***LOVELY PROFESSIONAL UNIVERSITY***

***PHAGWARA, PUNJAB***

***A PROJECT REPORT***

***INT301***

***ACADEMIC TASK: 24 –****. Use any open-source software to generate report to acquire and analyse the volatile data that is temporarily stored in random access memory of your machine.*

***Submitted By:***

***Name of student:*** *Sonu Kumar Bharti*

***Registration Number:*** *11912136*

***Roll No:*** *21*

***Section:*** *KE015*

***Student Declaration***

***To whom so ever it may concern***

*I,* ***Sonu Kumar Bharti, 11912136****, hereby declare that the work done by me on* ***“Open-Source Software” on******08, April 2023****, is a record of original work for the Partial fulfillment of the requirements for the award of the degree,* ***B Tech (CSE)****.*

*Sonu Kumar Bharti (11912136)*

*Dated: 08 April 2023*

***CONTENT OF THE REPORT***

|  |  |  |
| --- | --- | --- |
| ***Sr.***  ***No.*** | ***TITLE*** | ***PAGE***  ***NO.*** |
| 1. | *Cover page* | 1 |
| 2. | *Student Declaration* | 2 |
| 3. | *Introduction* | 4 |
| 4. | *System Description* | 5 |
| 5. | *Analysis Report* | 6 |
| 6. | *Bibliography* | 11 |
| 7. | *Results and Discussion* | 11 |
| 8. | *Conclusion* | 12 |

***Introduction***

The system runtime status in the device's random-access memory (RAM) contains important information about running processes and programs. Extracting this information is essential for forensic analysis, troubleshooting, or performance tuning. The Volatility Framework is powerful tool that allows you to explore the health of your RAM and extract valuable information from it. This report uses the Volatile Framework to examine the system runtime state of the device's memory. It describes the basic concepts and techniques of memory forensics and provides an overview of the volatility framework. Next, we'll run a hands-on demonstration of using the Volatile Framework to extract key information from the device's RAM. This report covers a variety of topics including memory imaging, volatile plugins, memory analysis techniques, and how to interpret the extracted information. By the end of the report, readers will have a good understanding of how to explore the system runtime state of a device's RAM and use the Volatility Framework to extract valuable information.

* 1. ***Objective of the project:***

The goal of this project is to explore the system runtime state of the device's memory and extract the information it contains using the Volatile Framework. The specific goals of this project are:

• Understand the basic concepts and techniques of memory forensics.

• Familiarize yourself with the Volatility Framework and its plugins.

• Perform memory imaging of device RAM and extract relevant information using volatility.

• Analyse extracted information to draw conclusions about system runtime state, running processes, and possible malicious activity.

Achieving these goals will demonstrate the importance and effectiveness of memory forensics in investigating security incidents, analysing system performance problems, and troubleshooting software problems. We also discuss the usefulness of the volatility framework as a powerful and flexible storage forensics tool.

**1.2 *Description of the project*:**

This project explores the system runtime state of the device's memory and extracts information from it using the Volatility Framework. A project can be divided into the following steps:

• Preparation: The first step is to prepare your system and environment for memory forensics. This includes setting up a test system with the necessary hardware and software components, installing the volatility framework, and understanding the basic concepts of memory forensics.

• Memory Imaging: The next step is to capture a memory image of the device's memory. This can be done with various tools such as dd, FTK Imager, WinPMEM. A memory dump is a binary file that contains the entire contents of RAM at the time of the dump.

• Volatility Analysis: Once the memory image has been captured, the next step is to analyze it using the Volatility Framework. The Volatility Framework is a collection of plugins that can be used to extract different types of information from memory dumps. Plugins can be used to analyze running processes, network connections, open files, registry keys, and other system information.

• Interpretation of Results: After analyzing the memory image, the results are interpreted to draw conclusions about the run-time status of the system. This includes analyzing information extracted from memory dumps, looking for signs of malicious activity, identifying running processes, and determining the root cause of system performance issues. The purpose of this project is to demonstrate the effectiveness and importance of memory forensics in investigating security incidents, analyzing system performance problems, and troubleshooting software problems. This project uses the Volatility Framework to demonstrate the usefulness of this powerful storage forensics tool.

