Windows Memory Forensics

Technical Report · March 2023

1 author:

Derar Abu Sheihka
Arab American University
6 PUBLICATIONS   **2** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   Best Kali Linux Tools for Hacking View project

Project   CSP Audit and Compliance View project

Windows Memory Forensics I

Derar Abu Sheikha

Contents

OVERVIEW

Memory forensics has been a crucial part of an investigation process for some time now. RAM can provide – and provides – invaluable information on what is happening on the system at any given moment. It is also able to reveal traces of activity that has already taken place, leaving a trail of breadcrumbs behind – in memory. In this series we will introduce the process of memory analysis – from acquiring a sample of RAM to extracting juicy data and interpretation of contained information.

Windows Memory Forensics is a technique used in digital forensics investigations to extract and analyze volatile data from the memory of a Windows computer system. The memory of a computer is a valuable source of information that can contain evidence of system and user activity, including running processes, network connections, user credentials, and malware artifacts.

There are several tools and techniques available for Windows Memory Forensics, including:

Memory Imaging: This involves creating a bit-by-bit copy of the memory of the target system using tools like FTK Imager or DumpIt. This copy can then be analyzed using memory forensics tools.

Volatility Framework: Volatility is an open-source memory forensics framework that supports the analysis of memory dumps from Windows, Linux, and Mac systems. It provides a range of plugins to extract information from the memory dump, including running processes, network connections, registry keys, and user accounts.

Rekall Framework: Rekall is another open-source memory forensics framework that provides a range of plugins for analyzing memory dumps from Windows, Linux, and Mac systems. It offers advanced memory analysis features like automated memory reconstruction, cross-memory analysis, and memory diffing.

Windows Sysinternals Suite: This is a collection of Windows system utilities that includes tools like Process Explorer, TCPView, and Autoruns. These tools can be used to identify and analyze running processes, network connections, and system configuration settings.

Manual Analysis: Memory dumps can also be analyzed manually using tools like WinDbg or OllyDbg. This approach requires more advanced technical skills and can be time-consuming, but it provides more granular control over the analysis process.

In summary, Windows Memory Forensics is a powerful technique that can provide valuable insights into system and user activity on a Windows computer. It requires specialized tools and technical expertise, but it can yield critical evidence in digital forensics investigations.

MEMORY ACQUISITION

MEMORY CAPTURE TOOLS

Before we can even think about memory analysis, we need to figure out a way to obtain an image of RAM. There are several tools available to us that allow for the RAM capture. An important thing to remember is that any tool we choose will leave a trace on the system we are investigating, which is why we never install any memory acquisition tool on the source system (if not absolutely necessary for some reason). Be sure to document your steps so that in the case of a lawsuit, acquired evidence may be accepted in the court of law and the entire investigation process would not be disputable.

Memory capture tools are software programs used to create a copy of the memory (RAM) of a computer system. These tools are commonly used in digital forensics investigations to extract volatile data from a computer's memory, such as running processes, network connections, and user activity. Here are some popular memory capture tools:

FTK Imager: FTK Imager is a forensic imaging tool that can be used to create a memory dump of a Windows system's RAM. It can also create images of hard drives, network shares, and other storage devices.

DumpIt: DumpIt is a lightweight memory capture tool developed by the Volatility Foundation. It can be run from a USB drive and is designed to quickly create a memory dump of a Windows system.

Redline: Redline is a free memory analysis tool developed by FireEye. It includes a memory capture feature that allows investigators to create a memory dump of a Windows system and analyze it using the Redline application.

WinPmem: WinPmem is an open-source memory acquisition tool developed by the Volatility Foundation. It can create a memory dump of a Windows system without installing any drivers or modifying the system.

Magnet RAM Capture: Magnet RAM Capture is a memory capture tool developed by Magnet Forensics. It can create a memory dump of a Windows system and analyze it using the Magnet AXIOM digital forensics platform.

