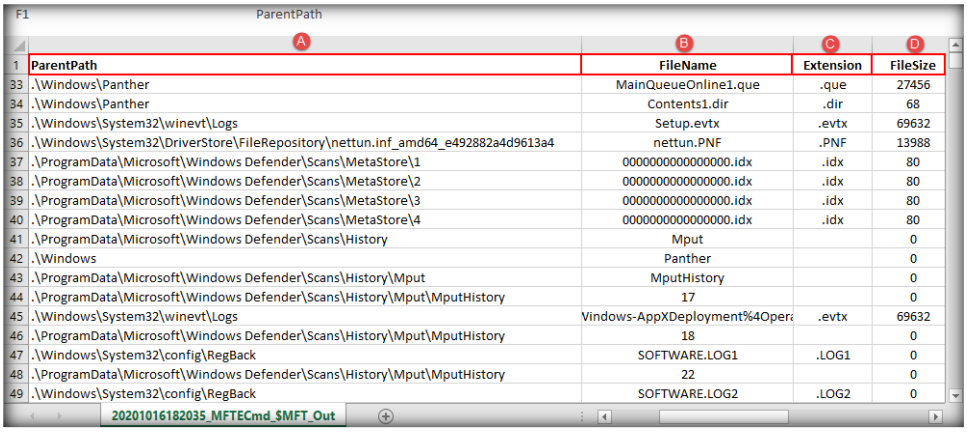


	A	B	C	D	E	F
1	**EntryNumber**	**SequenceNumber**	**InUse**	**ParentEntryNumber**	**ParentSequenceNumber**	**ParentPath**
2	0	1	TRUE	5	5	.
3	1	1	TRUE	5	5	.
4	2	2	TRUE	5	5	.
5	3	3	TRUE	5	5	.
6	4	4	TRUE	5	5	.
7	5	5	TRUE	5	5	.
8	6	6	TRUE	5	5	.
9	7	7	TRUE	5	5	.
10	8	8	TRUE	5	5	.
11	8	8	TRUE	5	5	.
12	9	9	TRUE	5	5	.
13	9	9	TRUE	5	5	.
14	10	10	TRUE	5	5	.
15	10	10	TRUE	5	5	.
16	11	11	TRUE	5	5	.
17	24	1	TRUE	11	11	.\$Extend
18	25	1	TRUE	11	11	.\$Extend

FIGURE 8.47: Select the B icon on the Home tab

Note: We will be using this formatting for the rest of the .CSV files as well in this lab.

63. Scroll down to view all the files inside the **$MFT** files. Drag the horizontal scroll bar to the right to find many notable entries about the files reflected in the parsed **$MFT** file, such as their **ParentPath** (**A**) or their location in the NTFS file system, **FileName** or name of the files (**B**), their **Extension** (**C**), and **FileSize** (**D**).

