After collecting the RAM data from the infected PC, the next critical step is to properly analyse it and discover any significant evidence. The digital forensic investigation team might do a root cause analysis on the data they have gathered to identify the primary reasons of a breach and evaluate the programmes or activities that were carried out on the system.

Volatility Workbench was chosen to analyse the RAM image of the victim machine. It is a command-line tool that lets DFIR teams doing digital forensic investigations in gathering and examining volatile data, or significant information kept in RAM on a computer (Volatility Workbench - a GUI for Volatility Memory Forensics, n.d.). Although this data is temporary, it may include crucial details for investigators. The data may be extracted and analysed using Volatility Workbench without changing or having an impact on the system under investigation.

A screenshot of a computer

Description automatically generated

Figure 1: Volatility Workbench

The Volatility Framework, a popular tool for memory forensics, provides the foundation for the open-source graphical user interface (GUI) known as Volatility Workbench. Memory forensics is the process of extracting useful data from a computer system's volatile memory (RAM) and looking for signs of criminal or suspicious activity.’

|  |  |
| --- | --- |
| Features | Explanation |
| Open source | As an open-source tool, it will offer investigators a wide variety of capabilities to examine volatile memory and extract useful information from it. As a result, the digital forensics community may work together and share information, which assuring its ongoing support and development. |
| Easy to use | The Volatility Workbench's graphical user interface (GUI) is simple to use and straightforward to interface. Hence, It reduces the need for users to manually enter command line parameters by offering a graphical user interface (GUI). Instead, the tool's user interface consists of simple menus, buttons, and choices. This improves accessibility and reduces cognitive effort associated with memorising and accurately formatting command line options during the analysis process. |
| Auto loading | When the system is loaded, the auto-loading function loads the first memory dump file discovered in the current folder. Users will save time and effort because of not having to search for and manually choose the file for examination. Users may rapidly start their investigation and concentrate on looking through the memory dump's contents thanks to the simplified access to memory dump files method. |
| Time stamping | This feature is crucial for documentation, analysis repeatability, forensic integrity, error tracking, troubleshooting, and memory analysis process efficiency. It offers a list of instructions that were carried out in order, ensuring a precise chronology of what was done. Time stamps are useful for recording mistakes, duplicating analysis procedures, and debugging problems. They additionally help in the process optimization and general efficiency of the analysis. |
| Easy saving dumped information | Volatility Workbench makes the process of conserving and storing analysed data simpler by allowing for simple keeping of dumped data. With only a few clicks, users can easily save the results, doing away with the need for manual extraction and outside equipment. The secure storing of crucial analytical output for later reference or sharing is ensured by this function, which increases efficiency and saves time. |

1. Launch the Volatility Workbench application. There is a command description which show the step of using the application. First, select the.mem file that was previously created by clicking the 'Browse Image' option.

A screenshot of a computer

Description automatically generated

Figure 2: Browse the mem file.

2. Next, select the “Window” from the Platform drop-down menu and clicked process list.

A screenshot of a computer

Description automatically generated

Figure 3: Select the platform used

A screenshot of a computer

Description automatically generated

Figure 4: Running the process list.

A screenshot of a computer

Description automatically generated

Figure 5: Process list run completely.

3. Once the process list was showed, choose a command from the dropdown menu and click the 'Run' button. In this scenario, select the 'window.psscan.PsScan' command to tun the psscan. The psscan can offer information about the chance of a rootkit by examining the unlinked or hidden processes (*Psscan - Digital Forensics and Incident Response [Book]*, n.d.). Once the command selected, the description of the chosen command will be displayed in the 'Command Description' area.

A screenshot of a computer

Description automatically generated

Figure 6: Choosing command.

A screenshot of a computer

Description automatically generated

Figure 7 : The command description shows the function of the command.

4. When clicked the “run” button, the chosen command will be performed on the memory analysis data.

A screenshot of a computer

Description automatically generated

Figure 8 : The file was performing with the psscan.

5. Once the process finished, the result of using the command which is psscan will be shown in the log below.

A screenshot of a computer

Description automatically generated

Figure 9 : Result of using psscan was displayed.

6. Clicked the “save to file” button to export the result. Save the result in text file.

A screenshot of a computer

Description automatically generated

Figure 10 : Exporting result.

7. Once the file is exported, a text file is created. The generated result enables us to see the processes that have been executed and easily access their details. For instance, we can identify suspicious processes that lack a parent process ID, which may indicate malicious activity. This is how we analyze and investigate the RAM image to gain insights and make determinations.

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure 11 : Result.

