University of Pittsburgh
School of Information Sciences
IS 2621/TEL 2813 Security Management

(The first version of this lab was developed in Spring, 2014 by SFS student Sean Hutchinson)

Host Forensics Lab

Introduction:

In this lab, you will:

- Perform live acquisition of evidence from a victim computer
- Find digital artifacts through Memory analysis with Volatility
- Find digital artifacts through Windows Registry analysis with RegRipper
- Become familiar with some of the popular open source/free tools used in computer forensic investigations, such as AccessData's FTK Imager Lite and the Sysinternals suite
- Engage in standard processes of identifying, collecting, and analyzing relevant digital artifacts while keeping in mind the principles of sound forensic investigation, such as evidence preservation, documentation, evidence-based conclusions, etc.

Preliminary Questions:

1. State three general principles that a digital forensic investigator should follow. Hint: Look up ACPO and ADAM

2. Concerning the authentication and reliability of digital evidence, the American Law Reports describe six aspects of a "comprehensive foundation," which digital forensic investigators should demonstrate. List these.

3. Consider your answers to the previous questions:

a. Do you think it is important to test your forensics tools, especially those which are open source, for proper functioning prior to any kind of formal engagement with a client? Why?

b. State and explain three things you would do in order to maintain the integrity of an original piece of digital evidence. These may involve hardware, software, or methodology at any stage of your investigation, e.g. acquisition, analysis, etc. Although it does qualify, tool-testing should not be listed here!

What You Will Need:

→ 4GB or 8GB USB removable media, formatted for NTFS
→ Virtualization software such as VMware Player (highly suggested), VirtualBox, or Parallels
→ Download the following tools or get them from the GSA:
FTK Imager Lite 3.1.1 (http://www.accessdata.com/support/product-downloads)
Volatility 2.3.1 Standalone (https://code.google.com/p/volatility/downloads/list)
RegRipper v2.8 and Plugin Updates April 2013 (http://code.google.com/p/regripper/downloads/list)

→ Download the file from the following link (if you have problem, you can get it from the GSA:
The link: https://pitt.box.com/s/im38n45h2dg3aasdft7v2rwlkccasnni
Password: SecurityManagement_Spring2015

Setup:

1. Extract FTK Imager Lite to its own folder on your **USB drive**.

2. Extract everything from the RegRipper to its own folder, go into that folder and create a new folder named 'plugins'. Now, extract the April 2013 plugin updates to the plugins folder – make sure there is not another directory level when you extract these. It should just be rrv2.8\plugins\ a bunch of Perl scripts and a few other files.

3. Extract the SECMGMT-WIN7-HF virtual machine, import it into VMware Player, VirtualBox, Parallels, etc., start it (double check that it is host-only, no connection required) Password is: 3yeZp!easy
Note: The VM was created in VMware Player, but there are plenty of online documentations to assist you if you need to convert it. However, the VMware player is highly suggested in this lab since there are several compatible issues when you convert that VM disk. If you are stick to use VirtualBox or Parallels, You may contact the GSA to address these issues. Ideally, you will be able to 'Open an Existing Virtual Machine' and select the .vmx file.

Part 1: Live Acquisition of RAM

As you know, attackers often target servers or exploit employee workstations or end-user devices through phishing, and these systems and devices are on practically all the time, or at least most of the day. As such, acquiring images and/or potential evidence from a live machine is a likely situation in many digital forensic investigations. Moreover, when you encounter a victim system it may not be feasible to shut it down, or an intruder may still be poking around the system/network and you may not want to alert him or her to your presence. Besides, there is a plethora of potential evidence to be had from a victim system left running in its current state. The volatile contents of RAM should be captured as a snapshot of what is running on the system at that precise time, not to mention that it contains information that is not necessarily kept on the system's physical drive. Let's start here and capture the memory from the victim system.

There are many tools that can capture the memory from a live system, but we will be using FTK Imager Lite from AccessData. It is a GUI tool and compared to some other similarly purposed command-line tools, it leaves a larger footprint on the machine; however, all considering, its impact on the system is still rather minimal and it tends

to collect more reliable images. As an investigator, you will have to decide if it is more acceptable to lose the volatile information contained in RAM or, alternatively, to interact with the system and alter some information on the physical drive.

