Intro to Cyber Forensics Lab Grading Sheet

Project: Lab #7 – Memory Forensics
Member Name: VINUSHA GOUD POTTOLLA
Member Name: OLUWATOYOSI KEHINDE
Member Name: BOINAPELLY AKSHITH RAO
Member Name: AMANI PONMAN
Member Name: YADLAPALLI IAKSHMIDAR

Executive Summary / 4 points
+ ✓ -
❑ ❑ ❑ Executive summary is brief and focused to the point of the project
❑ ❑ ❑ The summary clearly illustrates the objectives of the laboratory exercise

Apparatus / 4 points
❑ ❑ ❑ The apparatus are clearly illustrated and documented

Procedures / 12 points
❑ ❑ ❑ Adequate information provided to allow re-creation of work
❑ ❑ ❑ Consistent level of coverage throughout the project – nothing overly detailed or omitted
Problem Solving / 5 points
❑ ❑ ❑ All problems identified
❑ ❑ ❑ Alternative solutions identified
❑ ❑ ❑ Solutions attempted listed
❑ ❑ ❑ Final solution detailed (what fixed the problem and why?)
Conclusions & Recommendations / 5 points
❑ ❑ ❑ Tie back to the learning objectives identified in the executive summary - critical
❑ ❑ ❑ Conclusions stated in a logical fashion
❑ ❑ ❑ Conclusions are viable based on the procedures and results
❑ ❑ ❑ Recommendations practical & relevant

Format & Grammar / 5 points
❑ ❑ ❑ Table of Contents present
❑ ❑ ❑ Report written in past tense
❑ ❑ ❑ Proper voice (no I's, We's, Our's or The group)
❑ ❑ ❑ Paper easy to read (fonts, spacing, etc.)
❑ ❑ ❑ Proper credit given to sources in bibliography (APA style)
❑ ❑ ❑ Paper is cohesive and consistent in tone
Spelling & grammar errors: minus one half point for each, up to a max deduction of 5 points – at that
time, paper is returned for correction and re-submission with a one letter grade penalty.

Final Score _____ / 35

Contents

1.Executive summary

The exercise's main goal was to familiarize participants with the procedures involved in acquiring volatile memory and analysing it using forensic techniques and tools. Being vigilant and adhering to all rules and procedures was necessary to ensure that the data was safe, undamaged, and intact both throughout acquisition and processing. Group 1 carried out this experiment on December 6, 2024, with a focus on collaboration and rigorous adherence to forensic criteria.

During the memory acquisition phase, DumpIt a dependable tool for volatile memory capture with minimal interference—was used. For integrity preservation, the obtained memory dump was safely moved to a forensic workstation. The major tool for analysing the contents of the memory dump file was the Volatility 3 software, which supported Python 3.13.10.

Setting up the proper operating system profile, looking at running and stopped processes, researching parent-child relationships, and spotting malicious or hidden activity were the main phases. Every one of these procedures offered a thorough analysis of the memory condition at the time of collection, which gave forensic investigators useful information.

Throughout the exercise, controlled procedures were used to guarantee that the data was safe and undisturbed. The entire exercise clearly emphasizes that a systematic and organized approach to memory forensics would produce reliable and repeatable results.

2. Apparatus

ITEM/PART	MODEL NUMBER	VERSION	USAGE
Memory Acquisition Tool	DumpIt	Latest	To acquire memory from the system.
Analysis Tool	Volatility	3	Forensic analysis of the acquired memory.
Python Interpreter	N/A	3.13.10	Dependency for running Volatility.

Table 1: The Hardware and Software used for the lab exercise.

3. Procedures

3.1. Time-Log

#	DATE	Time(24htr)	ACTION TAKEN / INVESTIGATION LEAD
1	December 02,2024	19:20	Opened the existing Windows Virtual Machine
2	December 02,2024	19:26	Downloaded Dumplit.exe and took the memory dump
3	December 02,2024	19:40	Analyzed evilprofessor.vmem using volatility
4	December 02,2024	19:58	Results were summarized

Table 2: Time log of the actions taken during investigation

3.2. Procedure

The Group 1 team completed two lab components on December 6, 2024, including memory acquisition and processing. To dump volatile memory, the first section (Memory Acquisition Lab) runs DumpIt.exe with administrator rights. The raw file dump was safely moved to a forensic workstation for analysis after the memory dump file was successfully created.