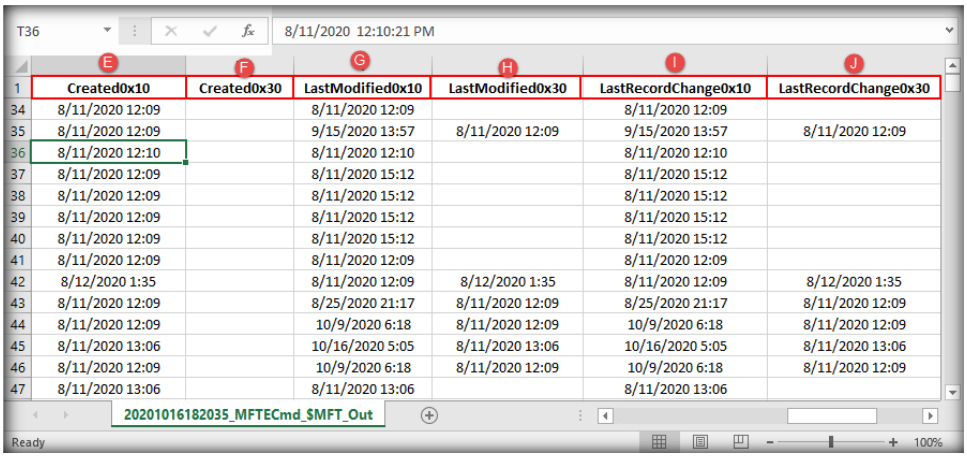


	A	B	C	D
1	**ParentPath**	**FileName**	**Extension**	**FileSize**
33	.\Windows\Panther	MainQueueOnline1.que	.que	27456
34	.\Windows\Panther	Contents1.dir	.dir	68
35	.\Windows\System32\winevt\Logs	Setup.evtx	.evtx	69632
36	.\Windows\System32\DriverStore\FileRepository\nettun.inf_amd64_e492882a4d9613a4	nettun.PNF	.PNF	13988
37	.\ProgramData\Microsoft\Windows Defender\Scans\MetaStore\1	0000000000000000.idx	.idx	80
38	.\ProgramData\Microsoft\Windows Defender\Scans\MetaStore\2	0000000000000000.idx	.idx	80
39	.\ProgramData\Microsoft\Windows Defender\Scans\MetaStore\3	0000000000000000.idx	.idx	80
40	.\ProgramData\Microsoft\Windows Defender\Scans\MetaStore\4	0000000000000000.idx	.idx	80
41	.\ProgramData\Microsoft\Windows Defender\Scans\History	Mput		0
42	.\Windows	Panther		0
43	.\ProgramData\Microsoft\Windows Defender\Scans\History\Mput	MputHistory		0
44	.\ProgramData\Microsoft\Windows Defender\Scans\History\Mput\MputHistory	17		0
45	.\Windows\System32\winevt\Logs	Vindows-AppXDeployment%4Opera	.evtx	69632
46	.\ProgramData\Microsoft\Windows Defender\Scans\History\Mput\MputHistory	18		0
47	.\Windows\System32\config\RegBack	SOFTWARE.LOG1	.LOG1	0
48	.\ProgramData\Microsoft\Windows Defender\Scans\History\Mput\MputHistory	22		0
49	.\Windows\System32\config\RegBack	SOFTWARE.LOG2	.LOG2	0

FIGURE 8.48: Analyzing the parsed contents of the Target file $MFT

64. On dragging the horizontal scroll bar further to the right, you will observe **Created0x10** (**E**) and **Created0x30** (**F**), which represent the creation date and time of any file; **LastModified0x10** (**G**) and **LastModified0x30** (**H**),

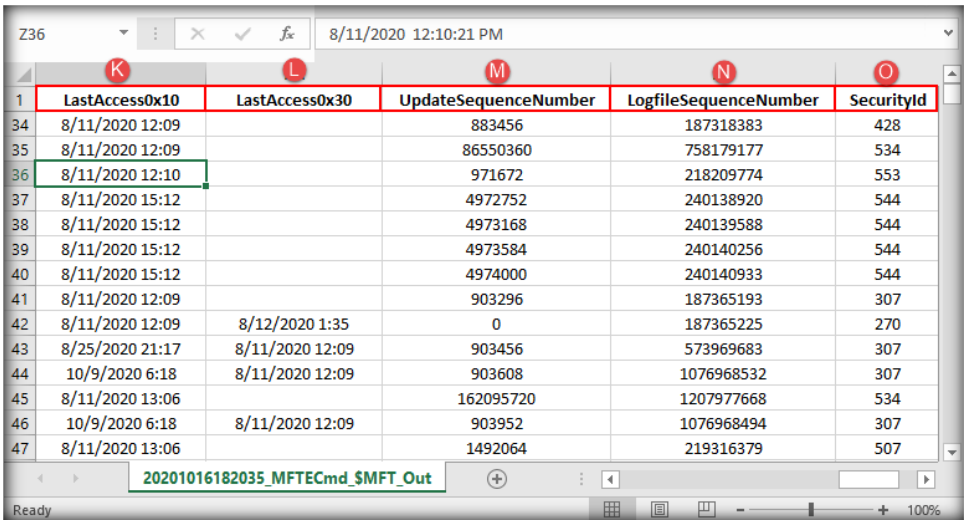
which represent the last modified date and time of any file; and **LastRecordChange0x10** (**I**) and **LastRecordChange0x30** (**J**), which represent the date and time when any change was recorded in any file.