With the VM started and logged in...

1. Insert your USB stick. If necessary, connect it to the VM. Navigate to it in Windows Explorer and execute FTK Imager Lite. Click Yes in the User Account Control window.

2. Find the little memory stick icon in the FTK Imager Lite toolbar. If you hold the mouse over it, it should say 'Capture Memory'. Click it to view the Memory Capture window.

3. Select Browse and set your Destination Path to somewhere on your removable USB drive. Then, you may change the name of the captured memory file to whatever you like. Note, FTK Imager will create a raw memory capture – just be aware of different formats used in different tools. Leave the boxes unchecked and click the Capture Memory button. This will take a little while as it is capturing 1GB of memory.

4. When the capture has completed, the Status line should read 'Memory capture finished successfully'. Congratulations, you have captured the memory from a live system. Click Close, but keep FTK Imager Lite running. We will analyze the memory capture later.

Part 2: Live Acquisition of Windows Registry Hive Files

The Windows registry contains information on so many aspects of the operating system and applications that they are too numerous to go over individually, but needless to say, the registry provides a ton of evidential information. In a very basic overview, the registry is a repository for system and application settings and are stored in files called hives. These hive files are typically referenced by keys, which, in turn, contain values for various data types. If you've ever taken a peak at the registry, you have probably noticed the set of five root keys, namely the following: HKEY_CLASSES_ROOT; HKEY_CURRENT_USER; HKEY_CURRENT_CONFIG; HKEY_LOCAL_MACHINE; and HKEY_USERS. These last two are master keys, and the others are derived keys, meaning that they contain values derived from links to one or another master key.

Normally, an investigator would do a full disk image acquisition from a live system, but since we are limited on USB storage space and, for the purposes of this lab, we are only interested in certain sub-keys of the HKLM and HKU master keys, we will simply export the corresponding hive files of interest. The evidentiary value of these sub-keys will become more apparent when we run analysis on them. Let's get to it.

1. Using FTK Imager Lite again, we will locate and export some relevant registry hive files. First, click the Add Evidence Item icon in the FTK Imager Lite toolbar (should be the first icon), click Next with the Physical Drive option selected. The physical drive (the VM's virtual drive) should be selected already, just click Finish in order to mount it.

2. In the Evidence Tree pane, expand the physical drive, partition, and such, until you have expanded the 'root' directory.

3. Locate and export the following registry hive files. You simply right-click the file, select Export Files..., and choose a location on your USB removable media drive. You do not need any of the .LOG files.

Hive Key	Hive File
HKLM\SAM	Windows\System32\config\SAM
HKLM\SECURITY	Windows\System32\config\SECURITY
HKLM\SOFTWARE	Windows\System32\config\SOFTWARE

HKLM\SYSTEM	Windows\System32\config\SYSTEM
HKU\SID	Users\SECMGMT-WIN7\NTUSER.DAT
HKU\SID_Classes	Users\SECMGMT-WIN7\AppData\Local\Microsoft\Windows\UsrClass.dat

Good job! Now that we have a raw memory dump and some registry hive files, let's analyze them.

Part 3: Memory Analysis

For this part of the lab, you can disconnect your USB drive from the VM and make it accessible to your main operating system. We will be using Volatility Framework to analyze the raw memory dump that you saved to your USB drive. Volatility uses plugins, which makes it rather extensible, and the framework provides a powerful and dynamic memory analysis tool used by many investigators throughout the digital forensics community. I encourage you to explore this tool as much as possible. To start, you might check out the basic usage instructions at the following link: https://code.google.com/p/volatility/wiki/BasicUsage

1. Start a command shell and navigate to where your standalone Volatility executable is. Note, it may be helpful to have your memory image and the standalone Volatility exe in the same directory, whether on your USB drive or copied onto your main hard drive.