Belkasoft Live RAM Capturer: Belkasoft Live RAM Capturer is a free memory capture tool developed by Belkasoft. It can create a memory dump of a Windows system and analyze it using the Belkasoft Evidence Center digital forensics software.

In summary, memory capture tools are essential for digital forensics investigations, as they allow investigators to extract volatile data from a computer system's memory. There are many memory capture tools available, each with its own features and capabilities. It is important for investigators to choose the right tool for their specific needs and to use it properly to ensure the integrity of the data captured.

Memory capture tools are often used in digital forensics investigations to collect volatile data from a running computer system. This volatile data can include running processes, network connections, system configuration settings, and user activity. Memory capture tools work by creating a copy of the computer's RAM, which is stored as a file on disk. This file can then be analyzed using memory analysis tools to extract information relevant to the investigation.

The choice of memory capture tool will depend on a number of factors, including the type of system being investigated, the purpose of the investigation, and the skill level of the investigator. Some memory capture tools are free and open-source, while others require a commercial license. Some tools are designed to be used by novice investigators, while others require advanced technical skills to use effectively.

Memory capture tools can be run in a number of ways. Some tools require installation on the target system, while others can be run from a USB drive or over a network connection. In some cases, the memory capture tool may need to be run in a specialized environment, such as a pre-boot environment or a live CD/DVD.

It is important to note that memory capture tools can potentially alter the state of the system being investigated. This is because the act of capturing memory requires the tool to interact with the computer's RAM. To minimize the risk of altering the system, it is important to choose a memory capture tool that is known to be reliable and to follow proper procedures for using the tool.

In summary, memory capture tools are essential for digital forensics investigations, as they allow investigators to collect volatile data from a running computer system. There are many factors to consider when choosing a memory capture tool, and proper procedures must be followed to ensure the integrity of the data collected.

FTK IMAGER

Access Data FTK Imager is a well-known forensic tool. Besides allowing an investigator to mount and image physical disk or logical partition, create a custom content image or export specified files from evidence, it allows us to capture memory from a running system. This tool must be run with privileges of the local Administrator. Using a portable version of FTK Imager running from a USB stick, we create the RAM image by selecting "Capture Memory" from the top-left dropdown menu. Do not copy your FTK Imager executable files onto the machine that is being investigated.

Including pagefile can be beneficial to our investigation as this file contains memory sections that have been "paged out" to the disk. Although it lacks structured data like (kernel) memory structures contained in RAM image, pagefile can provide valuable information about "raw" data present in the memory. For example, the IP address of a C2 server or malicious domain names found in pagefile.sys which can give us a clue about adversary activities happening on the system, even if it does not give us precise context.

The output format from FTK Imager is by default ".mem". If desired, we can choose to store memory in .ad1 format. The default format works well with most of the memory parsing and analysis tools, including the ones discussed in this series: Rekall and Volatility.

