Memory-based Malware Detection

Jessica Soto
Illinois Institute of Technology
201 East Loop Road
Wheaton, Illinois
jsoto9@hawk.iit.edu

Peter Stanczak
Illinois Institute of Technology
201 East Loop Road
Wheaton, Illinois
pstanczak@hawk.iit.edu

Manav Nagla
Illinois Institute of Technology
201 East Loop Road
Wheaton, Illinois
mnagla@hawk.iit.edu

Pablo Jusue
Illinois Institute of Technology
201 East Loop Road
Wheaton, Illinois
pjusue@hawk.iit.edu

ABSTRACT

This section of the paper is a brief introduction to the paper, so it's length will be 1/5 or 1/4 of the page.

This paper provides an analyze of the basic memory forensic methodology which is used to detect malware which is running the RAM.Magnet AXIOM and the volatility framework capabilities will be investigated but also there are going to be studied different tools and procedures beyond the Magnet AXIOM to do the forensic analysis.

A malware study will be done with the aim of identify the malware type which is more suitable for this project.Based on the studies which have been named previously a initial lab environment will be set up, which will consist in a three virtual machines which are going to run Windows systems. Two of them will be used to capture the RAM malware and the third one will be use to analyze it.

Categories and Subject Descriptors

H3.4 [[**Systems and Software**] **Performance evaluation (efficiency and effectiveness)**]: Miscellaneous

General Terms

Technology, Cybersecurity, Computer Science, Malware, Information

Keywords

Malware, Memory, Analysis, Forensic

1. INTRODUCTION

In this section we will make a brief introduction about what the project covers and what is the main aim of it. The introduction will be a 1/5 to 1/4 of the final paper.

Most of the tools which usually are used to detect malware are designed to search for malware in the hard disk. However, in this project we are going to focus on the Random Access Memory (RAM) malware and techniques to detect it.

This analysis will be developed in a Windows computer, in which we will analyze the system in two situations: With and without malware.

This will let us know if our system success in detecting malicious modifications of the data which is stored in the memory. Magnet AXIOM and the volatility foundation are the tools which will be used to perform this analysis. As we can see in the malware classification point we will analyze the different types of malware so we can select the one which is better for our system. As we will explain the best malware for check our tools is a trojan which have been designed to attack a 64-bit windows.

The methodology we are going to follow can be seen in the following point, starting by identifying the most suspicious process which are in our system and the analyze all the activity of this process so we can detect if it is a malicious process or is only a regular one.

Finally, we will check the results with Virus Total so we can know if we have detected the malware correctly.

2. MEMORY FORENSIC METHODOLOGY

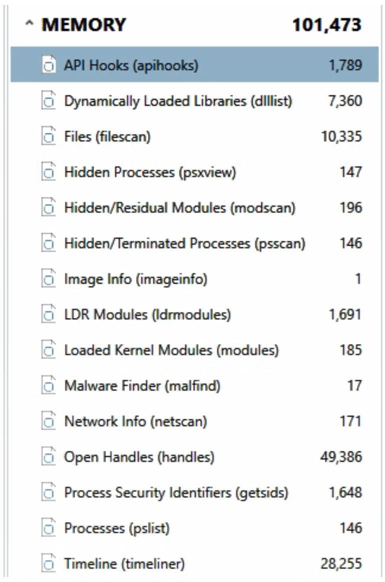
In this section we will explain the methodology we are going to use for detecting software which could be malicious and analyze it deeply, so we can discard no-malicious software from the one which is malicious. In the first place we are going to take images of the memory of the system. For this step we will use two machines, a Windows 7 and a Windows 10, to take the images and a Windows 10 machine, call investigation machine in which we are going to perform the analysis of the memory.

The first step as we have said above is taking a picture of the processes which are running in RAM. This will be done by using the MAGNET RAM capture tool. This is a free tool which takes a picture of the RAM and storage it as a .raw format.

When we have the images of the systems, the Windows 7 and the Windows 10 machines, we will be uploaded it to our investigation machine. In this machine we will start using the MAGNET AXIOM PROCESS.