Note: It is helpful to rename your standalone Volatility exe to 'volatility'

2. Typically, you would run the 'imageinfo' plugin to determine the proper profile to use with any particular memory image you may be analyzing, but since that can take some time, the profile we need to use for the memory image here is: Win7SP0x86. In the shell, enter the following command:

volatility --profile=Win7SP0x86 -f <YOURMEMORYIMAGEFILE>.mem netscan > netscan.txt

where <YOURMEMORYIMAGEFILE> is the name you chose for your memory capture in part 1. Note, it is often helpful to send the output to a text file, especially if you are using a Windows command shell – it's just easier to read. This command, which uses the 'netscan' plugin, lists all network connections, protocols, IP addresses, ports, and associated processes.

Q1: Do you see any suspicious processes or open ports? List the owner, process ID, and local socket that you suspect may be malicious.

3. Enter the following commands:

volatility --profile=Win7SP0x86 -f <YOURMEMORYIMAGEFILE>.mem pslist > pslist.txt
volatility --profile=Win7SP0x86 -f <YOURMEMORYIMAGEFILE>.mem dlllist > dlllist.txt

Q2: What is the ID and name of the process that invoked your suspected process, aka the Parent process?

Q3: From where in the file system does your suspected process execute?

4. Use the **volatility -h** command to display a list of available plugins.

Q4: Using one of the plugins, list five privileges of your suspected process. Note, you can enter the -p <PID#> after the plugin name to get output related to only that process.

Part 4: Registry Analysis

For this part of the lab, we will use Harlan Carvey's RegRipper. Like most people, you probably do not know the registry through and through, or even hardly at all. This handy utility can save you a lot of time, frustration, and eye-strain, as it can be directed to extract specific information from the registry without actually going through a bunch of keys and values. In fact, RegRipper has no functionality built in to view the registry as you would with, say, AccessData's Registry Viewer. Like most extensible tools, RegRipper also makes use of plugins. You may select which plugins you want to run against a particular registry hive file by simply creating a text file and listing their names in it, then selecting your created file in the 'Profile' drop-down box in the RegRipper interface. RegRipper also comes with preloaded profiles, which may be used to analyze common keys in the registry, such as SYSTEM, SOFTWARE, SECURITY, NTUSER.DAT, etc. For the purposes of this lab, we will stick to these preloaded profiles since they are thorough and demonstrate the wealth of information that the registry contains. Let's get started.

1. Navigate to where you extracted RegRipper. Double-click rr.exe to start the application. At the bottom of the RegRipper window, it should read 'Profile List Populated.' If it does not, then you likely did not setup the plugins folder correctly – make sure that you have extracted the plugins, i.e. Perl files and a few text files, directly under the 'plugins' folder, which must be in the same directory as rr.exe, and try again.

2. Click the Browse button nearest the Hive File text box and go to where you saved the exported registry hive files from the VM (should be on your USB). Select the SOFTWARE hive file. Next, where it says 'Report File', select a location and file name for saving the output – you may want to name it after the hive file you are analyzing. Finally, under 'Profile', select the 'software' profile, and click Rip It.

3. Open and analyze the report file you just generated. You will notice that it contains quite a bit of information. Not all of it is relevant, but you are encouraged to browse through the information provided by each report you generate. It is often helpful to know the SID (Security Identifier) of users because many users' actions, settings, and activity are organized by their SID.

Q5: What is the SID of the currently logged on user? Hint: Hit Ctrl-F and search for 'profilelist'

4. In the same manner for the other registry hive files, create report files using each hive file's corresponding profile. Answer the following list of questions. Also, take notice of the ALERTs listed at the bottom of some of the report files and for each question, **be sure to write down any related timestamps.**

Q6: Of which groups is the currently logged on user a member?

Q7: What is the timezone bias? What standard time is this?

Q8: What is the victim system's computer name?

Q9: What is the system's network interface ID? IP address and subnet mask?

Q10: What is the status of the firewall?

Q11: List any suspicious typed URLs. Have there been any other connections made to this address? If so, say which plugin provided this information and list those connections?

Q12: List any other files that you think may be related to the suspected process you identified in part 3.

Q13: What kind of shell does the backdoor process provide?

Q14: Approximately when do you think the intrusion occurred?