When selecting the location to write the output capture to, remember that we must not point it to the device we are imaging. Configure FTK Imager to store the resulting image on an attached external media, brought in by an examiner.
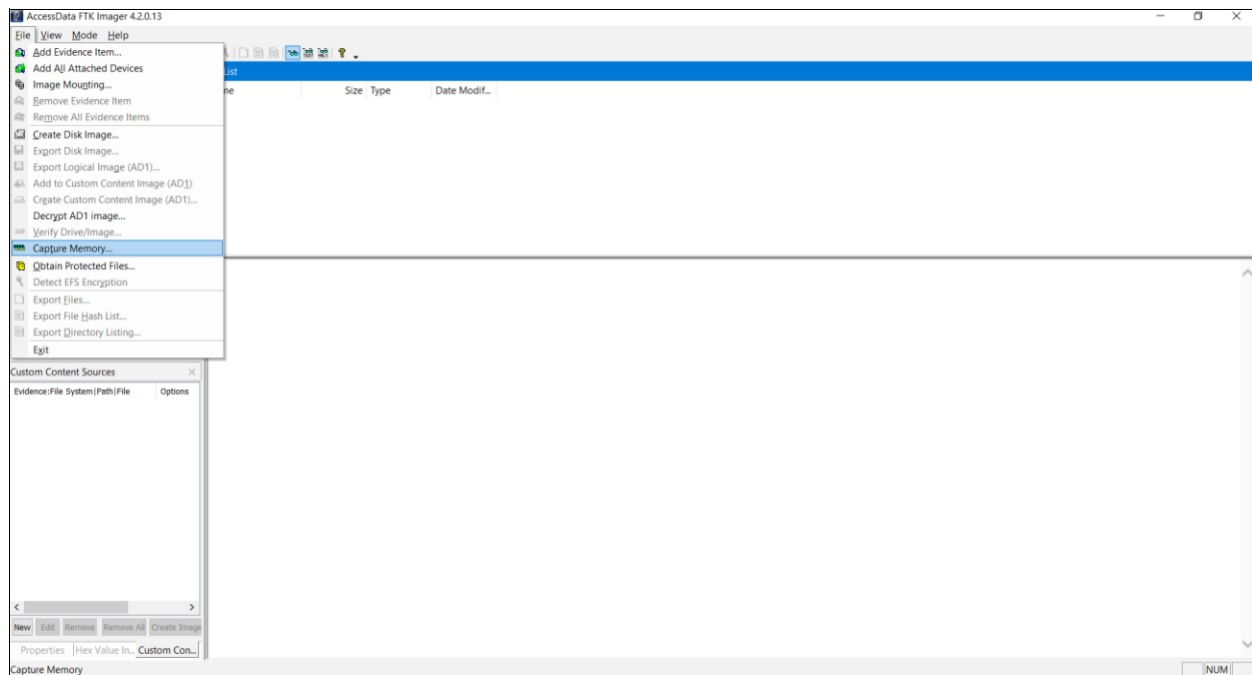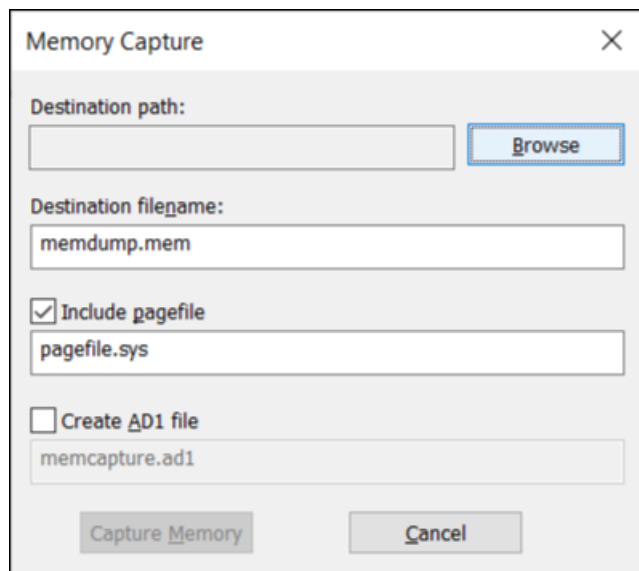

Figure 1: FTK Imager

Figure 2: FTK Imager - Memory Capture Dialog

WINPMEM

WinPMEM is another well-known tool used for obtaining memory images from Windows systems. It is an easy-to-use command line program which, by default, stores the resulting memory image in AFF4 format. AFF4 is an open-source standard which is

designed to store multiple types of data – including disk and RAM images. For subsequent analysis we may need to extract raw memory from acquired AFF4 files. WinPMEM allows for such extraction, as well as Volatility's and Rekall's imagecopy plugins.

```
winpmem_v3.3.rc3.exe -o E:\memdump.aff4
winpmem_v3.3.rc3.exe E:\ memdump.aff4 -ExtractPhysicalMemory -
o E:\memdump.raw
```

Capture memory in raw format:

```
winpmem_v3.3.rc3.exe --format raw -o E:\memdump.raw
```

UNSTRUCTURED ANALYSIS

Runtime information about a live system is stored in the computer's memory in a structured way. Kernel structures have a defined format and store data about running processes, handles, DLLs, other information that is stored about drivers, console content, network packets etc. But what if we are not overly concerned in all the information the operating system needs to keep the system up and living? What if we focus on some clue itself, not on the context surrounding it? If that were to be the case, unstructured analysis then comes into place.

