Digital Forensics
RAM Analysis

Presented by
Christie Gross
RAM Analysis - Definition

- RAM capture is the process of capturing live memory from a running computer system. RAM analysis consists of performing forensic analysis on the data gathered from the live computer.

- After conducting a memory dump on any live machine to capture RAM, the memory image can be used to determine information about running programs, the operating system, and the overall state of a computer, as well as to locate deleted or temporary information that might otherwise not be found on a normal image.
Until recently, RAM analysis and capture was not a mandatory step in investigations, or even in triage situations where analysts were attempting to gather forensic data on site.

However, with new tools that allow entry into locked systems and with the growing importance of temporary files, RAM analysis is quickly becoming a pivotal and mandatory part of the digital forensics process.
Volatile memory access (live forensics) is useful in law enforcement situations where data would be lost by powering off a suspect machine.

- The longer a machine is off, the more data becomes lost.

- The following can be found using RAM capture: Processes, Network connections, Open files/Configurations/Encryption keys, Open/Active Registry keys, Exploit-related information, Zero-day attacks and rootkits, and Kernel-level structures.
A multitude of new tools have become available within the past decade (both paid and free), which serve to better aid forensic analysts in the capturing of RAM data.

- **FTK Imager** – Windows based program capable of capturing local drives and local RAM.

- **Volatility** – Kali Linux tool capable of analyzing RAM from a memory dump disk image.

- **Win64dd/MWMT DumpIt** – A program that can be installed on any Windows machine in order to capture/dump local RAM from that computer. This is oftentimes a tool of choice for triage forensics.
The first step of this tutorial consists of locally capturing RAM data using **FTK Imager** and/or **Win64dd**. This would be done in a triage situation in the field.

The second step consists of looking at the captured RAM data to view the output of the RAM image file in order to gather additional information from a system.
This tutorial will also cover analyzing captured RAM with volatility on a Kali Linux machine. Please note that to properly view converted RAM folder contents instead of simply the original bits, a more complex, paid program such as EnCase will be required.
Beginning the Tutorial


- Please download the memdump.mem file for Kali Linux from: [https://samsclass.info/121/proj/p4-Volatility.htm](https://samsclass.info/121/proj/p4-Volatility.htm)
Forensics Tutorial 10 – RAM Analysis

Explanation Section

RAM Capture/Analysis – Definition
RAM capture is the process of capturing live memory from a running computer system. RAM analysis consists of performing forensic analysis on the data gathered from the live computer. The process of capturing the memory is known as dumping. The tools and methods of dumping the memory on a running computer differ with the operating system. After conducting a memory dump on any live machine to capture RAM, an analyst can use the memory image to determine information about running programs, the operating system, and the overall state of a computer, as well as to potentially locate deleted or temporary information that might otherwise not be available with a normal forensic image.

RAM analysis has become increasingly important in the scope of forensic investigations. Until recently, this was not a mandatory step in investigations, or even in triage situations where analysts were attempting to gather forensic data on site. However, with new tools that allow entry into locked systems and with the growing importance of temporary files, RAM analysis is quickly becoming a pivotal and mandatory part of the digital forensics process. It is necessary to have knowledge of how to conduct a proper RAM analysis in the digital forensics workplace.

RAM Capture and Analysis Tools
A multitude of new tools have become available within the past decade (both paid and free), which serve to better aid forensic analysts in the capturing of RAM data. This tutorial will cover some of the free tools available within Windows and Kali Linux systems. This tutorial will cover the following tools:

- **FTK Imager** – Windows based program capable of capturing local drives and local RAM.
- **Volatility** – Kali Linux tool capable of analyzing RAM from a memory dump disk image.
- **Win64dd/MWMT DumpIt** – A program that can be installed on any Windows machine in order to capture/dump local RAM from that computer. This is oftentimes a tool of choice for triage forensics.

In the scope of this tutorial there will be several steps. The first consists of locally capturing RAM data, while the second consists of looking at the captured RAM data to view the output of the RAM image file. This tutorial will also cover analyzing captured RAM with volatility on a Kali Linux machine. Please note that to properly view converted RAM folder contents instead of simply the original bits, a more complex, paid program such as EnCase will be required.

In This Tutorial
This tutorial will examine the process for capturing RAM in Windows with FTK Imager and Win64dd/MWMT DumpIt. It will then go over the process of loading the resulting files into Forensic Toolkit for analysis to see what the RAM capture looks like. We will then view the DumpIt raw image file created on the Windows 8 machine in volatility on Kali Linux.
LEARNING OBJECTIVES:

- Capture RAM with FTK Imager, and view resulting RAM data in FTK
- Capture RAM with DumpIt(Win64dd) and view the resulting file in FTK
- Examine a sample RAM dump in the volatility tool

Part 1 – Capture RAM with FTK Imager

1. Login to the Virtual Lab website (https://v5.unm.edu/cloud/org/ialab), and enter the ‘NEST Digital Forensics – Part 3 vApp’. Click on the Windows 8 machine to open the VM.