	E	F	G	H	I	J
1	Created0x10	Created0x30	LastModified0x10	LastModified0x30	LastRecordChange0x10	LastRecordChange0x30
34	8/11/2020 12:09		8/11/2020 12:09		8/11/2020 12:09	
35	8/11/2020 12:09		9/15/2020 13:57	8/11/2020 12:09	9/15/2020 13:57	8/11/2020 12:09
36	8/11/2020 12:10		8/11/2020 12:10		8/11/2020 12:10	
37	8/11/2020 12:09		8/11/2020 15:12		8/11/2020 15:12	
38	8/11/2020 12:09		8/11/2020 15:12		8/11/2020 15:12	
39	8/11/2020 12:09		8/11/2020 15:12		8/11/2020 15:12	
40	8/11/2020 12:09		8/11/2020 15:12		8/11/2020 15:12	
41	8/11/2020 12:09		8/11/2020 12:09		8/11/2020 12:09	
42	8/12/2020 1:35		8/11/2020 12:09	8/12/2020 1:35	8/11/2020 12:09	8/12/2020 1:35
43	8/11/2020 12:09		8/25/2020 21:17	8/11/2020 12:09	8/25/2020 21:17	8/11/2020 12:09
44	8/11/2020 12:09		10/9/2020 6:18	8/11/2020 12:09	10/9/2020 6:18	8/11/2020 12:09
45	8/11/2020 13:06		10/16/2020 5:05	8/11/2020 13:06	10/16/2020 5:05	8/11/2020 13:06
46	8/11/2020 12:09		10/9/2020 6:18	8/11/2020 12:09	10/9/2020 6:18	8/11/2020 12:09
47	8/11/2020 13:06		8/11/2020 13:06		8/11/2020 13:06	

FIGURE 8.49: Analyzing the parsed contents of the Target file $MFT

65. Upon dragging the horizontal slider further to the right, you can observe other entries such as **LastAccess0x10** (**K**) and **LastAccess0x30** (**L**), which represent the last access date and time of any file, along with the **UpdateSequenceNumber** (**M**), **LogFileSequenceNumber** (**N**), and **SecurityId** (**O**)



	K	L	M	N	O
1	LastAccess0x10	LastAccess0x30	UpdateSequenceNumber	LogfileSequenceNumber	SecurityId
34	8/11/2020 12:09		883456	187318383	428
35	8/11/2020 12:09		86550360	758179177	534
36	8/11/2020 12:10		971672	218209774	553
37	8/11/2020 15:12		4972752	240138920	544
38	8/11/2020 15:12		4973168	240139588	544
39	8/11/2020 15:12		4973584	240140256	544
40	8/11/2020 15:12		4974000	240140933	544
41	8/11/2020 12:09		903296	187365193	307
42	8/11/2020 12:09	8/12/2020 1:35	0	187365225	270
43	8/25/2020 21:17	8/11/2020 12:09	903456	573969683	307
44	10/9/2020 6:18	8/11/2020 12:09	903608	1076968532	307
45	8/11/2020 13:06		162095720	1207977668	534
46	10/9/2020 6:18	8/11/2020 12:09	903952	1076968494	307
47	8/11/2020 13:06		1492064	219316379	507

FIGURE 8.50: Analyzing the parsed contents of the Target file $MFT

Note: The **date** and **time** format in the screenshots above might differ as per the Excel sheet formatting on your machine.

66. These entries are very important from a forensic point of view as you can identify and examine the **last access** and **modified** time of any specific file in the **NTFS** file system of the suspect machine. The **$MFT** file also holds information about files that are **deleted**, and its examination can help in the **recovery** of such files.

67. Upon examining the entries, close the **20201016182035_MFTECmd_$MFT_Output.csv** file.

Note: If you receive a message from **Microsoft Excel** stating whether you wish to save the changes, select **Don't Save**.

68. On **File Explorer**, navigate to **C:\KAPE Output\KAPE Module Output\FileFolderAccess** and you will see three .csv files: **20201016112054_AutomaticDestinations.csv** and **20201016112055_CustomDestinations.csv**, which contain information about parsed **Jump Lists**, and **20201016112056_LECmd_Output.csv**, which contains details about parsed **LNK files** from the **KAPE Target Output** folder

Note: The numbers in these file names include the date on which they were parsed. Hence, these numbers might differ in your lab environment.