STRINGS SEARCH & GREP

Strings is a well-known tool pre-installed on most Linux/UNIX-based OS. Strings searches through the content of a disk/directory/file and prints character arrays that appear to be (human readable) strings. As basic as it is, strings is a powerful tool to search for malicious domain names, IP or email addresses, filenames, contacts etc.

By default, strings outputs sequences of 4+ characters in length, formatted as 8-bit ASCII chars. Using switches, we can override default settings and search for big *(-e b)* and little *(-e l)* endian Unicode strings with a minimal length of our choice *(-n XX)*. Another hack to remember is printing the offset within the file where the string was found. For decimal or hexadecimal offsets, use *-td* or *-tx*, respectively. Finally, the *-f* switch comes in handy in case we need to perform a string search on a folder or disk image: it will include a filename where the string was found into the output.

In most cases, strings output becomes quite lengthy. Consider redirecting the output into a file or piping it to the *less* command. When looking for a specific string or pattern, we can use GREP/EGREP for postprocessing. GREP scans a file and prints out any matching strings. Use *-i* option to make the search case-insensitive, or *-v* to show only lines that
do not match the specified term. You should consult GREP's man page before using it to reveal the true power of this tool. EGREP, or Extended GREP, supports extended regular expressions search. Finally, AGREP a.k.a. Approximate GREP, allows for using various

queries, arbitrary wildcards, regular expressions – virtually everything that can be used with GREP, but more general. (Note that tools are not fully compatible.) The processing of AGREP is usually faster, compared to the same query performed by GREP. Additionally, AGREP has several built-in searching algorithms. Also, for processing each query, it will select the best-suited to complete the task.

Of course, GREP can be used not only to scan STRINGS output, but to scan an entire process's memory or any file or directory. However, if the file's encoding is different from ASCII, GREP will need some tweaks to fully cooperate with you   .

BULK EXTRACTOR

To go beyond the basic strings search and filtering, we will need a more sophisticated instrument. Bulk Extractor is a powerful tool which scans through the file or image (not limited to memory dump, crash dump or hibernation file) without parsing the filesystem structure. The output consists of so-called feature files with data extracted. These features include MAC and IP addresses, domain names, phone numbers, e-mail addresses, websites visited and even network packets that were in memory at the time of acquisition. When multiple of these information types are collected, histograms become populated. Histograms can be of great use: for example, we can infer that the most frequently hit MAC addresses and the topmost used email account belong to the device being investigated and to the nearest network switch/access point.
Bulk Extractor does not run every scanner by default. Scanners can by disabled or enabled to address specific requirements of the ongoing investigation. For example, information scanners such as Outlook and Facebook need to be directly invoked.

One of BulkExtractor's useful features, although not run by default, is the wordlist scanner. It works similarly to STRINGS – it looks for character arrays and saves them to the output feature file. Carving features include jpeg, rar or zip extraction, as well as previously mentioned network packets. The latter allows an investigator to open pcap files in Wireshark and look at network communication – note that packets will not have timestamps included. This is quite logical – packets themselves do not carry any such information; timestamps are added by packet capturing tools when speaking about real-time network monitoring. If Bulk Extractor carves for packets in memory, no timestamp can be present or reasonably added.

Bulk Extractor comes with a graphical interface, BEViewer, to view results. After opening a XML file with the report, the investigator can go through any extracted data listed in the left pane. The middle region shows the precise location of the scanned file where the data were found. On the right side of the window, data are shown in context, in a hexadecimal view.
Not every piece of data is shown in BEViewer. For example, a pcap file is present in the output folder, but cannot be investigated in the GUI, with the same applying to MFT,

INDX or LogFile: these NTFS filesystem files (or their parts currently present in memory) are carved from memory and stored for further analysis, performed by the investigator.