2. At the login screen of the Windows 8 machine use the password letmein.

3. Launch AccessData FTK Imager from the Windows desktop. Navigate to File>>Capture Memory. This will launch a new window with options for capturing live memory on the local machine.

4. The resulting window will ask you to select a destination path for the memory dump. This file can be placed anywhere – for the purposes of this tutorial, a folder called Memory Dump has been created on the Desktop. It is also possible to name this file in the Destination Filename field. It is possible to
choose to include the pagefile. A pagefile is a reserved portion of a hard disk that is used as an extension of random access memory (RAM) for data in RAM that hasn’t been used recently. For the purposes of this tutorial we have elected to include the pagefile in the memory dump. In this tutorial we are not creating an AD1 file, but this is an available option. Click **Capture Memory**.

5. The memory capture will begin. A progress bar will display while the capture is being completed. You can also view the destination to which the memory is being dumped, as well as the status of the dump itself.
6. Once the memory dump has been completed, the open window will state whether or not the memory dump completed successfully. Close the open memory dump window. Inside the folder to which you saved the memory dump, there should be two files, `memdump.mem` (or whatever you named it), and `pagefile.sys`. These two files contain all information that was present in memory when the memory dump was processed. In the scope of an investigation, this data can include passwords, temporary Internet files, deleted files that have not yet been written over, etc., and can be pivotal in creating a timeline of actions for a potential suspect, and in finding evidence that may not otherwise have been located.
7. Load the file into Forensic Toolkit. The steps are precisely the same as for creating a case and adding an evidence file. These steps can be found in Tutorial 4 – Viewing Image Contents with FTK. The memory dumped file will be displayed as a system file evidence type within FTK. From this point on you should be able to view the contents of the memory file. In this case you will only see slack/free space. However, there is information stored within these areas that can be recovered and used as evidence.
8. Click on **DriveFreeSpace02** in the **Explore** tab. We will see that there is data written within this area of slack space. Temporary information about processes and programs being run at the time of the capture may be found within this information. It is also possible to find temporary Internet files and other information not available by simply imaging the disk. Within this particular section of slack space we can see that it contains information about the system drivers and information about the operating system including BIOS version, System Manufacturer, etc.

Note if we would like to better view the information in a more user-friendly mode instead of simply viewing the raw bits, there are paid programs that can do this. EnCase has this capability.
9. This image file can be treated like any other: that is, it can be hashed and analyzed, and data found within RAM can be used as evidence. When looking at RAM data, be sure to check for temporary files such as Internet data or information about malicious programs running processes within RAM. Be sure that when running a memory dump that you do not restart the machine before attempting to dump the memory – this will result in most everything being wiped from RAM, as RAM is not saved when a computer is shut down and restarted. This is what makes memory dumps ideal for forensic triage – if analysts can have access to a computer in an on or used state in the field, much more data can be captured than if they were to simply shut it down and take it back to the forensic lab.

Part 2 – Capture RAM with Win64dd/MWMT DumpIt

1. MWMT DumpIt is the replacement program for Win64dd produced by Moonsols. This program comes in both a free and paid ($190/person) version. DumpIt is a memory dumping program that many triage forensic analysis use in order to create quick RAM memory dumps in the field. It is then possible to analyze the resulting file in any forensic analysis tool, in this case Forensic Toolkit. For reference, DumpIt can be downloaded from this site: http://www.moonsols.com/windows-
memory-toolkit/. Note that the program is called Windows Memory Toolkit (MWMT) on the website.

2. On the **Windows 8 machine**, launch MWMT DumpIt from **Desktop>>Forensic Tools>>MWMT-v1.4**. A new program window will open that resembles the Windows command prompt.

```
C:\Users\ASM User\Desktop\Forensic Tools\MWMT-v1.4\DumpIt.exe

Dumplt - v1.3.2.20110401 - One click memory memory dumper
Copyright (c) 2007 - 2011, Matheiu Susiche <http://www.msuiche.net>
Copyright (c) 2010 - 2011, MoonSols <http://www.moonsols.com>

Address space size: 2147483648 bytes ( 2048 Mb)
Free space size: 14702796800 bytes ( 14021 Mb)

* Destination = \??\C:\Users\ASM User\Desktop\Forensic Tools\MWMT-v1.4\WIN
t-20140508-185618.raw

--> Are you sure you want to continue? [y/n] 
```

3. DumpIt will ask whether or not to continue with the memory dump. The destination location of the memory dump is also shown – the memory dump file will be located within the DumpIt folder and saved in a .raw format for analysis purposes. Enter ‘y’ in order to proceed with the memory dump. The memory dump will begin to run, designated by the ‘Processing’ text within the window. Once the memory dump has completed, the program window will state that the dump completed successfully and may or may not close automatically.
4. Navigate to the C:\Users\ASM User\Desktop\Forensics Tools\MWMT-v1.4 location, and you will see the created raw file WIN8-0-20140508-185618.raw (the file name will change depending on the system type and date and time of the dump. You can move this file into the dumps folder for ease of use if desired.