This tool takes the .raw file we have created and create a case to process the evidence of it.

When the image is processed the MAGNET EXAMINE will be used to check all the information which have been processed by the PROCESS tool. MAGNET EXAMINE will run the volatility's functions to show all the information. All the information processed by MAGNET AXIOM is available by using the volatility open source software by the command line.



MEMORY	101,473
API Hooks (apihooks)	1,789
Dynamically Loaded Libraries (dlllist)	7,360
Files (filescan)	10,335
Hidden Processes (psxview)	147
Hidden/Residual Modules (modscan)	196
Hidden/Terminated Processes (psscan)	146
Image Info (imageinfo)	1
LDR Modules (ldrmodules)	1,691
Loaded Kernel Modules (modules)	185
Malware Finder (malfind)	17
Network Info (netscan)	171
Open Handles (handles)	49,386
Process Security Identifiers (getsids)	1,648
Processes (pslist)	146
Timeline (timeliner)	28,255

Figure 1: Volatility functions

There are several volatility functions which help us with the analysis of the processes. The Axiom framework has a graphic view of the hierarchy of the processes which were running on memory. However, it is not quite easy to evaluate if there is any suspicious process using this GUI, so we are going to use the volatility command line to execute the function pstree so we can check if there is any suspicious process. First, we are going to check the critical Windows processes of the system which are shown in the following picture.

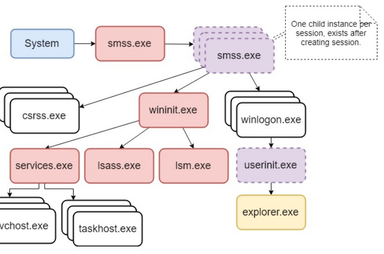


Figure 2: Windows Critical processes

If everything looks correct, we will examine the processes looking for suspicious processes like, the ones which have spelling mistakes, so their names are like authorized processes ones. We need also to identify all the hierarchy of this suspicious processes because we may have more than one suspicious process which we need to analyze. For example, a process which is running a system software like the notepad from another path which is not related to the system32 directory is very suspicious because that is not the path of the notepad software.

Now we will search for the network information of these processes, if they have any unauthorized connection against a foreign server.

When we have identified the processes, which are suspicious we are going to dump it to a file using Axiom Examine, so we can perform a more accuracy analysis of if they are malicious or not. There are several tools here which are going to help us to analyze these processes.

The first one is the IDA, which will allow us to analyze the assembler code of the process for searching for suspicious functions like functions which create persistent connections, backdoors which are hidden in the OS.

Also, we have the string command which will allow us to parse the file dumped to a String, so we can analyze it in a more easy way.

After all the analysis which we have performed we can state if a process is a malware or not but also, we are going to compares our results with the one which Virus Total is going to give us
Out methodology can be summarized in 6 points:

- Identify rogue processes
- Analyze process DLLs and handles
- Review network artifacts
- Look for code injection
- Check for rootkits
- Dump for further analysis

3. CAPABILITIES OF MAGNET AXIOM AND THE VOLATILITY FRAMEWORK.

The volatility framework allows to perform different test over a memory image which have been taken before. Magnet AXIOM is built based on volatility, so when AXIOM process a memory image uses volatility functions to perform the analysis of the memory. AXIOM allows the user to perform the forensic process by creating a case and its present all the data analyze by the Volatility functions in a organized way to the user. The suite of Magnet Axiom which is going to be used in this project is formed by the RAM capture, the AXIOM PROCESS and the AXIOM EXAMINE.

The RAM capture allows the user to capture an image of the size of memory which he selects, it can be full memory or a piece of it. Then this software creates an .raw file which can be processed by volatility and MAGNET AXIOM PROCESS.

MAGNET AXIOM PROCESS creates a forensic file and analyze all the evidences by using the volatility functions. Then it launches automatically the MAGNET EXAMINE to show the user the data obtained.

MAGNET EXAMINE shows the user all the data organized by how they were obtained. Each volatility function has a point with the information it has acquired.