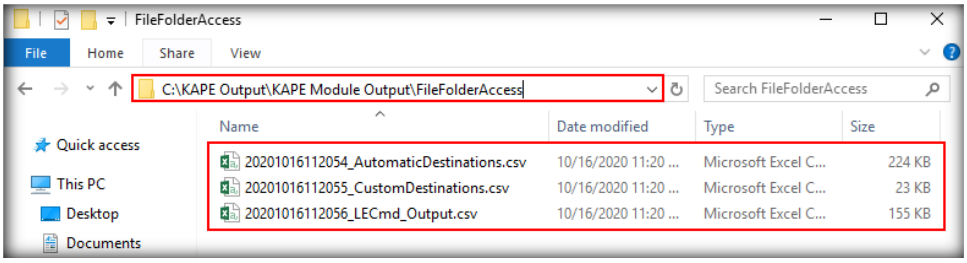


FIGURE 8.51: View the contents of the FileFolderAccess folder in KAPE Module Output

69. We will first examine the **Jump List** files in **20201016112054_AutomaticDestinations.csv** file. Double-click the file to open and view its contents.

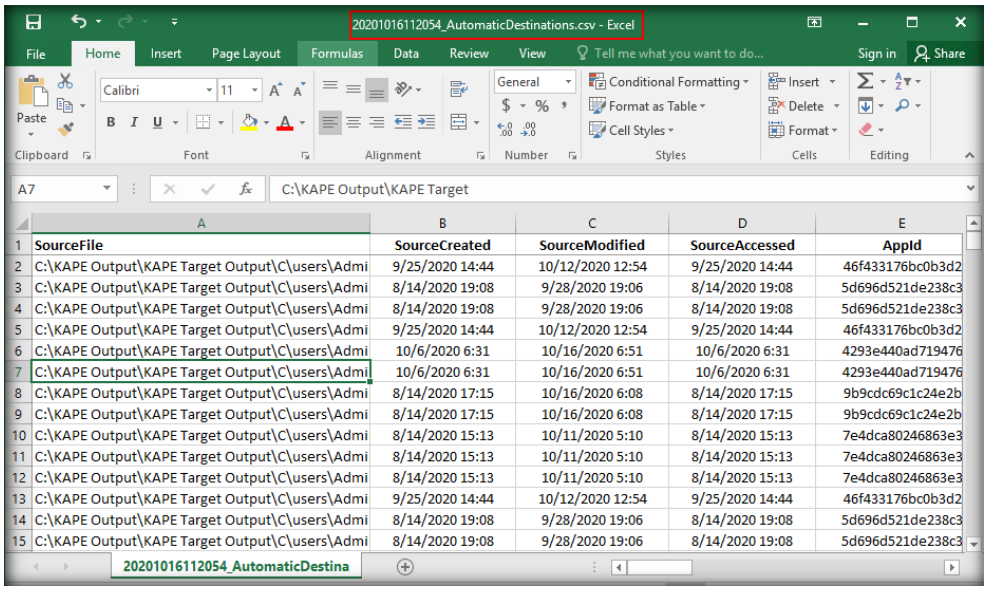
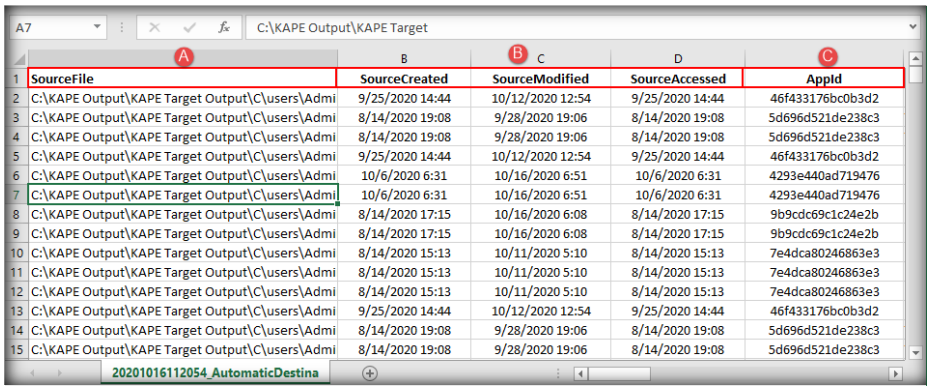