Part 5: Memory Acquisition Tools

Imagine that at some point during your forensic analysis, you find that certain system DLLs have been compromised. Since many applications, including forensics tools, may also make use of system DLLs, it is quite possible that your acquisitions and/or analysis of evidence has been compromised or misled in some way. As a digital forensic investigator, you need to be able to identify which of your tools may have accessed these compromised DLLs and which of your tools have not. This is one reason why it is generally recommended that your live-analysis tools leave a minimal footprint.

As mentioned in Part 1, for a live acquisition of RAM on a victim or suspect system, the FTK Imager Lite tool from AccessData leaves a relatively large footprint on the system. Find another tool that can be used to acquire RAM from a live Windows environment, and show with screen shots that your selected tool has a smaller footprint on the system. Note, you do not need to do any memory capture for this exercise. Since FTK Imager Lite should have been running when you performed the memory capture in Part 1, you may want to use output from one or another Volatility plugins in order to gauge this tool's footprint. Perhaps, you may want to simply run FTK Imager and your selected tool and then issue the 'tasklist' command or use Sysinternals Process Explorer (Mentioned in Part 6). For a more complete understanding of how these tools impact the system, you should include process information such as amount of memory usage, number of threads, handles, and names and volume of accessed DLLs, and any other information that you deem pertinent.

Q15: Complete tasks in Part 5 and attach screen shots or insert them below.

Part 6: Registry Research

As you know, the registry is a big database that holds a lot of information about a given system and its applications. Depending on the specifics of an investigation, you may be interested in only certain information. Of course, the nature of digital forensic investigations requires that you conduct your analysis in a timely manner and as close as possible to the time of the reported intrusion. As such, you really do not have the time to search through the registry without knowing where you need to look, let alone the fact that enough information in the registry is not in any obvious location and may employ key names and values which are subtle at best. Whether you are testing your own tools, an attacker's tools, or a piece of malware, the point is that much time and work can be saved by doing a little research on how and where a system stores and uses relevant information. This will also improve your credibility in a court of law.

For this exercise, you will utilize Sysinternals Process Monitor (ProcMon.exe) in order to capture information written to and accessed from the registry. When ProcMon first launches, it automatically starts capturing information. You will want to stop the capture (Ctrl+E), clear the display (Ctrl+X), and use the toolbar to select only the 'Show Registry Activity'. Consider actions likely to take place during or after an intrusion – WWOD (What Would Oscar Do), and perform two different actions/changes to some setting on the system. You will notice in the display that you get flooded with registry activity if you do not set any filter; thus, for each action you take, consider what might be a likely keyword pertaining to your action or setting change, and proceed to add a filter (Ctrl+L). In the Process Monitor Filter window, under where it says “display entries matching these conditions”, choose “Detail” from the first drop-down box, and choose “contains” from the second drop-down box, then enter your keyword. Leave all other settings as they are and press the Add button. Click Apply, then click OK. Start the capture (Ctrl+E) again and proceed to perform an action/change a setting. If you do not see anything in the display, try different

keywords for the filter, and **remember to remove the old keyword filter.**

Q16: For each action/setting change, briefly describe what you did and provide a screen shot of the corresponding registry changes. Attach screen shots or insert them below.

Bonus: Alternate Data Streams

See if you can find the alternate data stream (ADS). To earn bonus points, you must obtain the name in the ADS. There are a few ways that you can locate it. Regarding the tools used in this lab, consider the possible plugins that might help you find the ADS, e.g. like how it was created. Also, you may want to check out the tools available in the Sysinternals Suite, which can be found at: http://technet.microsoft.com/en-us/sysinternals/bb842062.aspx Remember, you must first locate the file containing the ADS, then access the ADS, then record the name within it.

Q17: What is the name?

For a good analysis using Volatility, check out "Solving the GrrCon Network Forensics Challenge with Volatility" at http://volatility-labs.blogspot.com/2012/10/solving-grrcon-network-forensics.html

Submission

Submit the report to the GSA by email (lej17@pitt.edu) or stop by his office at Room 410 to hand in it to him.

*This lab was inspired by case studies and exercises from the following sources:

[1] Steven Anson et al. *Mastering Windows Network Forensics and Investigation*. 2nd Ed. Sybex, 2012.