4. TOOLS BEYOND VOLATILITY FRAMEWORK AND AXIOM:

4.1 Process Explorer

It is an application used to find out what registry key and other object are open on the system and what DLLs are the using. The DLLs as we have explained in our methodology can be used by the malware to create persistence. For this reason an application which can find out which DLLs process are loaded or opened. This is a microsoft tool which will work in our system so it will be very helpful for distinguish between that processes which seems suspicious but they are not of the ones which actually are. Also if we detect any suspicious activity been performed by a process which can kill it using this software which also have an intuitive GUI for handling all its features.

4.2 Process Monitor

It is an aplication which monitor files and registry processes in real time. Process monitors have huge amount of powerful monitoring and filtering capabilities like non-destructive filters which allows the users set filters without losing data or process tree which shows the relationship referenced in a trace. Also because it captures the thread stacks for each operation sometimes is possible to identify the root cause of an operation. Monitoring the process is the first step in our methodology, so tools which allows us to perform this task in an easy way are very helpful for this project. This is a microsoft tool so it runs over a Windows system like the one we are going to use.

4.3 Psfile

It is an application which is helpful to determine a list of files that are opened remotely. PsFile allows the user to close opened files by name or by identifier. This is will be very helpful for our project because when a malware is detected, it also important to close all the files or connections which it is handling. With this tool we can use the IDs extracted from the other tools and use them to close the files.

4.4 MD5sums

It is an application which generates hashes for file integrity verification. MD5sums calculates the MD5 message digest for one or more files. This allows the user who is downloading a file to know if the file is free from malware by using this tool to compare the MD5 sent by the sender which the one calculated in the users system if they have the same value the file has not been modified and no malware have been introduced.

4.5 NMap

Nmap is a free open source utility which is used for network discovery and security auditing. NMap is based in the use of IP packets for determine if a host is available in a network, it also allows to audit the services which the hosts are offering, operating systems and more features. This tool will be very useful for the step described in the methodology in which when a process have been detected as suspicious, we will check their connections or the services which they are offering. This tool is a basic tool in every security analyzer system because it allows to perform so many different functions in an easy and full documented way. It is a tool which is available for all the operating systems so will work in our Windows machine.

5. MALWARE TAXONOMY AND CLASSIFICATION

In this section we will analyze the different types of malware according to different There are different categories in which we can classified the malware based by their malicious features, the platforms for which they were developed, the infection vectors or other malware features. According to their malicious features we distinguish between:

- Virus which is a malicious software which replicate by itself.
- Worm which not only replicate by itself it is also a standalone malware.
- Backdoor which allows unauthorized access to compromised computers
- Exploits which attacks a software vulnerability to gain authorized access
- Backdoor which allows unauthorized access to compromised computers
- Exploits which attacks a software vulnerability to gain authorized access
- Trojan which is non-replicating but it is a deceiving software which a hidden functionality.
- Hacktool which is a tool for attacking and scanning.
- Rootkit which is an actively hiding software.
- Spyware which invades the user privacy.

For our project we are going to select trojans as the ones which are going to use because they have characteristics which our system can analyze following the methodology which have explained in the last point.

The malware usually is created for a determined platform and does not work in other systems. The operating system of the machine which the malware is going to attack. Most of the malware which is designed for windows systems. We will work with a 64-bit Windows platform so the trojan we will select is a W64.

Another way to classify them will be by their infection vectors. Malware can attack different parts of the system, for example, network attacks like network worms which attack through a vulnerability, bluetooth attack or file infections. However, we will going to study the ones which performs their attacks using memory. Malware can be evaluated also by analyzing other features like polymorphic or metamorphic which are viruses which can modify their code by themselves.