Before we proceed, let us speak a bit about an image that we are about to analyze. It was captured from a system infected by malware known as Khalesi, which has been a part of the Kpot campaign. It exhibits multiple standard malware behaviors, such as network communication with C2 server, creating persistence via scheduled tasks, Run registry key, multiple files dropped on the disk etc. However, as it employs several checks to ensure that the sample is not being debugged/analyzed on a Virtual machine, it eventually crashed, so not all actions were performed and thus cannot be deeply analyzed. VT analysis:
https://www.virustotal.com/gui/file/4e87a0794bf73d06ac1ce4a37e33eb832ff4c89fb9e4266490c7cef9229d27a7/behavior/Dr.Web%20vxCube

Example of running BulkExtractor on memory image:

```
$ bulk_extractor -o whitepaper/pony/bulk /mnt/hgfs/AnalystVMShare/memdump-
pony-2.mem
bulk_extractor version: 1.6.0-dev-rec02
Hostname: siftworkstation
Input file: /mnt/hgfs/AnalystVMShare/memdump-pony-2.mem
Output directory: whitepaper/pony/bulk
Disk Size: 9126805504
Threads: 2
Attempt to open /mnt/hgfs/AnalystVMShare/memdump-pony-2.mem
 9:13:04 Offset 67MB (0.74%) Done in  0:13:25 at 09:26:29
 9:13:06 Offset 150MB (1.65%) Done in  0:07:37 at 09:20:43
 9:13:07 Offset 234MB (2.57%) Done in  0:05:37 at 09:18:44
 9:13:08 Offset 318MB (3.49%) Done in  0:04:33 at 09:17:41
 9:13:09 Offset 402MB (4.41%) Done in  0:03:51 at 09:17:00
 9:13:10 Offset 486MB (5.33%) Done in  0:03:22 at 09:16:32
 ...
 9:18:49 Offset 8875MB (97.24%) Done in  0:00:09 at 09:18:58
 9:18:51 Offset 8959MB (98.16%) Done in  0:00:06 at 09:18:57
 9:18:57 Offset 9042MB (99.08%) Done in  0:00:03 at 09:19:00


     (timeout in 60 min.)
All Threads Finished!


*******************************************
** bulk_extractor is probably CPU bound. **
 **    Run on a computer with more cores    **
 **       to get better performance.        **
*******************************************
MD5 of Disk Image: fca945d20e5a1222672c98879959cd45
Phase 2. Shutting down scanners
Phase 3. Creating Histograms
Elapsed time: 365.561 sec.
Total MB processed: 9126
Overall performance: 24.9666 MBytes/sec (12.4833 MBytes/sec/thread)
Total email features found: 1099

analyst@sift:~$ BEViewer &
[1] 20920
analyst@sift:~$ Bulk Extractor Viewer Version 1.6.0-dev-rec02
ReportSelectionManager.setReportSelection image file:
/mnt/hgfs/AnalystVMShare/memdump-pony-2.mem, features file: null,
selection type: None
```

After the memory image has been processed, run BEViewer and point it to the report.xml file included in output directory. We are now able to go through the extracted artifacts, perform searches etc. In the picture below, domain_histogram is reviewed. The results

