Case Name: Memory Analysis

Couse Name: IST402

Instructor: Robert Price

Date: 03/25/2025

Examiner Name: Jaspreet Singh

Table of Contents

**LIST OF ILLUSTRATIVE MATERIALS3**

TABLES3

FIGURES3

**EXECUTIVE SUMMARY4**

BACKGROUND4

EVIDENCE4

**COLLECTION AND ANALYSIS5**

COLLECTION5

ANALYSIS 6

DumpIT – Memory Capture6

Armitage – Simulating an Attack7

Volatility – Memory Analysis8

**CONCLISION10**

DumpIT10

Armitage10

Volatility11

List of Illustrative Materials

**Tables1**

Table 1: Evidence4

**Figures5**

DumpIT

Figure 1: DumpIt.exe ram1.dd created 6

Figure 2: Copying ram1.dd6

Figure 3: Volatility-2.5.exe 7

Figure 4: Volatility-2.5.exe find “chrome”7

Armitage

Figure 5: Using Armitage to exploit system 8

Volatility

Figure 6: Creating ram2.dd file9

Figure 7: Moving the ram2.dd file into the Local C: Drive 9

Figure 8: Finding “445” 10

# EXECUTIVE SUMMARY

## Background

Throughout this investigation, the examiner primarily focuses on memory forensics and detecting signs of a compromised system through volatile data analysis. Since data stored in RAM is lost once a machine is powered down, the examiner captures a memory image beforehand using a forensic tool called DumpIt, which creates a full dump of the system’s physical memory as a .raw image file.

To simulate an attack scenario, the examiner uses Armitage, a graphical interface for Metasploit, a widely used penetration testing framework. Armitage enables the examiner to identify and exploit system vulnerabilities through an accessible and visual interface.

After the simulated attack is carried out, the examiner applies Volatility, an open-source memory forensics tool written in Python, to analyze the captured RAM images. This tool extracts valuable data such as running processes, network connections, and system-level activity. By examining memory before and after the simulated attack, the examiner can identify signs of compromise and unauthorized access within the system.

### Evidence

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | File Name/Path | Source | Date/Time Collected | Examiner |
| Evidence | ram1.dd | Windows Server | 03/30/2025 at 8:35 PM: Captured before the attack | Jaspreet Singh |
| Evidence | ram2.dd | Windows Server | 04/30/2025 at 9:06 PM: Captured after the systems been compromised | Jaspreet Singh |

# COLLECTION AND ANALYSIS

## Collection

The examiner began by first powering on the target Windows Server machine and launched basic applications such as Google Chrome to ensure that identifiable processes would be able to be present in memory. This was crucial for the examiner to do because it helps provide clear indicators when reviewing running processes later in the analysis phase of this forensic investigation.

Before any simulated attack was executed, the examiner used the forensic memory acquisition tool called DumpIt, to create a snapshot of the system’s physical memory. Then once the tool was executed through the Windows command prompt line, and after confirming its operation, it was able to generate the examiner with a raw memory image. This initial dump of memory was saved and renamed to ram1.dd. Therefore, by capturing this image before the attack it was able to provide the examiner with a baseline reference of the system’s normal and uncompromised state.

Once this memory image has been completed by using DumpIt, the examiner initialized a simulated an attack on the system using Armitage on a Kali Linux machine. Then after confirming that the system was successfully compromised, the examiner was able to return to the Windows Server machine to perform a second memory capture like before by using the forensic tool DumpIt. Then once this second image has been generated the examiner was able to rename this image to ram2.dd and this memory image represents how the system is after an attack has occurred.

Then the examiner was able to take both memory images, ram1.dd and ram2.dd, and transfer them to a secure location to perform further analysis. Having two separate captures of the system, one before and one after the attack, was crucial for the examiner to conduct a forensic review and compare them by highlighting any changes in running processes, network connections, or other memory-based artifacts introduced by the simulated attack. These memory images were then later analyzed by the examiner by using the open-source analysis tool Volatility, a memory forensics tool that enables in-depth examination of volatile data.

## Analysis

#### DumpIt.exe directory

The examiner begins by opening the command prompt and running the following command c:\>DumpIt.exe to capture a complete snapshot of the system’s physical memory (RAM). Once executed the examiner would view the directory and see that a new raw file was added which includes the whole snapshot of the system. The examiner then renames this file to ram1.dd by using the following command C:\> ren \*.raw ram1.dd. Once this is completed the examiner can check the directory and verify that ram1.dd is present.

A computer screen shot of white text

AI-generated content may be incorrect.

Figure 1: DumpIt.exe ram1.dd created

#### Copying the Memory Dump

The examiner is creating a copy of the raw file ram1.dd by using the following command, copy x:\\*.dd c:\. Then the examiner is able to view the directory again by using the command dir to verify that a copy has been created.

A computer screen shot of a program

AI-generated content may be incorrect.