6. INITIAL LAB ENVIRONMENT

Our lab environment is divided in three different VMware machines. The first one is the investigation machine in which we will perform the analysis of the memory images taken of the other two machines. The test machines are a Windows 10 and a Windows 7 machine which data will be deleted each time we sign out. This will help us to perform test in a new machine without mixing different malwares. The features of these machines are:

Investigation machine Windows 10

- RAM: 4GB (or the minimum windows support)
- HDD: 64 GB

Windows 10

- RAM: 8GB
- HDD: 128 GB

Windows 7

- RAM: 4GB (or the minimum windows support)
- HDD: 64 GB

There will be a share folder between the three machines, so we can storage the memory images taken in the infected machines and analyze them in our investigation machine. The software installed in the investigation machine is the suite of MAGNET, MS Office and the volatility framework

7. REFERENCES

[1] ACM computing classification system 1998
`https://www.acm.org/publications/computing-classification-system/1998/`

[2] Hunting malware with memory analysis
`https://technical.nttsecurity.com/post/102egyy/hunting-malware-with-memory-analysis`

[3] Process explore tool
`https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer`

[4] Process monitor tool
`https://docs.microsoft.com/en-us/sysinternals/downloads/procmon`

[5] PsFile tool
`https://docs.microsoft.com/en-us/sysinternals/downloads/psfile`

[6] MD5sums tool
`http://www.pc-tools.net/win32/md5sums/`

[7] Non-malware RAM software database
`https://www.nist.gov/software-quality-group/national-software-reference-library-nsrl`

[8] Windows processes
`https://securitybytes.io/blue-team-fundamentals-part-two-windows-processes-759fe15965e2`

[9] Digital forensic poster
`https://digital-forensics.sans.org/media/poster_2014_find_evil.pdf`

[10] Creating a baseline of process activity for memory forensics
`https://www.sans.org/reading-room/whitepapers/forensics/creating-baseline-process-activity-memory-forensics-35387`

[11] Indicator of compromise in memory forensics
`https://www.sans.org/reading-room/whitepapers/forensics/indicators-compromise-memory-forensics-34162`

[12] Techniques and Tools for Recovering and Analyzing Data from Volatile Memory
`https://www.sans.org/reading-room/whitepapers/forensics/techniques-tools-recovering-analyzing-data-volatile-memory-33049`

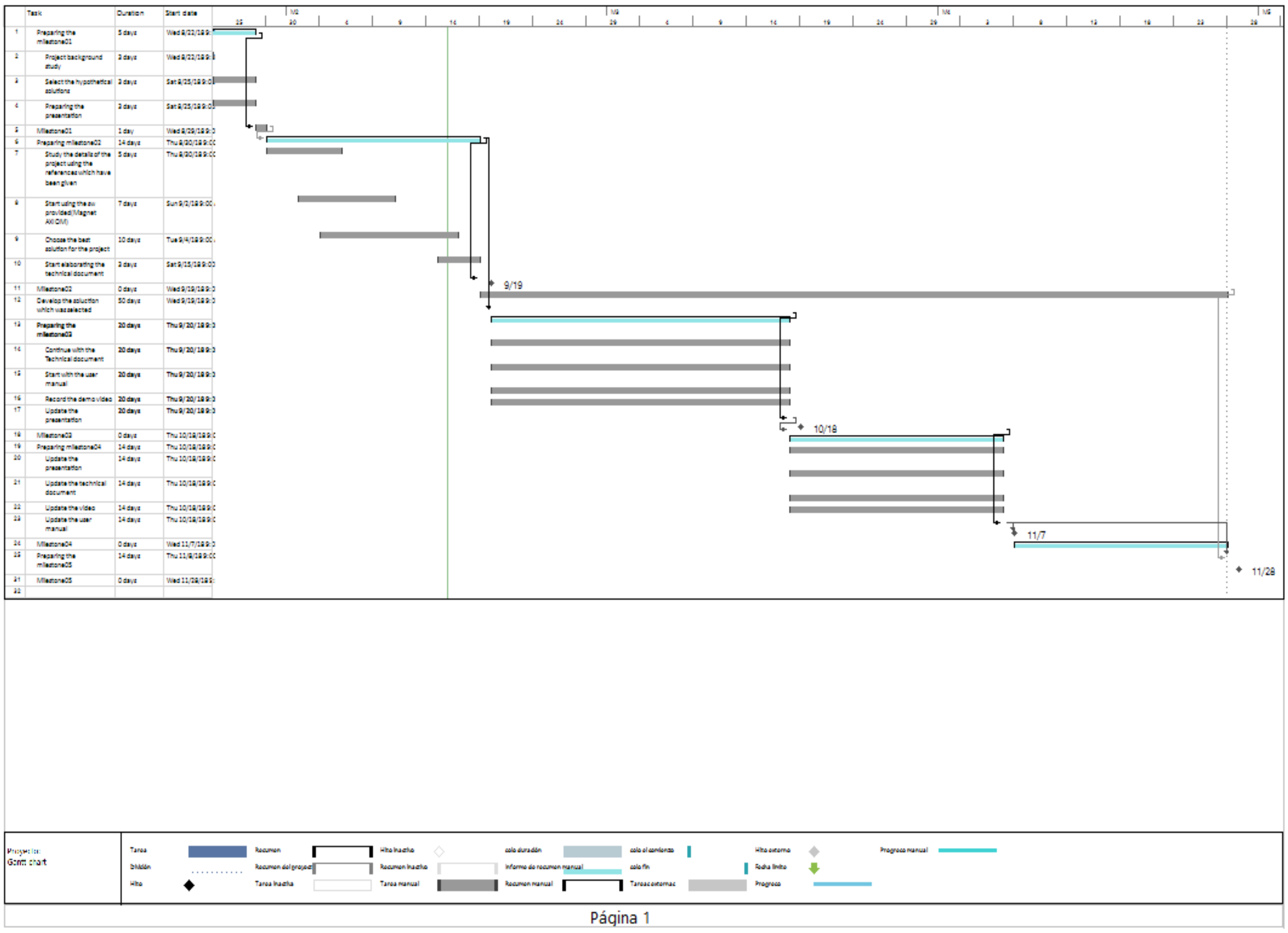
[13] Volatility foundation
`https://github.com/volatilityfoundation/volatility/wiki/`
`Samples`