Top of Form

**1.3 *Scope of the project*:**

The scope of this project is to explore the system runtime state of the device's memory and extract the information present in memory using the Volatility Framework. The purpose of this project is to understand the basics of operating systems, computer memory and RAM, and to use the Volatility Framework to extract digital artifacts from volatile memory in a forensically sound manner. This project involves learning various techniques used in Volatility, such as image acquisition, profiling, and using plugins to extract and analyse data from the device's RAM. The scope of the project also includes examining various plugins for Volatility to extract specific information such as running processes, network connections and registry keys. The ultimate goal of this project is to provide a comprehensive report on the results and analysis of information extracted from the device's RAM. This can be useful in digital forensic investigations, incident response, and system security assessments.

***System Description***

**2.1 *Target system description:***

The target system for this project is a device running a regular Windows 11 operating system. The device has 8 GB of RAM and 1 TB of storage for the framework to work efficiently. The device runs as a normal device and does not work using virtual machines.

**2.2 *Assumptions and Dependencies:***

We assume that the target device has not been compromised and is free of malware and viruses. This project's dependencies include the Volatility Framework and the Python3 programming language.

**2.*3 Functional/Non-Functional Dependencies:***

Volatility Framework has both functional and non-functional dependencies. Functional dependencies include Python 3.6-3.9 and access to target system dumps. Non-functional dependencies include the target device must have at least 4 GB of RAM and sufficient disk space available to store the memory dump.

**2.4 *Data set used in support of your project:***

System does not use data set for support.

***2.5 Methodology:***

The methodology for this project includes the following steps:

Step 1: Capture a memory dump of the target system.

Step 2: Analyse the dump using the Volatility Framework.

Step 3: Extract information from the dump

Step 4: Analyse the extracted information to understand the runtime status of your system.

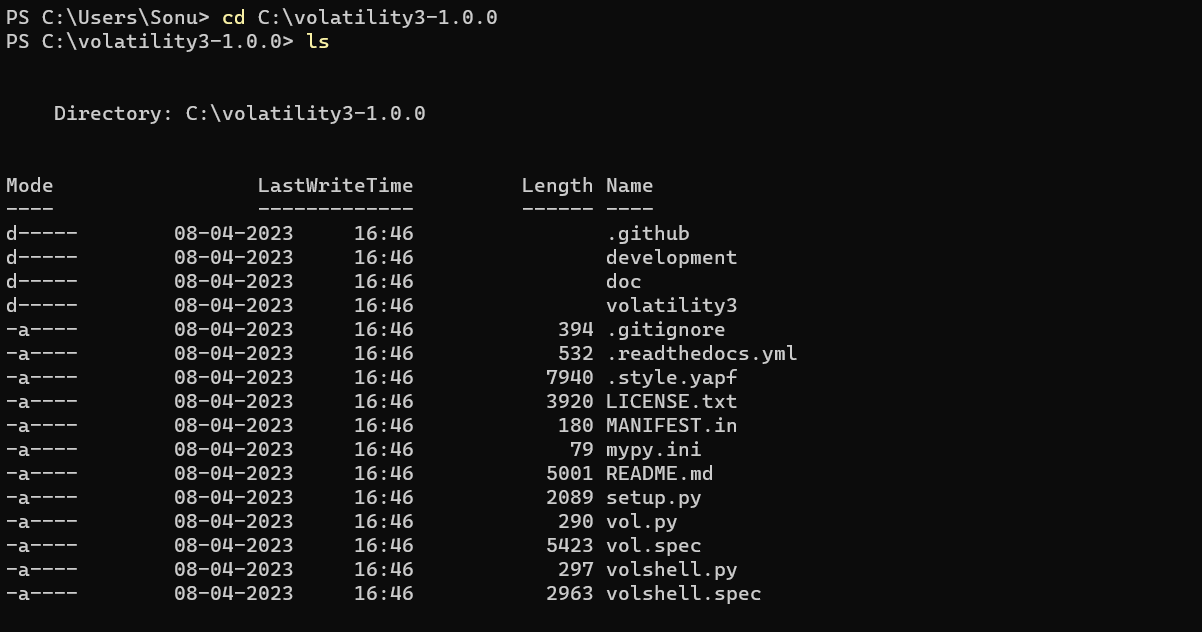
***Analysis Report***

**3.1 *System snapshots and full analysis report****:*

Step 1: Download and Install Volatility Framework First, you need to download and install Volatility Framework on your Windows 11 device. Download from Volatility Framework official website https://www.volatilityfoundation.org/ and follow the instructions to install, or

