CS4677 Computer Forensics: Lab 6 – Introduction to malware analysis

The purpose of this lab is to introduce you to basic malware analysis techniques. The two predominant methodologies for conducting malware analysis are static analysis and dynamic analysis. In static analysis, tools are used to dissect and analyze a malicious article without ever running it. Static analysis can offer a comprehensive view into the behavior of the malicious article, but is often much slower than dynamic analysis, and can be extremely difficult, especially if the functionality of the malware has been deliberately obscured.

Dynamic analysis involves executing a malicious article within a fully instrumented environment in order to observe the behavior of the malicious item. Dynamic analysis can yield very fast results, but often fails to reveal the full behavior of the item under analysis. To mitigate the threat posed by live malware, dynamic analysis is often conducted in a "sandbox" environment. A sandbox is a system/network that allows for running of malicious software while minimizing the chance that the malicious software can infect hosts outside the sandbox or propagate off of the sandbox network. A well configured sandbox provides enough services to convince the malware that it is running on a host in the real world, is generally well instrumented in order to capture the observable behavior of the malware, and is often rapidly reconfigurable (via VMware snapshots for example) in order to restore the sandbox to a pristine state.

I. Configuring your sandbox

This lab focuses primarily on dynamic analysis. You will make use of one or more of your virtual machines to observe the behavior of a piece of malware. You will begin by configuring your virtual machine(s) with the tools necessary to observe the behavior of the malware—some of these tools are new, but you are already familiar with many of them from other labs. Once your virtual machines are properly configured, you will take a snapshot of them before introducing any malware.

1. Start both your Windows XP and your Fedora virtual machines. They should both be using bridged networking.

2. On your Windows VM you should have desktop icons for regshot, Immunity Debugger, and IDA Pro Free. On the C: partition you should also find a folder named
   `C:\Program Files\SysInternals`
   that contains all of the SysInternals tools. If you do not have one or more of these tools, then use a browser to download and install them from
   http://192.168.100.6/cs4677/software

3. Download the following file to your Windows desktop. **THIS IS REAL WORLD MALWARE DO NOT RUN IT YET**.
   http://192.168.100.6/cs4677/lab6/malware.exe.bin

4. On your Fedora VM, create a lab6 directory and download the following file into it
   a. http://192.168.100.6/cs4677/lab6/malware.exe.bin

5. Reconfigure both virtual machines for host only networking, and acquire new IP addresses for each virtual machine.
   a. Windows:
      i. `ipconfig /release`
      ii. `ipconfig /renew`
   b. Linux
      i. `pkill dhclient`
      ii. `dhclient eth1`
   c. Make note of your Linux IP address

6. Once you have installed all of the required software (or verified that it's installed already) take a snapshot of each virtual machine using the VM/Snapshot/Take snapshot menu option. If you ever wish to reset the state of the lab, you can revert to your saved snapshot using VM/Snapshot/Revert to snapshot.

7. In your Windows VM, modify the network settings (Start/Settings/Network connections) for the Local Area Connection by right clicking and selecting Properties.
    a. Select TCP/IP Properties
    b. Click "Use the following DNS server addresses and enter the IP of your Linux VM.
    c. Click Advanced, then under Default gateways click Add and enter the IP address of your Linux VM.
    d. The purpose of these changes is to direct all network traffic from the host that will be infected to a host that you control

II. A quick look at the malware

Before executing a piece of malware, you should perform at least some cursory static analysis of the malware.

1. In your Fedora VM run the strings and file utilities against malware.exe.bin. You may wish to read the man page for strings to see if there are any options that may be of interest.
2. Refer to Part II of the questions section, below.

III. Preparing to run the malware

One of the most important aspects of dynamic malware analysis is change detection. Before infecting a host with a piece of malicious software, you should consider all of the changes that the software might make to your system and be prepared to record those changes. You should also be prepared to record events that are observable but perhaps leave no permanent artifact. Simple changes that you might wish to observe include changes to the windows registry, changes to the file system, creation of new processes, and generation of any network traffic. In this portion of the lab you will instrument the Windows VM so that you can observe the behavior of the malware when you finally launch it.

1. The Windows registry is constantly being accessed by both the operating system and running programs. The best way to observe changes to the registry is to take snapshots of the registry before infection and after infection. When doing this you want to make sure that you minimize the changes taking place to the registry so that you can associate the changes with the malware rather than other running software. Several of the tools that you will use in the lab create registry entries the first time they are executed. Accordingly, these tools should be run once before you ever take any registry snapshots.
    a. In your SysInternals folder, launch then close procexp.exe and procmon.exe
    b. Launch and then close IDA Pro
    c. Launch and then close Immunity Debugger

2. Now take a snapshot of the Windows registry using regshot
    a. Launch regshot
    b. Choose the HTML document format
    c. Change the Output path to your Desktop.
    d. Choose 1st shot, then shot (no need to save, but DON'T CLOSE THE PROGRAM YET)
    e. You have captured the state of the registry prior to infection.
    f. Leave regshot open in the background somewhere for now.