FIGURE 8.52: Viewing parsed Jump Lists in AutomaticDestinations

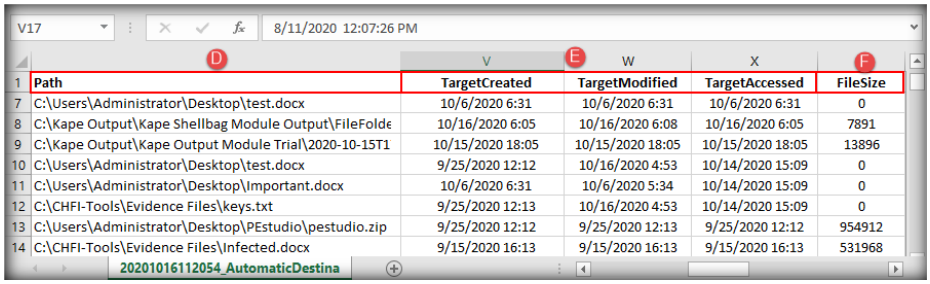
70. Some of the notable entries found in the **20201016112054_AutomaticDestinations.csv** file are **SourceFile** or the path of the **Module source** (**A**), **SourceCreated**, **SourceModified**,

SourceAccessed (**B**) **APPId** information (**C**), path of the target **Jump Lists** (**D**), **TargetCreated**, **TargetModified** and **TargetAccessed** date and time (**E**), and **FileSize** (**F**), as shown in the screenshots below:



	A	B	C	D	E
1	SourceFile	SourceCreated	SourceModified	SourceAccessed	AppId
2	C:\KAPE Output\KAPE Target Output\C\users\Admi	9/25/2020 14:44	10/12/2020 12:54	9/25/2020 14:44	46f433176bc0b3d2
3	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 19:08	9/28/2020 19:06	8/14/2020 19:08	5d696d521de238c3
4	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 19:08	9/28/2020 19:06	8/14/2020 19:08	5d696d521de238c3
5	C:\KAPE Output\KAPE Target Output\C\users\Admi	9/25/2020 14:44	10/12/2020 12:54	9/25/2020 14:44	46f433176bc0b3d2
6	C:\KAPE Output\KAPE Target Output\C\users\Admi	10/6/2020 6:31	10/16/2020 6:51	10/6/2020 6:31	4293e440ad719476
7	C:\KAPE Output\KAPE Target Output\C\users\Admi	10/6/2020 6:31	10/16/2020 6:51	10/6/2020 6:31	4293e440ad719476
8	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 17:15	10/16/2020 6:08	8/14/2020 17:15	9b9cdc69c1c24e2b
9	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 17:15	10/16/2020 6:08	8/14/2020 17:15	9b9cdc69c1c24e2b
10	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 15:13	10/11/2020 5:10	8/14/2020 15:13	7e4dca80246863e3
11	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 15:13	10/11/2020 5:10	8/14/2020 15:13	7e4dca80246863e3
12	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 15:13	10/11/2020 5:10	8/14/2020 15:13	7e4dca80246863e3
13	C:\KAPE Output\KAPE Target Output\C\users\Admi	9/25/2020 14:44	10/12/2020 12:54	9/25/2020 14:44	46f433176bc0b3d2
14	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 19:08	9/28/2020 19:06	8/14/2020 19:08	5d696d521de238c3
15	C:\KAPE Output\KAPE Target Output\C\users\Admi	8/14/2020 19:08	9/28/2020 19:06	8/14/2020 19:08	5d696d521de238c3