DLL Injection is a method that lets users run code in another program's memory by manipulating it into loading an external DLL file. This is hazardous since it enables attackers to run scripts on programs covertly. As a result, a DLL injector might be used by hackers to introduce malicious code into a trustworthy program, leading to prohibited behaviours like system hacking or data theft. The malicious DLL injector which named is found on the downloads file as shown in figure X. The suspicious file name called “DLLInjectorv2.exe” and the path location is “/img\_HDD Image (Windows 10 Infected).E01/vol\_vol3/Users/Maryam.var/Downloads. Next, the investigator will extract the file and uploaded in on the online database which is Virus Total for further analysis.

A screenshot of a computer

Description automatically generated

Figure 11 : In downloads file.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11 : Source File Metadata For The ‘DLLInjectorv2.exe’ File

Based on the information provided, this file has undergone analysis by 43% security vendors flagged the file as malicious. it can be inferred that the DLL Injector v2 is considered a suspicious application that may be used by hackers to secretly inject malicious code into running processes.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 12: Analysis Result For The ‘DLLInjectorv2.exe’ File

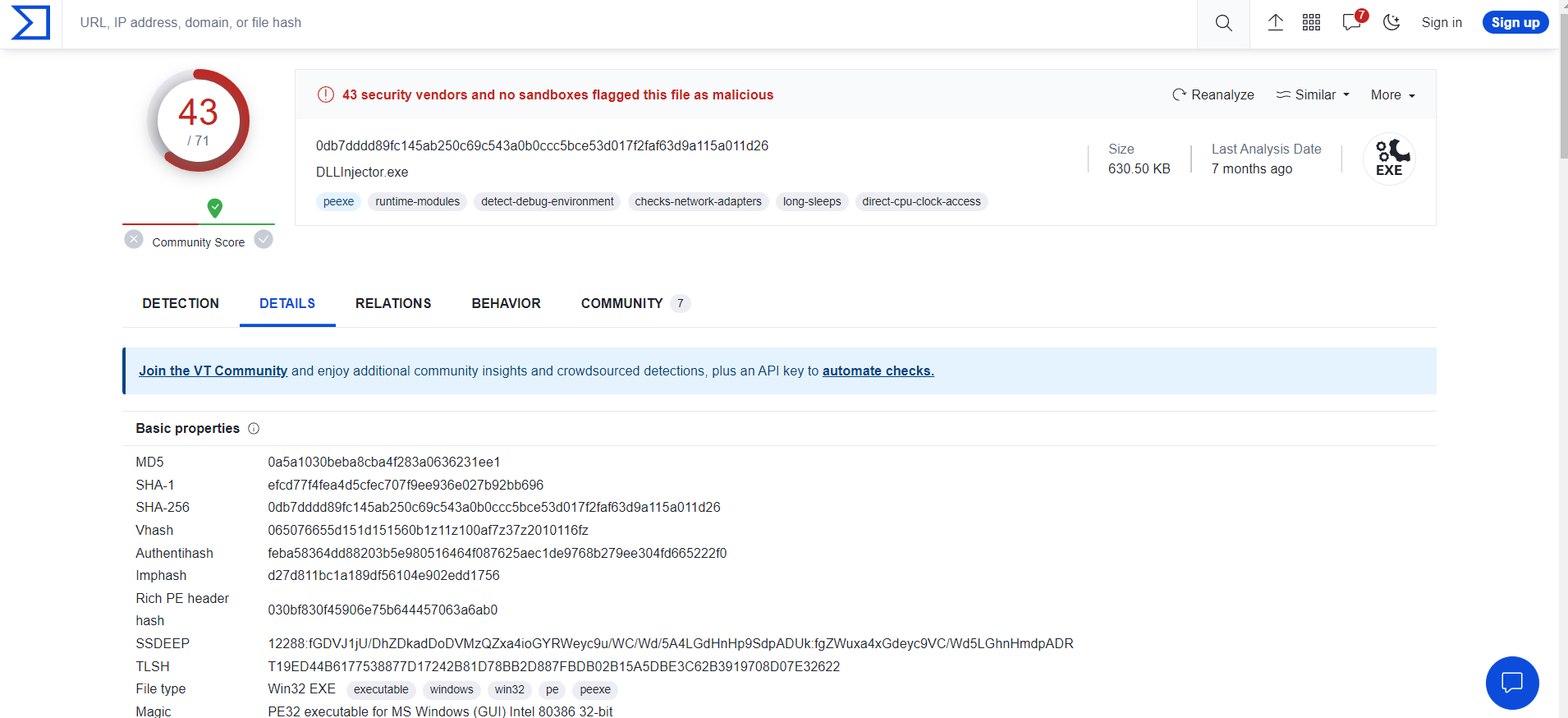


Figure 11 : Analysis Result For The ‘DLLInjectorv2.exe’ File