depict the domain described in public reports has been found in the memory image. However, this cannot be taken as evidence that the domain has indeed been visited. If the domain were present in a file loaded into memory, it could have been carved out by BulkExtractor without visiting it. Nevertheless, if the domain were to be malicious at the time of taking the memory image, it is probable that the system has been compromised.
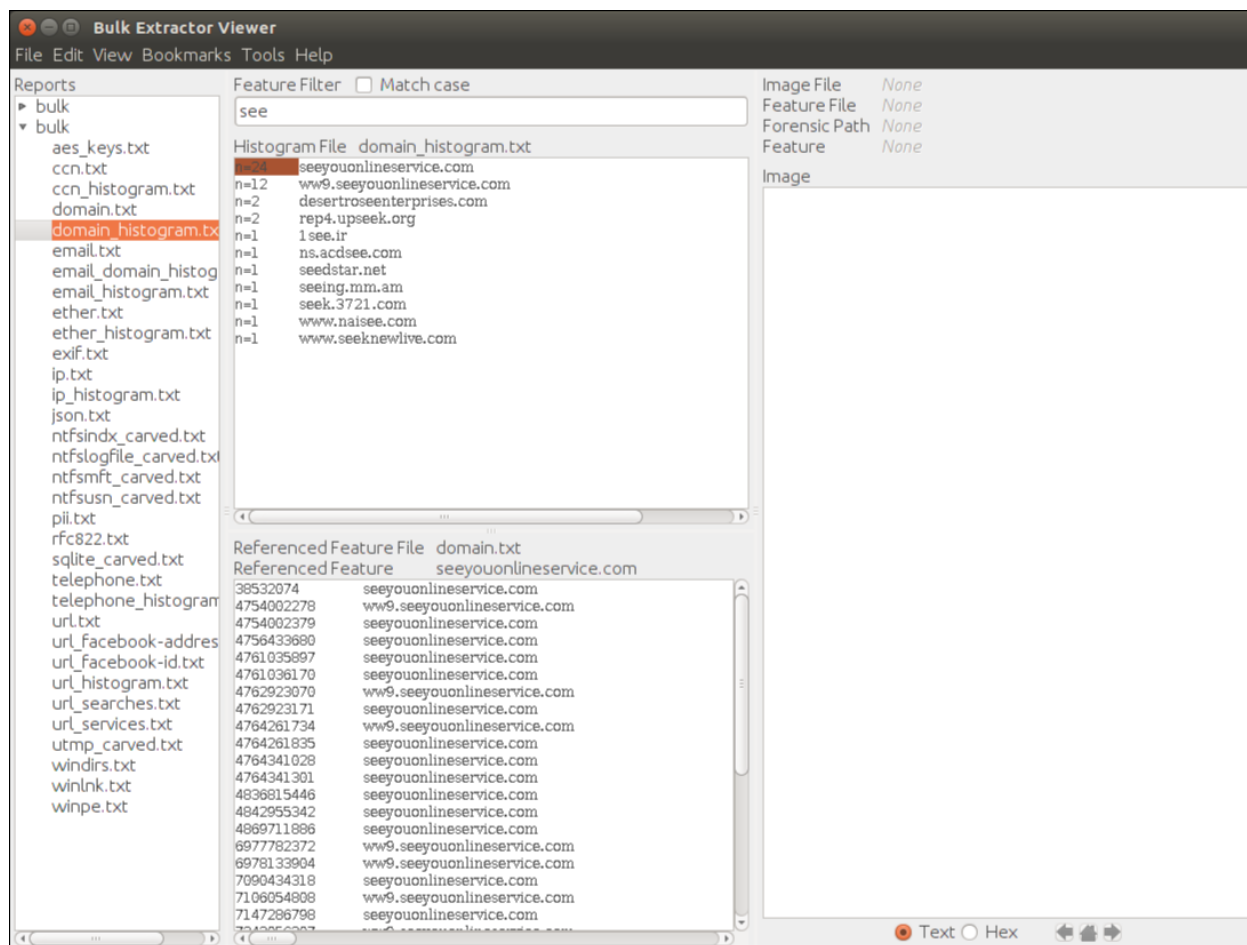