FIGURE 8.53: Analyzing parsed Jump List entries in AutomaticDestinations



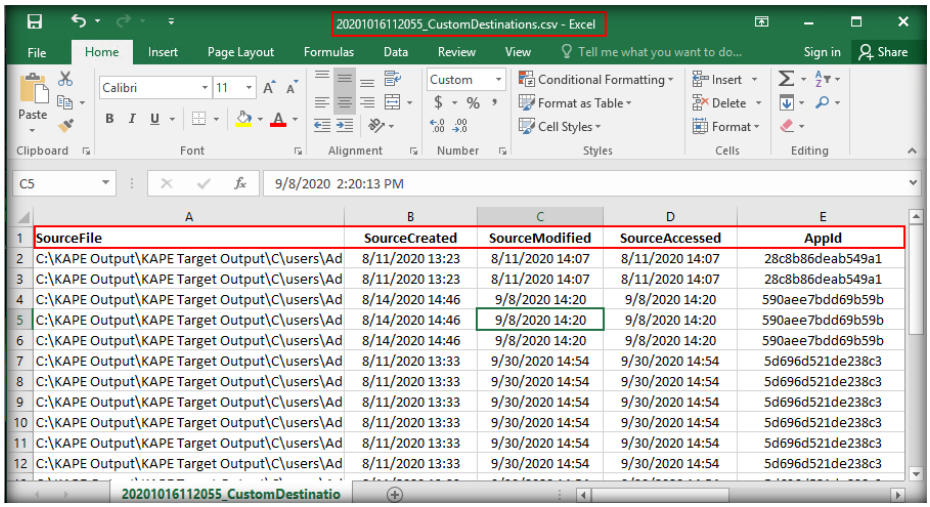
	D	V	W	X	F
1	Path	TargetCreated	TargetModified	TargetAccessed	FileSize
7	C:\Users\Administrator\Desktop\test.docx	10/6/2020 6:31	10/6/2020 6:31	10/6/2020 6:31	0
8	C:\Kape Output\Kape Shellbag Module Output\FileFolde	10/16/2020 6:05	10/16/2020 6:08	10/16/2020 6:05	7891
9	C:\Kape Output\Kape Output Module Trial\2020-10-15T1	10/15/2020 18:05	10/15/2020 18:05	10/15/2020 18:05	13896
10	C:\Users\Administrator\Desktop\test.docx	9/25/2020 12:12	10/16/2020 4:53	10/14/2020 15:09	0
11	C:\Users\Administrator\Desktop\Important.docx	10/6/2020 6:31	10/6/2020 5:34	10/14/2020 15:09	0
12	C:\CHFI-Tools\Evidence Files\keys.txt	9/25/2020 12:13	10/16/2020 4:53	10/14/2020 15:09	0
13	C:\Users\Administrator\Desktop\PEstudio\pestudio.zip	9/25/2020 12:12	9/25/2020 12:13	9/25/2020 12:12	954912
14	C:\CHFI-Tools\Evidence Files\Infected.docx	9/15/2020 16:13	9/15/2020 16:13	9/15/2020 16:13	531968

FIGURE 8.54: Analyzing parsed Jump List entries in AutomaticDestinations

71. Upon completing the examination of the **Jump Lists** in **AutomaticDestinations**, close the .csv file.

Note: If you get a **Microsoft Excel** message asking whether you wish to save the changes made to the file, click **Don't Save**.

72. On opening the **20201016112055_CustomDestinations.csv** file, you will receive similar type of information as found in the **20201016112054_AutomaticDestinations.csv** file.



	A	B	C	D	E
1	SourceFile	SourceCreated	SourceModified	SourceAccessed	AppId
2	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/11/2020 13:23	8/11/2020 14:07	8/11/2020 14:07	28c8b86deab549a1
3	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/11/2020 13:23	8/11/2020 14:07	8/11/2020 14:07	28c8b86deab549a1
4	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/14/2020 14:46	9/8/2020 14:20	9/8/2020 14:20	590aee7bdd69b59b
5	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/14/2020 14:46	9/8/2020 14:20	9/8/2020 14:20	590aee7bdd69b59b
6	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/14/2020 14:46	9/8/2020 14:20	9/8/2020 14:20	590aee7bdd69b59b
7	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/11/2020 13:33	9/30/2020 14:54	9/30/2020 14:54	5d696d521de238c3
8	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/11/2020 13:33	9/30/2020 14:54	9/30/2020 14:54	5d696d521de238c3
9	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/11/2020 13:33	9/30/2020 14:54	9/30/2020 14:54	5d696d521de238c3
10	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/11/2020 13:33	9/30/2020 14:54	9/30/2020 14:54	5d696d521de238c3
11	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/11/2020 13:33	9/30/2020 14:54	9/30/2020 14:54	5d696d521de238c3
12	C:\KAPE Output\KAPE Target Output\C\users\Ad	8/11/2020 13:33	9/30/2020 14:54	9/30/2020 14:54	5d696d521de238c3