3. Launch procexp.exe to see a list of processes running on the Windows system

4. Launch procmon.exe to see a list of events taking place on the system. Procmon will generate a lot of noise so we will build a filter to focus on just the events generated by our malware
    a. Open the filter dialog(Ctrl-L) if it doesn't come up automatically.
    b. Use the editing tools at the top of the dialog to create a filter that reads
    `Process Name is malware.exe then Include`
    c. Click the Add button, then click OK.

5. On your Windows host (or Fedora host) launch wireshark and begin a packet capture on the host only interface (the one with the IP on the host only network). Choosing the wrong interface will result in no

packets captured. Depending on your configuration, you may be prompted to enter a password to enable the VM to capture packets on this interface.

IV. Infect the Windows VM

With your instrumentation in place, it is time to run the malware. You might want to take a second snapshot here (but don't overwrite the first one). You may also want to read through the questions section before proceeding.

1. In the Windows VM rename malware.exe.bin to malware.exe
2. Double click on malware.exe, make note of anything unusual that you see.
3. After a few moments, terminate the event capture in procmon (Ctrl-E)
4. Stop your wireshark capture and save the packets as lab6_initial.pcap. You will turn this file in.**
5. Use procmon's File/Save menu to save the events to a CSV file to turn in with your lab. You will turn this file in.**
6. Using regshot, take a second registry snapshot (Click 2nd Shot)
7. Once the shot is complete, click Compare. A browser will open showing the difference between the two snapshots. Use the browser file menu to save the page. You will turn this html file in.**
8. Scroll through the process list in Process Explorer and read through the output of procmon and regshot. Determine which process was likely created by the malware.
9. In Process Explorer, right click on the malicious process and select Create dump/Create mini dump and save the dump file into C:\cygwin\home\MyName
10. Open a cygwin terminal and run strings on the dump file you just created, saving the output of strings to a file named dumped_strings.txt. You will turn this file in.**
11. Use Process Explorer to kill the malicious process.
12. Reboot the Windows VM. Is the malicious process running when you login again? If so kill it.
13. Refer to Part IV of the questions section.

V. Network behavioral analysis

Most malware will attempt to communicate with the outside world, and thus generate network traffic. Depending on the capabilities of the malware and the configuration of your sandbox environment, you may observe very little of the malware's network behavior. This portion of the lab is somewhat free-form because I want you to consider how you can coax the malware into doing more by modifying the configuration of your sandbox.

1. Review the packets that you captured using wireshark.
2. Read through the questions in Part V of the questions section.
3. With the questions in mind, develop a strategy to modify your sandbox environment so that the malware will make some progress.

4. Restart wireshark, then relaunch the malicious process.

5. If the malware seems to be stuck, stop your wireshark capture and save the packets in a serialized file (lab6-1.pcap, lab6-2.pcap, etc) You will turn these in.**

6. Kill the malware process and return to step 1 in this section. Repeat this process until you feel that there is nothing else you can do to get the malware to make progress based on the behavior you have observed.

VI. Cleanup

When you feel that you can make no more progress with the malware, use the Revert to snapshot feature on the Windows virtual machine to return the VM to its the pre-infected state. You may not wish to do this until you have answered all of the questions in the lab. If you have prematurely reverted, then you will need to restart the lab from step I.7 above.

Questions:

Part II

1. (2 pts) What does the output of file tell us about the malware?
2. (3 pts) What if anything does the output of strings suggest about the malware?

Part IV

1. (1 pt) Following the initial infection, was malware.exe running when you looked at your Process Explorer output?
2. (2 pts) Which process was created by the malware?
3. (10 pts) What files were created by the malware?
4. (10 pts) What registry change(s) were most likely made by the malware (you should correlate what you see in the regshot output with what you see in the procmon output).
5. (10 pts) What was the malware attempting to achieve with the change(s)?
6. (5 pts) What does the content of the strings dump suggest about the possible behavior of the malware?
7. (2 pts) Following the reboot, was the malicious process running? How did this happen?

Part V

1. (10 pts) Explain the purpose of each packet you see in the packet capture beginning with the first packet and ending when you start to see a pattern of repetition
2. (5 pts) Why does the pattern begin to repeat?
3. (10 pts) Based on the packet capture how might you modify your sandbox environment to get the malware to do something more?

General

(10 pts) Based on the information gathered in the lab, what steps are required to clean this malware from an infected system? Verify that your suggested procedure are correct by applying them, rebooting the VM and verifying that the malicious process is not running.

Deliverables:

1. (5 pts) The saved procmon event file.
2. (5 pts) The save regshot before and after comparison file.
3. (5 pts) dumped_strings.txt
4. (5 pts) All pcap files generated during the lab.

Post your files and answers to the questions to Sakai NLT than 2355 on the due date.