Volatility 3, which requires Python 3.13.10, was used to examine the memory dump file (evil professor.vmem) on the forensic workstation for the second section, the Memory Analysis Lab. The windows.info. Info plugin was used to identify the proper OS profile before the study began. This was followed using a variety of plugins to investigate live processes, parent-child relationships, and potential malicious or concealed behaviour.

3.3 Figures

Memory Acquisition Lab

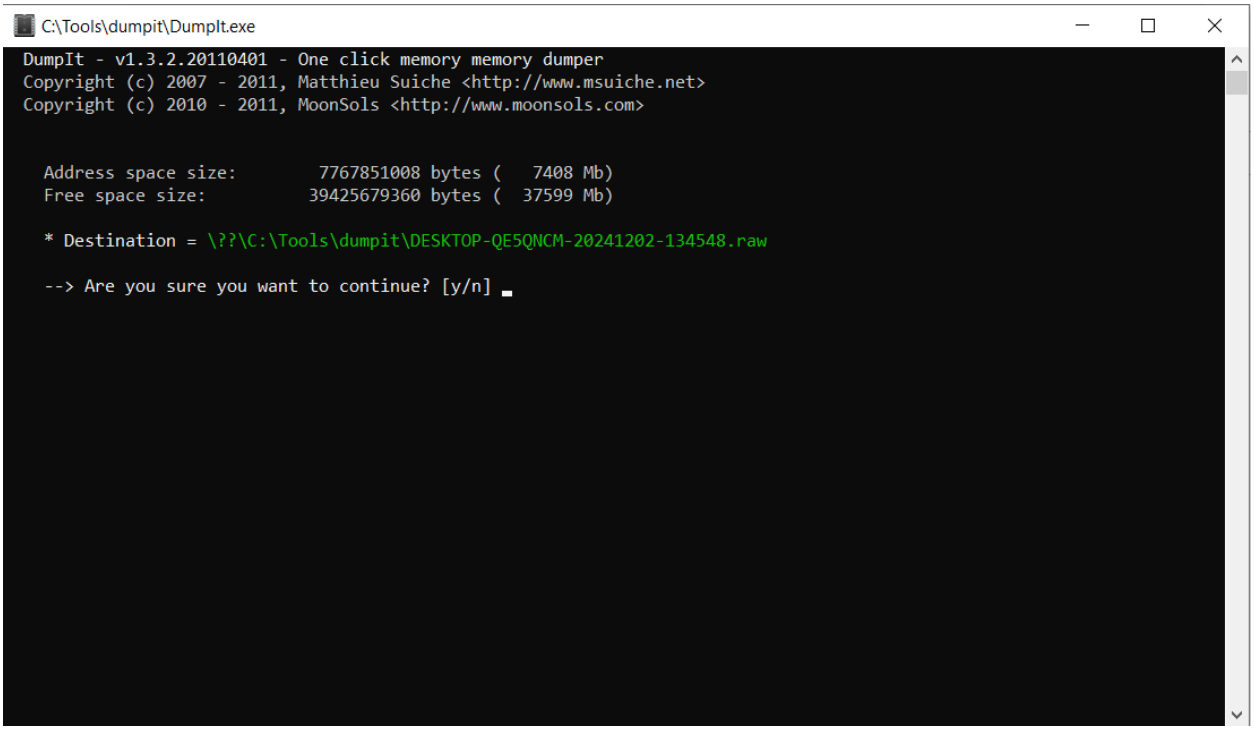

Fig 1: Asking for permissions to get the memory

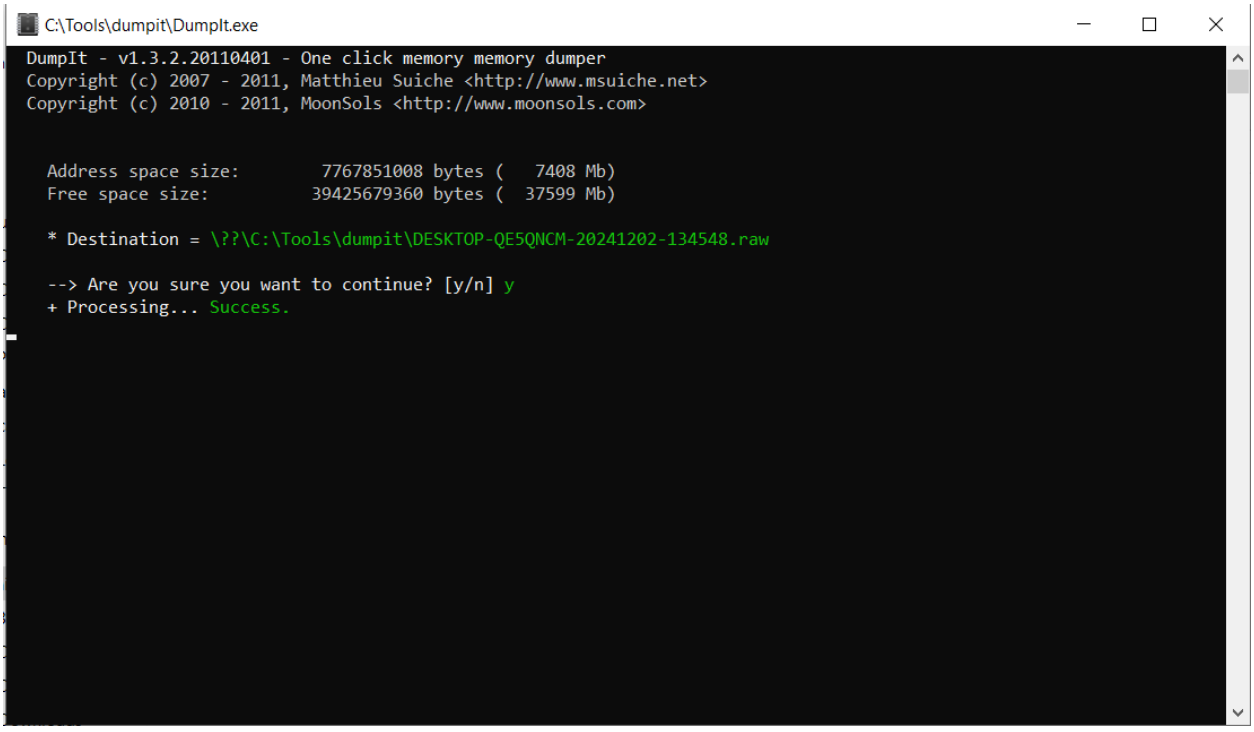

Fig 2: obtaining a raw file from the directory successfully