Download the latest version of Volatility using the following command:

**git clone https://github.com/volatilityfoundation/volatility3.git**



Step 2: Create a Memory Dump To extract the information present in RAM, you need to create a memory dump of your Windows 11 device. You can use any tool to create a memory dump, but in this example, FTK Imager Version 4.7.1 Tool. To create a memory dump, follow these steps:

1. Download FTK Imager from https://www.exterro.com/ftk-product-downloads/ftk-imager-version-4-7-1.
2. Install FTK Imager and run the tool.
3. Click on the File->Capture Memory.
4. Assign the Destination Path and Destination Filename, then click on capture memory.
5. Once you click on capture memory your memory dump will be created on the designated path.

Graphical user interface, text, application

Description automatically generated

Step 3: Analyze the Memory Dump with Volatility Framework After creating the memory dump, you can use the Volatility Framework to analyze the memory dump and extract the information present in RAM. To do this, follow these steps:

1. Open the Windows PowerShell on your Windows 11 device.
2. Navigate to the directory where you have installed the Volatility Framework using the 'cd' command. For example, if you have installed the Volatility Framework in the C:\volatility3-1.0.0, type 'cd C:\volatility3-1.0.0' and press Enter.
3. Type the following command to list the available profiles:

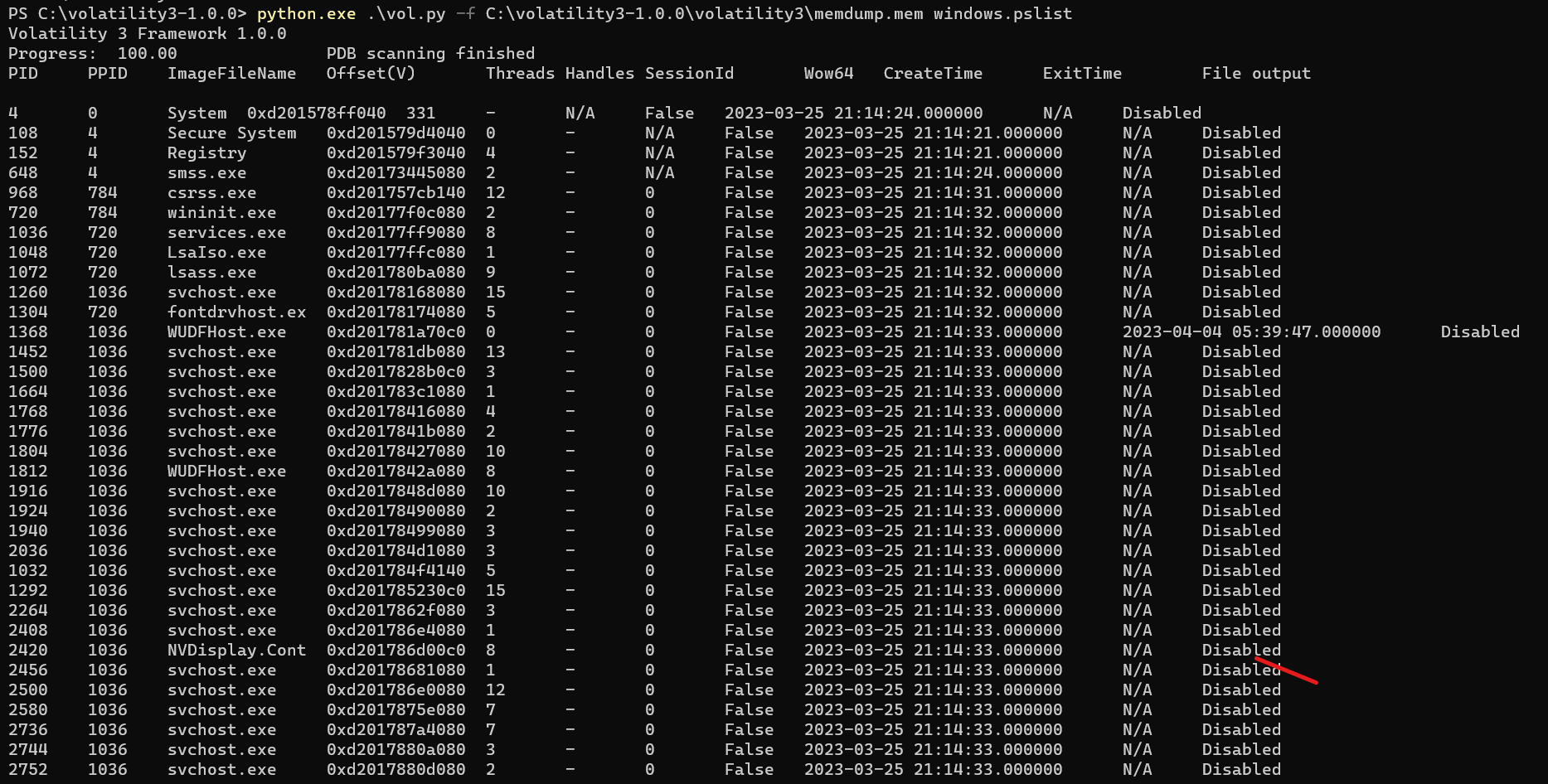
python.exe.\vol.py -f C:\volatility3-1.0.0\volatility3\memdump.mem windows.info