FIGURE 8.55: Analyzing parsed Jump List entries in CustomDestinations

73. Examine all the **Jump Lists** in the **20201016112055_CustomDestinations.csv** and close the file without saving.

74. Now, open the **20201016112056_LECmd_Output.csv** file. You will see notable entries such as **SourceFile** (**A**), **SourceCreated**, **SourceModified** and **SourceAccessed**–related **Timestamps** (**B**), **TargetCreated**, **TargetModified**, and **TargetAccessed**–related **Timestamps** (**C**), **FileSize** (**D**), **RelativePath** (**E**), **TargetIDAbsolutePath** (**F**), **TargetMFTEntryNumber** (**G**), **TargetMFTSequenceNumber** (**H**), **MachineID** (**I**), and **MachineMACAddress** (**J**), as shown in the screenshots below:

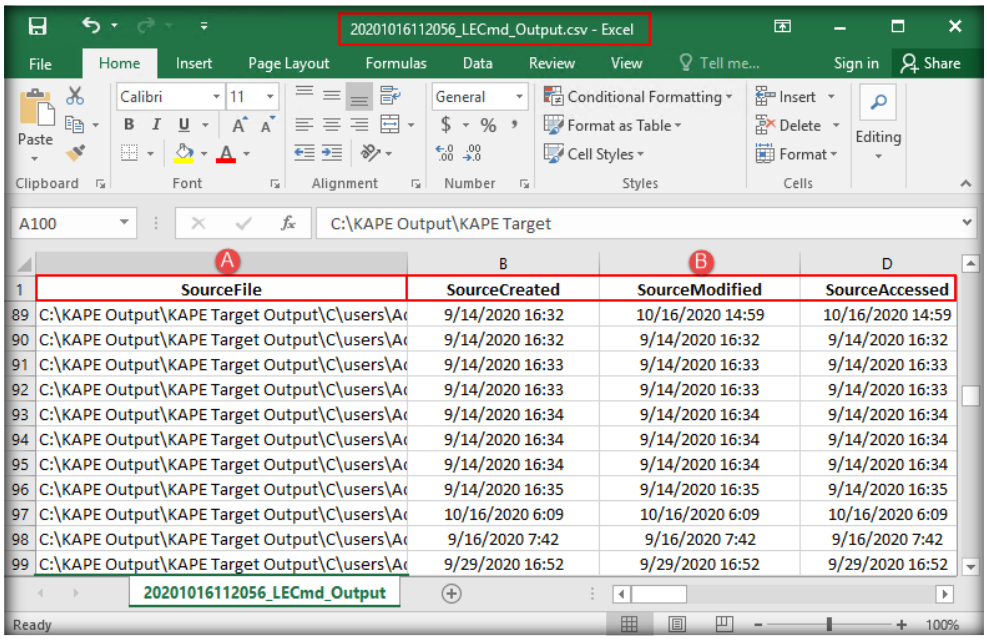


FIGURE 8.56: Analyzing LNK file entries

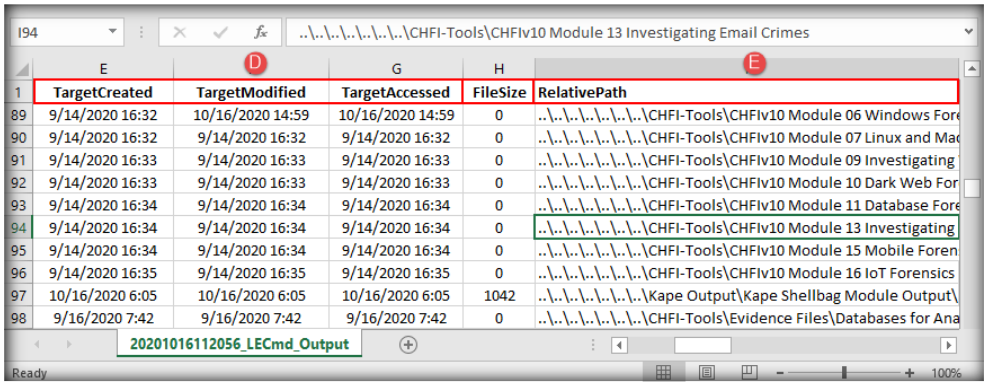
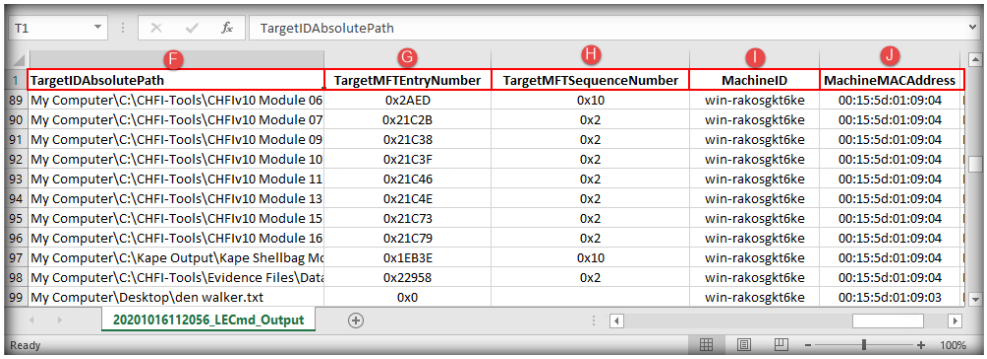


FIGURE 8.57: Analyzing LNK file entries



	F	G	H	I	J
1	TargetIDAbsolutePath	TargetMFTEntryNumber	TargetMFTSequenceNumber	MachineID	MachineMACAddress
89	My Computer\C:\CHFI-Tools\CHFIv10 Module 06	0x2AED	0x10	win-rakosgkt6ke	00:15:5d:01:09:04
90	My Computer\C:\CHFI-Tools\CHFIv10 Module 07	0x21C2B	0x2	win-rakosgkt6ke	00:15:5d:01:09:04