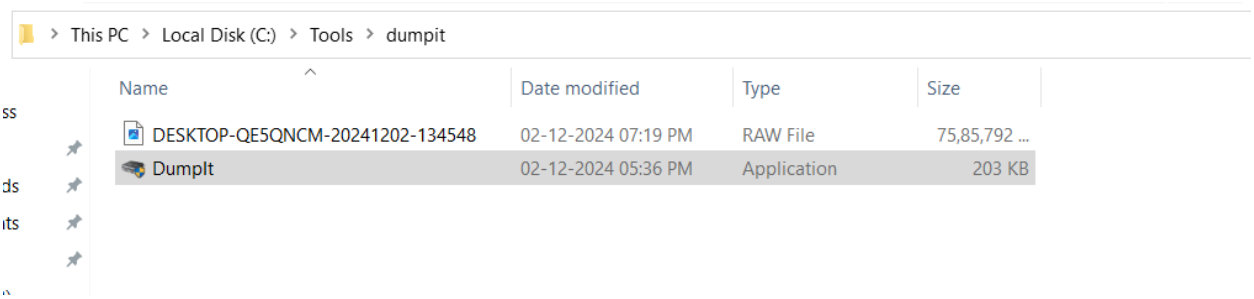

Fig 3: The obtained raw file

Memory Analysis Lab

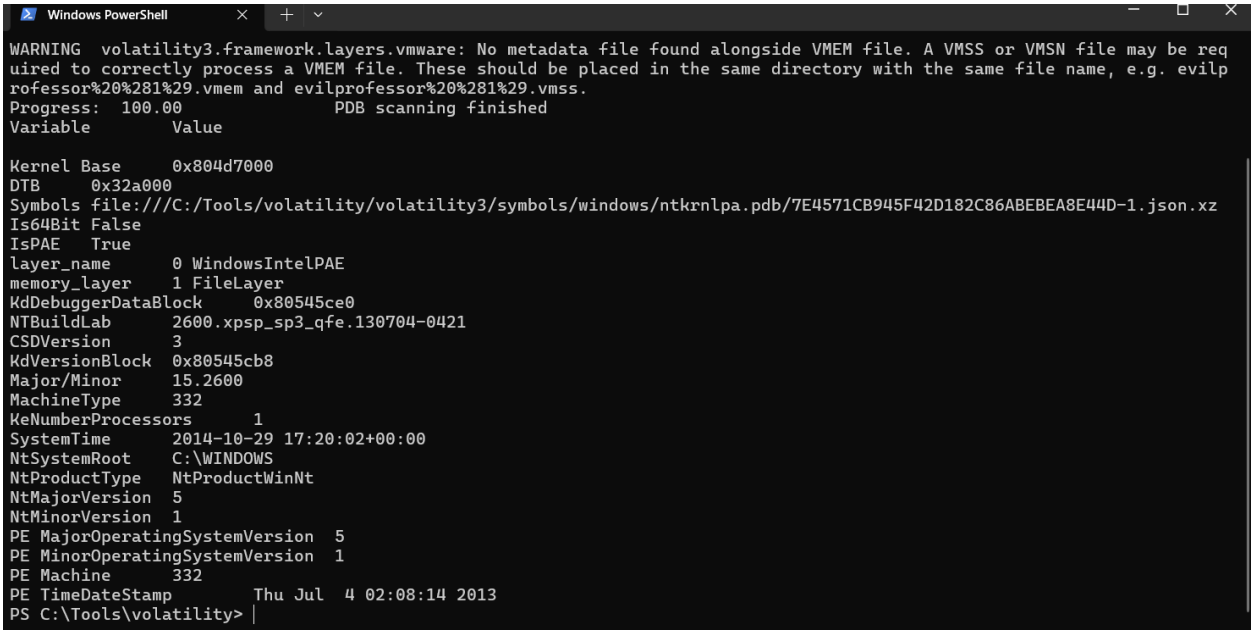

Fig 4: Checking the profile using volatility

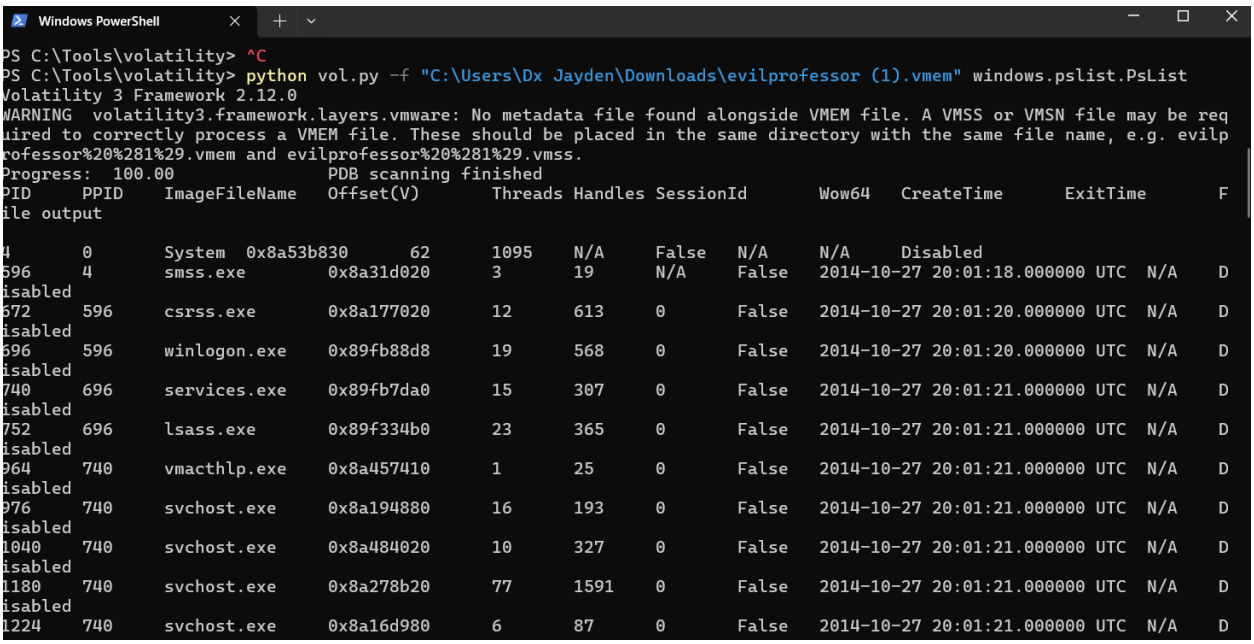

Fig 5: Finding the oldest processes, launching cygrunsrv.exe, detecting active processes, and verifying instances of the cmd.exe process