[14] The art of memory forensics
`https://www.memoryanalysis.net/amf`

[15] Malware Analyst Cookbook
`https://www.amazon.com/Malware-Analysts-Cookbook-DVD-Techniques/dp/0470613033`

[16] NMap tool
`https://nmap.org/book/man.html`

8. APENDIX



	Task	Duration	Start date
1	Preparing the milestone01	5 days	Wed 8/22/18 9:
2	Project background study	3 days	Wed 8/22/18 9:
3	Select the hypothetical solutions	3 days	Sat 8/25/18 9:0
4	Preparing the presentation	3 days	Sat 8/25/18 9:0
5	Milestone01	1 day	Wed 8/29/18 9:0
6	Preparing milestone02	14 days	Thu 8/30/18 9:00
7	Study the details of the project using the references which have been given	5 days	Thu 8/30/18 9:00
8	Start using the sw provided(Magnet AXIOM)	7 days	Sun 9/2/18 9:00
9	Choose the best solution for the project	10 days	Tue 9/4/18 9:00
10	Start elaborating the technical document	3 days	Sat 9/15/18 9:00
11	Milestone02	0 days	Wed 9/19/18 9:0
12	Develop the soluction which was selected	50 days	Wed 9/19/18 9:0
13	**Preparing the milestone03**	**20 days**	**Thu 9/20/18 9:0**
14	Continue with the Technical document	**20 days**	**Thu 9/20/18 9:0**
15	Start with the user manual	**20 days**	**Thu 9/20/18 9:0**
16	Record the demo video	**20 days**	**Thu 9/20/18 9:0**
17	Update the presentation	**20 days**	**Thu 9/20/18 9:0**
18	Milestone03	0 days	Thu 10/18/18 9:0
19	Preparing milestone04	14 days	Thu 10/18/18 9:0
20	Update the presentation	14 days	Thu 10/18/18 9:0
21	Update the technical document	14 days	Thu 10/18/18 9:0
22	Update the video	14 days	Thu 10/18/18 9:0
23	Update the user manual	14 days	Thu 10/18/18 9:0
24	Milestone04	0 days	Wed 11/7/18 9:0
25	Preparing the milestone05	14 days	Thu 11/8/18 9:00
31	Milestone05	0 days	Wed 11/28/18 9:
32			

Página 1