91	My Computer\C:\CHFI-Tools\CHFIv10 Module 09	0x21C38	0x2	win-rakosgkt6ke	00:15:5d:01:09:04
92	My Computer\C:\CHFI-Tools\CHFIv10 Module 10	0x21C3F	0x2	win-rakosgkt6ke	00:15:5d:01:09:04
93	My Computer\C:\CHFI-Tools\CHFIv10 Module 11	0x21C46	0x2	win-rakosgkt6ke	00:15:5d:01:09:04
94	My Computer\C:\CHFI-Tools\CHFIv10 Module 13	0x21C4E	0x2	win-rakosgkt6ke	00:15:5d:01:09:04
95	My Computer\C:\CHFI-Tools\CHFIv10 Module 15	0x21C73	0x2	win-rakosgkt6ke	00:15:5d:01:09:04
96	My Computer\C:\CHFI-Tools\CHFIv10 Module 16	0x21C79	0x2	win-rakosgkt6ke	00:15:5d:01:09:04
97	My Computer\C:\Kape Output\Kape Shellbag Mc	0x1EB3E	0x10	win-rakosgkt6ke	00:15:5d:01:09:04
98	My Computer\C:\CHFI-Tools\Evidence Files\Data	0x22958	0x2	win-rakosgkt6ke	00:15:5d:01:09:04
99	My Computer\Desktop\den walker.txt	0x0		win-rakosgkt6ke	00:15:5d:01:09:03

FIGURE 8.58: Analyzing LNK file entries

75. The examination of these entries about the **LNK files** and **Jump Lists** can provide useful information about the **user activities** on the system.
76. Thus, you can use the **KAPE tool** to acquire and process forensically valuable data and expedite the investigation.

Lab Analysis

Analyze the Target files and the parsed .csv files and document the results.

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

This page is intentionally left blank.