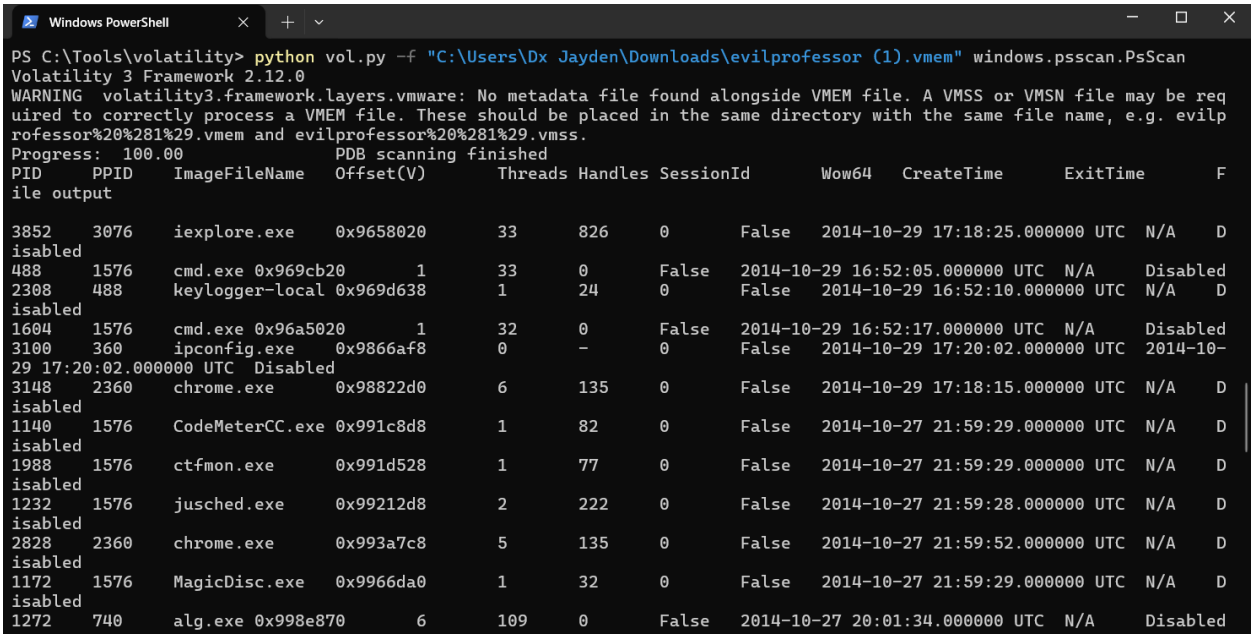

Fig 6: Checking the instances of cmd process

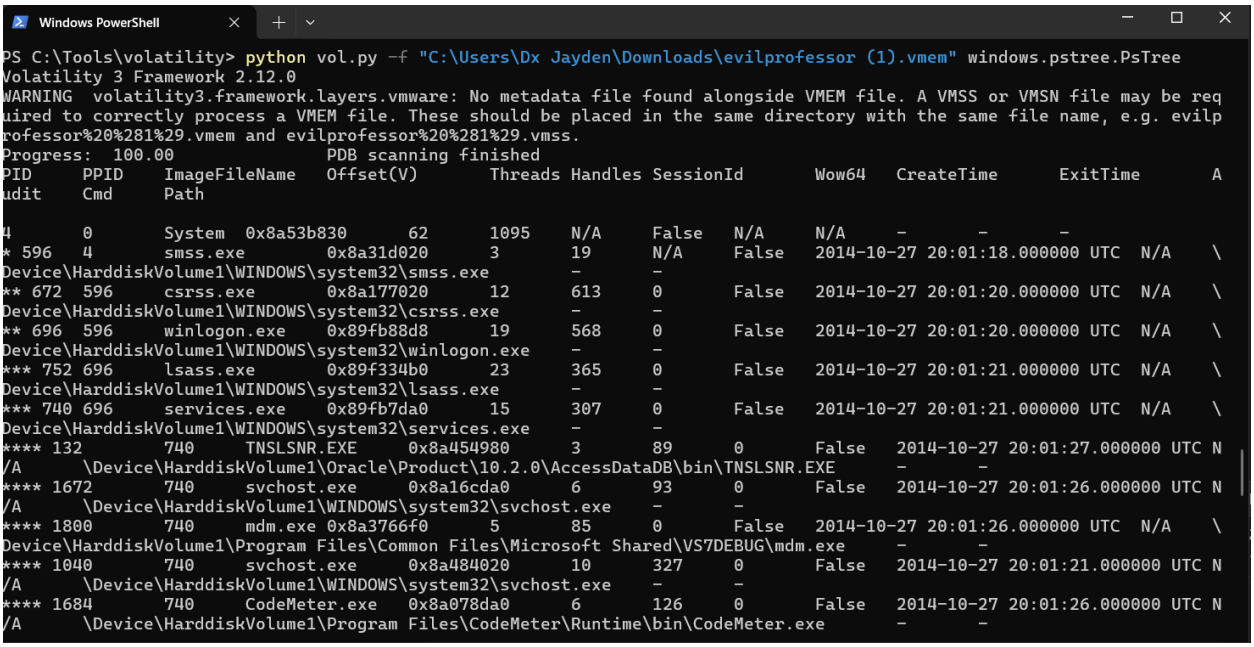

Fig 7: To check the parent-child relationships and identifying terminated processes, sshd.exe service spawning & spawning of mysqld-nt.exe