Text

Description automatically generated

1. Type the following command to analyze the memory dump:

python.exe.\vol.py -f C:\volatility3-1.0.0\volatility3\memdump.mem windows.pslist



This command will extract the process list from the memory dump and display it on the screen. You can use other Volatility Framework commands to extract other information present in RAM.

Step 4: Analyze the Memory Dump with Volatility Framework. Here are some Volatility Framework commands that you can use to extract other information present in RAM:

1. To extract the list of open network connections, use the 'netscan' command:

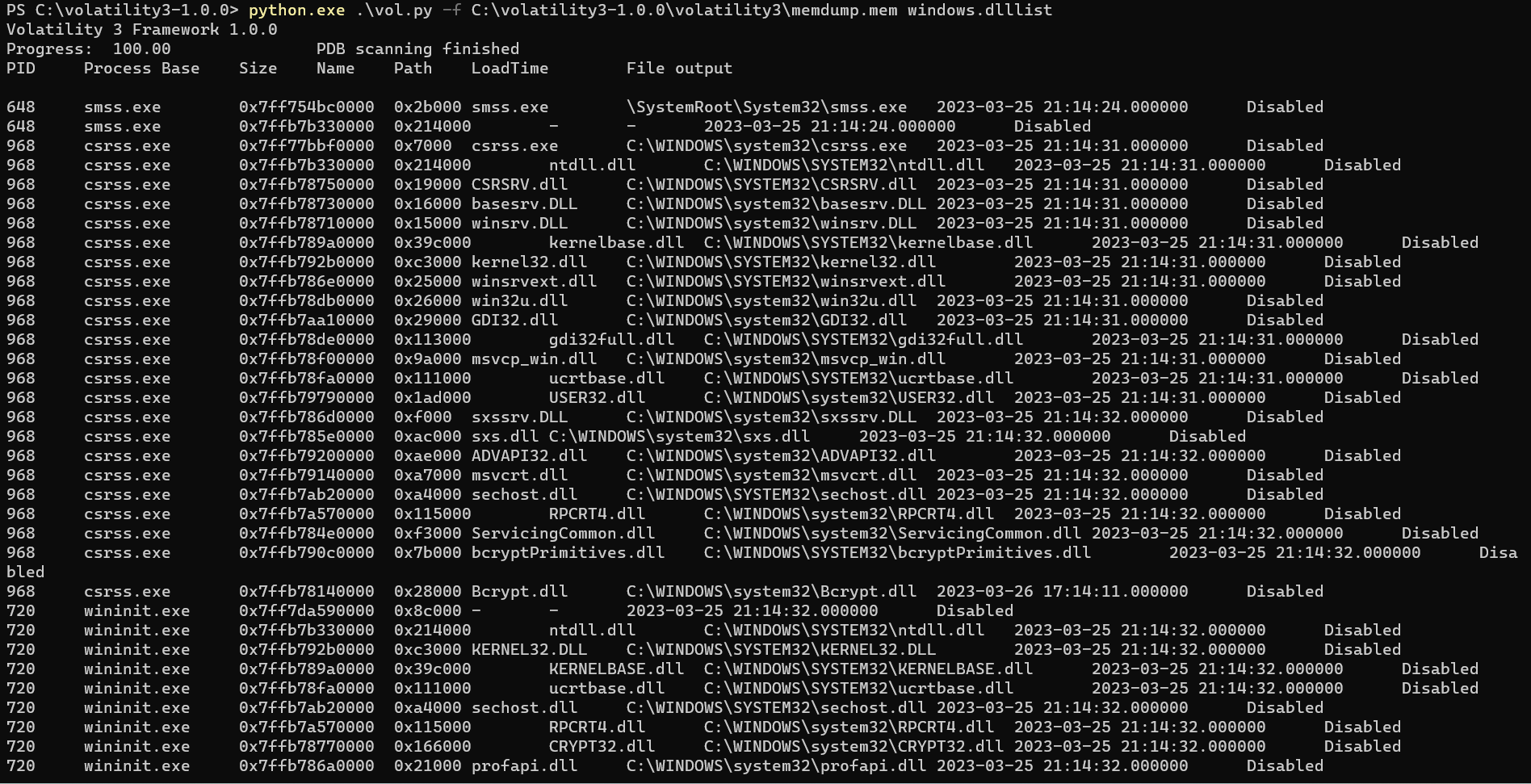
python.exe.\vol.py -f C:\volatility3-1.0.0\volatility3\memdump.mem windows.netscan