Figure 2: Copying ram1.dd

#### Volatility Image info

The examiner then runs a Volatility command on the ram1.dd file to gather information about the system’s memory. The imageinfo command provides the examiner with details about the system profile, including the operating system version, architecture, and number of processors. This information is essential for selecting the correct profile for further analysis, such as identifying running processes and analyzing network connections.

A computer screen shot of a program

AI-generated content may be incorrect.

Figure 3: Volatility-2.5.exe

#### Finding “Chrome”

The examiner uses the Volatility command on the ram1.dd file to filter and identify instances where chrome.exe was running at the time of the memory capture. After executing the command, the examiner detects two active instances of chrome.exe in the memory image. This information establishes a baseline for comparison with the post-attack memory capture to identify any anomalies or unauthorized processes.A screen shot of a computer

AI-generated content may be incorrect.

Figure 4: Volatility-2.5.exe find “chrome”

#### Armitage

The examiner utilizes Armitage, a graphical user interface (GUI) front-end for Metasploit, which is designed to simplify and enhance the penetration testing process. In this investigation, the examiner uses Armitage to exploit a vulnerability in the target Windows system by selecting the SMB (Server Message Block) category and identifying a specific exploit named ms09\_050\_smb2\_negotiate\_func\_index.

This exploit targets a vulnerability in the SMBv2 protocol, which allows remote attackers to gain elevated privileges on vulnerable Windows systems. After selecting the exploit, the examiner configures it by checking the option to "use a reverse connection," which establishes a connection back to the attacker's machine, enabling remote access and control over the compromised system. The examiner then launches the exploit, leading to the successful compromise of the target system with SYSTEM-level privileges.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5: Using Armitage to exploit system

#### Deleting ram1.dd

The examiner begins by deleting the original ram1.dd file using the command del ram1.dd and verifies its removal by executing the dir command to list the contents of the directory. After ensuring that the original memory dump is deleted, the examiner runs DumpIt.exe again to capture a new snapshot of the system’s RAM after the simulated attack. Once the new memory dump is generated, the examiner renames the file from its default .raw extension to ram2.dd using the command ren \*.raw ram2.dd. Finally, the examiner performs another dir command to confirm that ram2.dd is successfully created and prepared for analysis.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 6: Creating ram2.dd file

#### Windows Key moving file

The examiner successfully transfers the ram2.dd file to the local C:\ drive after accessing the target system by using the Windows key and Run dialog. The examiner typed \\192.168.1.100\C$ to establish a connection to the target system’s C:\ drive over the network and authenticated as an administrator to complete the transfer. This ensures that the post-attack memory dump is securely stored and ready for analysis.A screenshot of a computer

AI-generated content may be incorrect.

Figure 7: Moving the ram2.dd file into the Local C: Drive

#### Volatility find “445”

In the command prompt, the examiner launches Volatility and uses the netscan command to analyze the ram2.dd memory dump for network activity. This command scans for open, closed, or listening network connections captured in the system’s memory. To narrow down the results, the examiner applies a filter to display only connections related to port 445, which is commonly associated with the SMB (Server Message Block) protocol.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 8: Finding “445”

# Conclusion

Overall, throughout this lab, the examiner was able to gain hands-on experience using multiple forensic tools to capture, analyze, and identify potential system compromises through memory forensics. For instance, the examiner used the following forensic memory tools, DumpIt, Armitage, and Volatility to simulate a real-world cyberattack, analyze the system’s activities, and identify post-attack artifacts that were preserved in memory dumps.

### DumpIt

The examiner was able to utilize DumpIt to create a snapshot of the system’s volatile memory at two points during this lab. First was before the attack and after the attack was simulated. The first memory capture called ram1.dd provided the examiner with a baseline image of the system’s state prior to being exploited. Then after the attack was carried out by using Armitage, the examiner was able to create a second memory dump called ram2.dd which captured the systems memory after the attack had been carried out. Both these memory dumps were essential for conducting a comparative forensic analysis on the system and identifying any changes introduced by the simulated attack.

### Armitage

The examiner utilized the graphical interface for Metasploit called Armitage, to simulate a successful cyberattack against the target windows system. This was completed by the examiner by selecting the ms09\_050\_smb2\_negotiate\_func\_index exploit from the smb category. Then from there the examiner was able to compromise the system and gain system-level privileges. The use of the reverse connection allowed the examiner to maintain control over the system, providing the examiner with an opportunity to analyze the system after it has been exploited.

### Volatility

Finally, the examiner analyzed the memory images using Volatility, an open-source memory forensics tool. The examiner applied the appropriate profile, Win2008SP1x86, to identify running processes using the pslist command and examine network connections through the netscan command. The analysis revealed multiple listening connections on port 445 and highlighted an established connection as well. These findings confirmed the presence of post-exploitation activity and provided evidence of potential system compromise.