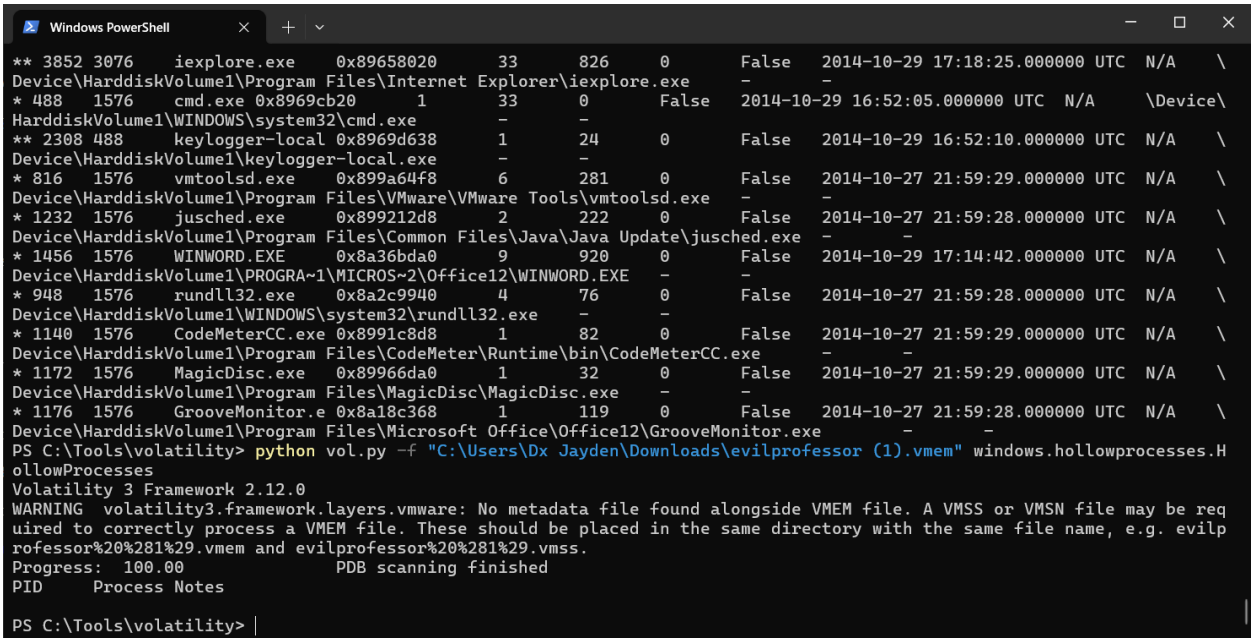

Fig 8: Checking of hidden processes

4. Problem Solving and Troubleshooting

Memory Acquisition Lab

Problem: Usage of DumpIt Tool to acquire memory for analysis

Solution: When the DumpIt utility was used, the proper administrator rights were used. To enable the tool to access the system memory, a permission screen first arose, requesting confirmation. The memory acquisition was finished without any more problems after the proper permissions were given.

Memory Analysis Lab

Problem 1: Determining Active Processes

Solution: When the windows.pslist.PsList plugin was executed, the RAM dump showed 28 running processes.

Problem 2: Detecting Instances of cmd.exe

Solution: Three instances of cmd.exe utilizing the windows.pslist.PsList and windows.psscan.PsScan plugins were found throughout the study. According to this plugin, we observe that three cmd.exe processes are active.

Problem 3: Identifying Parent-Child Relationships and Terminated Processes

Solution: The parent-child relationships were displayed and 27 terminated processes were found using the windows.pstree.PsTree plugin. The following are the names of some of the processes: Services, TNSLSNR, svchost, CodeMeter, lsass, winlogon, csrss, and SMS.

Problem 4: Determining the Oldest Executing Process

Solution: Keylogger-local was identified as the oldest running process by the windows.pslist.PsList plugin.

Problem 5: Identifying the Parent Process of sshd.exe

Solution: Using the windows.pstree.PsTree plugin, it was discovered that cygrunsrv.exe was the parent process of sshd.exe.

Problem 6: Identifying the Parent Process of mysqld-nt.exe

Solution: Using the windows.pstree.PsTree plugin, it was discovered that services.exe was the one that started the mysqld-nt.exe process.

Problem 7: Detecting Hidden Processes

Solution: No hidden processes were detected using the windows.hidden_modules.HiddenModules and windows.hollowprocesses.HollowProcesses plugins.

The reasons are given here as follows:

- No Malicious Activity Detected: System Integrity
- Memory Dump Completeness.
- No Use of Anti-Forensic Techniques.
- Type of Malware

5. Conclusion and Recommendations

The lab showed fundamental methods for acquiring and analyzing memories. Parent-child relationships, terminated processes, and running processes were all revealed by tools such as DumpIt and Volatility. The system seems to be clean during memory capture if there are no hidden processes.

Recommendations

Future laboratories should incorporate more complex situations that include realistic issues with malware obfuscation and anti-forensics. Furthermore, employing a variety of memory analysis methods would broaden viewpoints regarding forensic techniques.

6. Reference

- Volatility Foundation. (n.d.). *Volatility Framework Documentation*. Retrieved from Volatility Docs.
- DumpIt Tool Documentation. (n.d.). *Tool Guide for Memory Acquisition.*