A picture containing graphical user interface

Description automatically generated

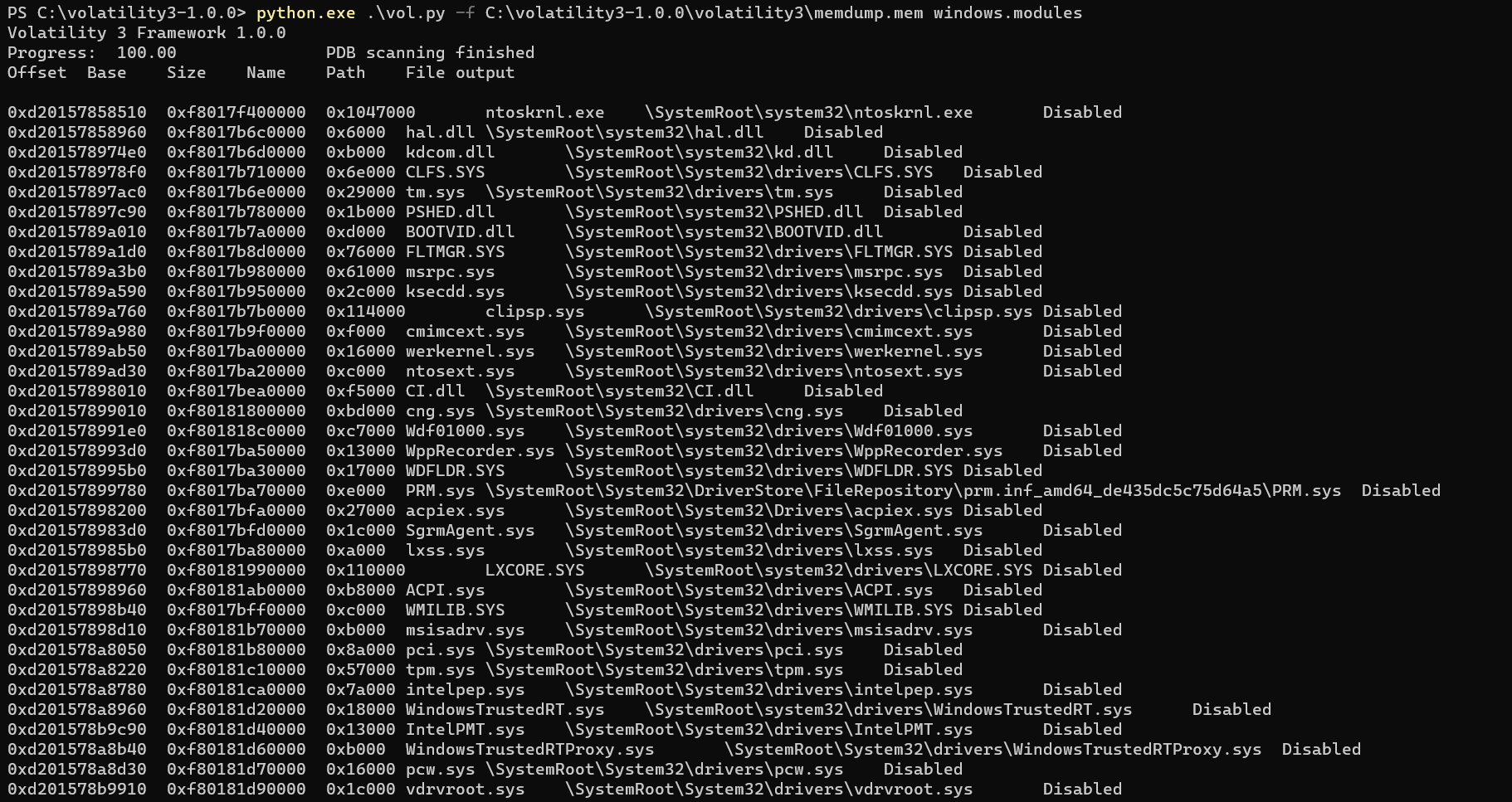
1. To extract the list of running processes and their associated DLLs, use the 'dlllist' command:

python.exe.\vol.py -f C:\volatility3-1.0.0\volatility3\memdump.mem windows.dlllist



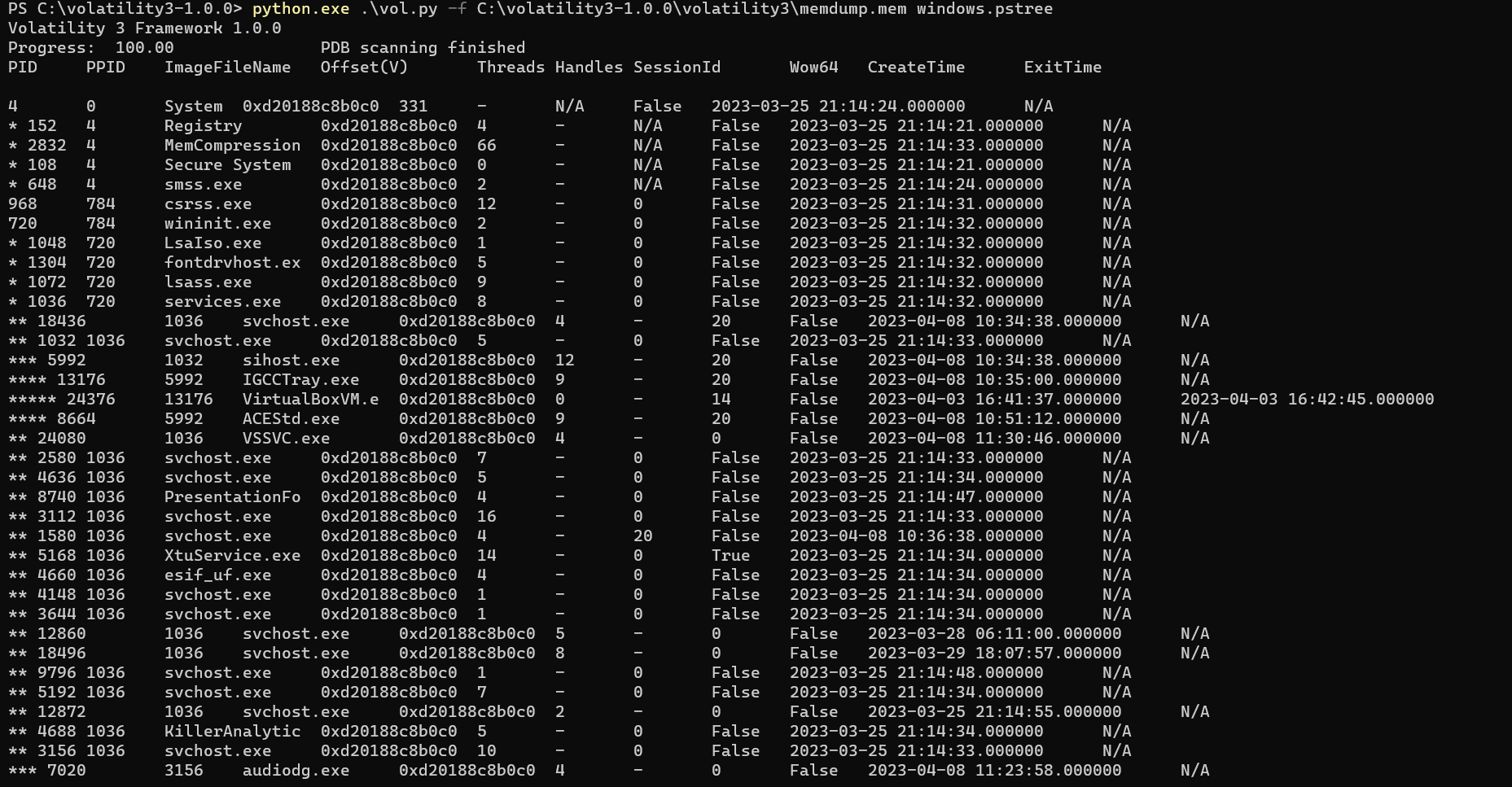
1. To extract the list of active network modules, use the 'modules' command:

python.exe.\vol.py -f C:\volatility3-1.0.0\volatility3\memdump.mem windows.modules

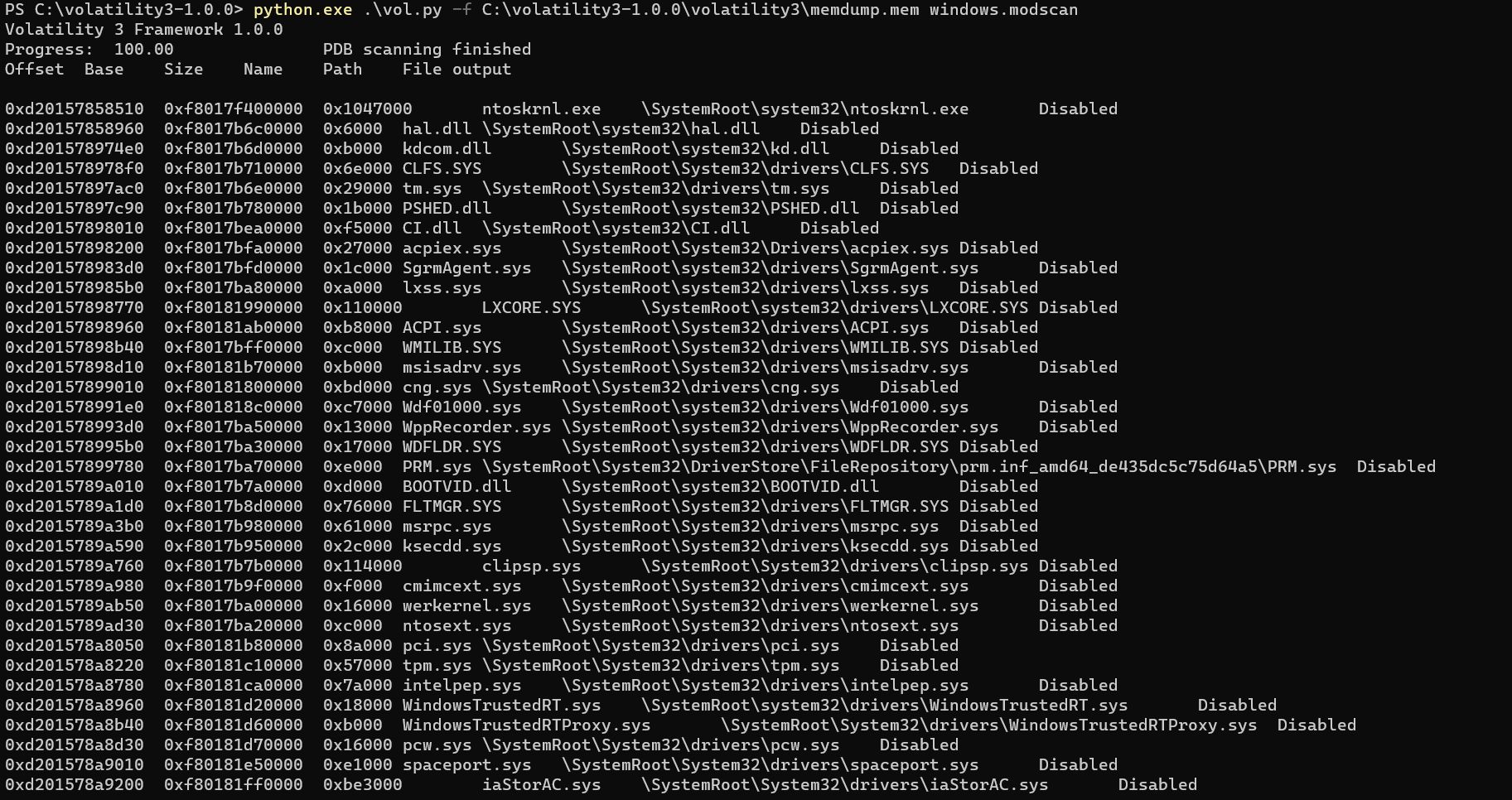


1. To view the process listing in tree form, use the ‘pstree’ commands:

python.exe.\vol.py -f C:\volatility3-1.0.0\volatility3\memdump.mem windows.pstree



1. To analyze the ldr data table entry structure by scanning physical memory for pool tags, use the ‘modscan’ command.



These are just a few examples of the many Volatility Framework commands that you can use to extract information present in RAM. You can explore the Volatility Framework documentation to find more commands and their usage.

Note: It is important to mention that analyzing a memory dump can be a complex and time-consuming process that requires knowledge and experience in memory forensics. It is recommended to use the Volatility Framework under the guidance of a trained professional. Additionally, the use of the Volatility Framework on a live system can be dangerous and may cause instability or damage to the system. Therefore, it is advised to only use it on a copy of the memory dump and not on the live system.

***Reference/Bibliography***

The system uses various open-source software libraries and APIs to collect information about system components and performance metrics. Reference materials used to develop the system include:

* Volatility Framework documentation
* Volatility Framework Symbol Table
* Python 3 Framework documentation
* FTK Imager Documentation
* "Windows Forensic Analysis Toolkit: Advanced Analysis Techniques for Windows 10" by Harlan Carvey.
* "Digital Forensics with Open-Source Tools: Using Open-Source Platform Tools for Performing Computer Forensics on Target Systems" by Cory Altheide and Harlan Carvey.