Figure 3: BEViewer GUI

Besides data shown in the GUI, BulkExtractor provides additional output. As mentioned previously, the tool carves network packets from the memory and these can be subsequently analyzed in WireShark or other packet viewers of the analyst's choice. In this case, we have searched for DNS requests for the above-mentioned domain with no luck. This is not at all evidence that the domain has not been contacted, only that the packet may have been overwritten in the memory by the time we managed to acquire it. Only one of the packets from C2 communication was left in the memory, TCP with FIN-ACK flags set – so most likely the last one left from that part of the communication.
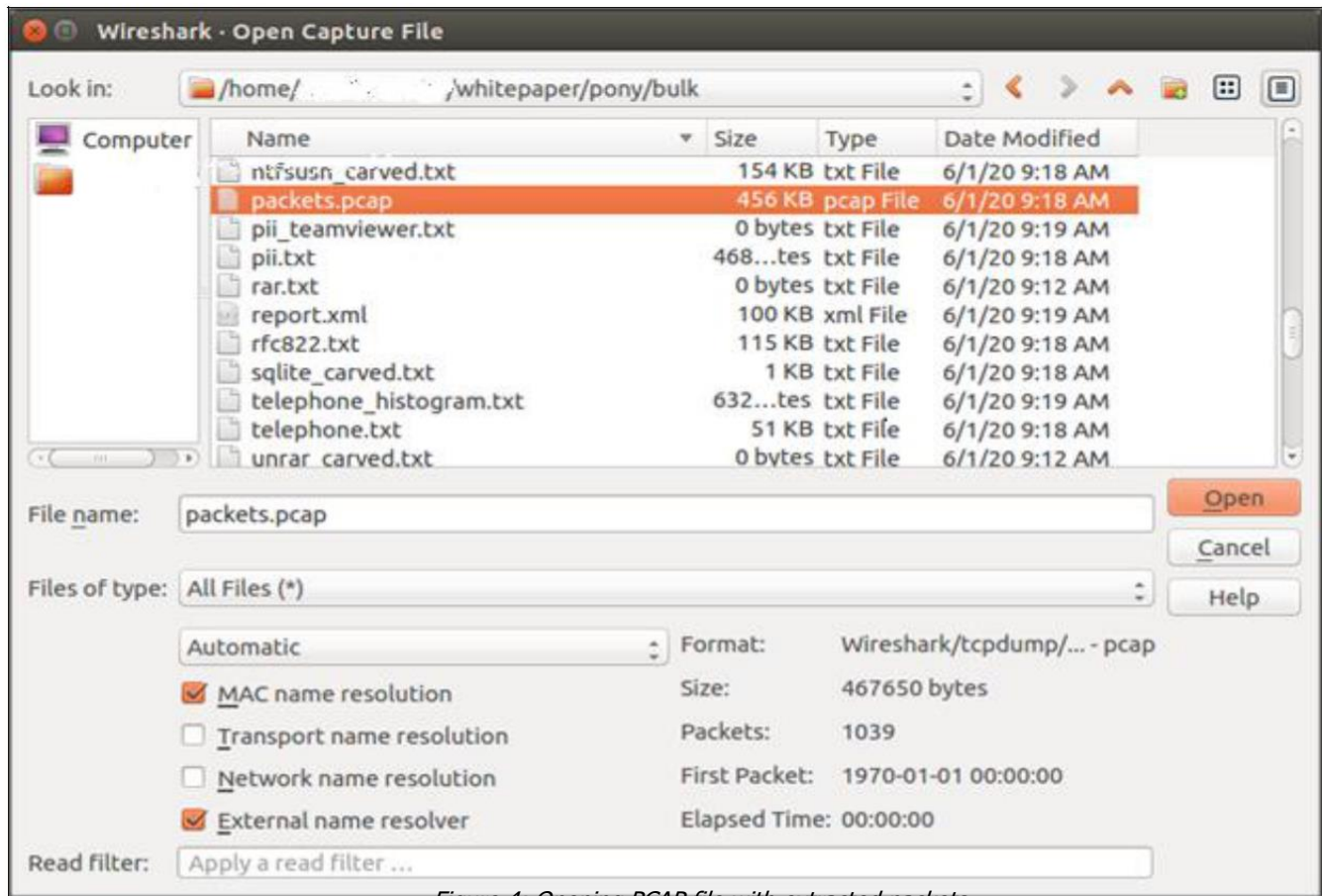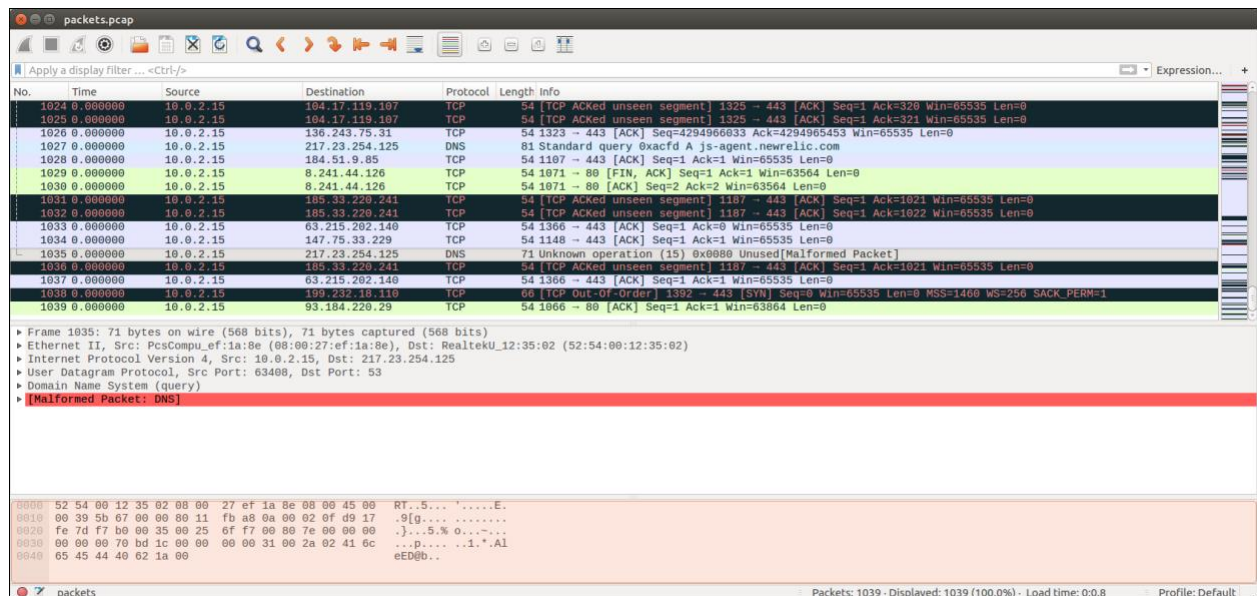
Figure 4: Opening PCAP file with extracted packets.