5. Following the steps used in the above section, create a new case within Forensic Toolkit, and add the memory dump as the evidence file. It should look similar to the memory dump generated with FTK Imager. Note that the data found in RAM will be slightly different, as the process you are running is different (DumpIt vs. FTK Imager). Although this affects RAM, as long as this change in RAM is explained within the investigation this small alteration in RAM is generally allowed.
Part 3 – Viewing a RAM Dump Disk Image with Volatility in Kali Linux

1. Login to the Virtual Lab website (https://v5.unm.edu/cloud/org/ialab), and enter the ‘NEST Digital Forensics – Part 3 vApp’. Click on the Kali Linux machine to open the VM.

2. At the login screen of the Kali Linux machine use the username root and the password letmein.

3. Open a new terminal window and enter the following commands in order to launch the volatility shell with available Python-based features.

   
   cd /usr/share/volatility
   python vol.py -h

   
   Upon launching volatility, a new terminal will open. This terminal will show various options for using volatility. Volatility is singularly for RAM memory analysis, and does not offer a way to actually perform a memory dump. Note that volafox (another RAM tool in Kali Linux) does the same thing, but for Mac RAM.
4. Note that a sample memory dump named `memdump.mem` has been placed on the desktop. The path to this file is `root/Desktop/memdump.mem`. We will be examining this RAM dump in volatility. There are a multitude of commands that can be run from volatility to analyze certain types of data located in RAM, including SAM login data (password hashes), local system information, the running processes at the time of the memory dump, console commands recently executed within the system, and services running at the time of the memory dump.

5. Enter the command following commands to find basic data about the machine on which the memory dump was conducted.

   ```
   python vol.py imageinfo -f /root/memdump.mem
   ```

   This command gives you several pieces of information. At this point, we only need to know the profile type of the memory dump, in this case Win2008SP1x86. We will use this in the next few steps.
6. In order to get the SAM login data and the hash value for the login password, it is necessary to get the hive list so to find the starting location in memory of where the registry information resides.

   python vol.py hivelist --profile=Win2008SP1x86 -f /root/memdump.mem
7. We now have a list of where several key items are located in the memory dump. Next we will extract the password hashes from the memory dump. To do this it is necessary to find the starting memory locations for the system and SAM keys. Look in the dump and copy the numbers in the first column that correspond to the SAM and SYSTEM locations. Then output the password hashes into a text file called hash.txt.

```
python vol.py hashdump --profile=Win2008SP1x86 -f /root/memdump.mem -y 0x86226008 -s 0x89c33450 > /root/hash.txt
```

The hash values in hash.txt file can now be cracked with any password cracking tool.
8. In addition to gathering hashes for SAM data, it is also possible to view processes that were running at the time of the memory dump. Enter the following command in order to display the running processes.

   python vol.py pslist --profile=Win2008SP1x86 -f /root/memdump.mem

Note the following columns:
Offset: The location in RAM of the process, in hexadecimal
Name: The process name, as it would be shown in Task Manager
PID: The process ID
PPID: The parent process ID – that is, the process that launched this process.
9. It is also possible to see any recent console commands that have been executed. Though this may not always be useful, in this case we can see that there were some console commands recently run that contain information about user accounts recently created, along with the passwords that were set for those users. Run the following command to display this information.

```
python vol.py consoles --profile=Win2008SP1x86 -f /root/memdump.mem
```
10. Finally, it is also possible to see the services that were running at the time of the memory dump. This can be used to identify malware or other malicious programs running at the time of the memory dump. Run the following command to display the services.

```
python vol.py svcscan --profile=Win2008SP1x86 -f /root/memdump.mem | more
```
Conclusion

It is clear that there is a large amount of data that can be recovered from RAM that includes a large amount of data that could not otherwise be recovered from a regular forensic image performed on a hard drive. Not only can sensitive data such as password hashes and usernames be recovered in order to be cracked, but other data such as malicious or illegal programs and services that can be used as evidence when building a case file. RAM collection and analysis is quickly becoming a pivotal part of forensic investigations, and thus, it is important that all forensic investigators understand the data that can be gathered and how to properly gather this data.

Try This

1. Locate a sample RAM memory dump from an online resource and perform the above analysis in volatility within Kali Linux. Note any information that could be important in an investigation.
2. Attempt to locate a free tool that will allow for RAM analysis in a user-friendly fashion.