***Results and Discussion***

The results of this project demonstrate that the Volatility Framework can be used to extract information from core dumps on target systems. You can analyze the extracted information to understand the runtime state of the system. Volatility Framework offers a wide range of plugins that can be used to extract different types of information from memory dumps.

In outline, the comes about of this venture highlight the viability and flexibility of the Volatility System for extricating data from memory dumps of target frameworks. The system gives a wide run of plugins that can be utilized to extricate profitable experiences into the runtime state of a framework, making it a important instrument within the field of computerized forensics. Be that as it may, as with any legal investigation, cautious translation, and relevant understanding of the comes about are fundamental for precise conclusions and decision-making.

***Conclusion***

In conclusion, using the Volatility Framework to examine a device's system runtime state can be a useful tool for forensic analysis. Volatility Framework provides various plugins that can be used to extract information from memory dumps of the target system. You can analyze the extracted information to understand the runtime state of the system. However, it is important to note that using the volatility framework requires a deep understanding of the framework and the system being analyzed.

In summary, open-source software tools like Volatility provide powerful capabilities for capturing and analyzing volatile data temporarily stored in your machine's RAM. These tools can provide valuable insight into system activity and help identify indicators of compromise. The ability to generate detailed reports using these tools is an important aspect of forensic investigations and helps present analysis results clearly and concisely.