Figure 5: Wireshark

CONCLUSION

In Part I of *Windows Memory Forensics* series, we have covered several topics. First, how to acquire a memory image. We have introduced important functions of FTK Imager and WinPMEM, both of which are well-known tools for memory acquisition. Then, we described how to analyze memory in an unstructured way. We have also provided basic information on strings search. Lastly, we have discussed Bulk Extractor, along with its operation and features.

In the next part of *Windows Memory Forensics* series, we will start with the structured analysis of Windows memory images, using Volatility.

In conclusion, memory capture tools are an important component of digital forensics investigations. They allow investigators to collect volatile data from a running computer system, such as running processes, network connections, system configuration settings, and user activity. This data can be critical in identifying and analyzing potential evidence in a digital forensic investigation.

There are many memory capture tools available, each with its own features and capabilities. Some are free and open-source, while others require a commercial license. The choice of memory capture tool will depend on a number of factors, including the type of system being investigated, the purpose of the investigation, and the skill level of the investigator.

It is important to follow proper procedures when using memory capture tools, as they can potentially alter the state of the system being investigated. Choosing a reliable tool and following proper procedures will help to ensure the integrity of the data collected.

Overall, memory capture tools are a valuable resource for digital forensics investigators, allowing them to extract critical information from a running computer system that might not be available through other means.