**CMSC 426 Computer Security Lab 4**

Name: Gokul Natarajan

Assigned: 5/2/2022

Due: 5/16/2022 at 11:59pm

Total points: 100

**Virtual Machine Setup**

To set up the lab, you will need to import the lab VM into VirtualBox. You can download the VM from the following link:

* <https://drive.google.com/file/d/1DSd0Sh_9QvmZSA0WNOLySFEYfA0aBxs3/view?usp=sharing>

Import the VM into VirtualBox by using the menu option File -> Import Appliance and selecting the downloaded Lab4\_VM.ova file. You will need **28 GB of available disk space** after downloading the .ova file, and before importing it! Once you have imported the VM, you can delete the .ova file to save space.

Start the VM and log in. The password to the VM is infected. By default, the VM’s network settings are configured to be in NAT mode, which allows it to connect to the internet.

**Part 1: Basic Static Analysis (25 pts)**

In this lab you will analyze CMSC426Lab4.exe, a malware sample which has been defanged so that it cannot communicate with its original command and control (C&C) infrastructure. It is located at C:\users\student\CMSC426Lab4.exe on the VM. **You should treat this file as if it were live malware**. Make sure to follow all instructions carefully.

**UNDER NO CIRCUMSTANCES should you run the malware until instructed to do so.** You will later be given directions to configure the network settings of the VM so that it is isolated on your VM and cannot reach your host machine or the internet. In this section of the lab you will be given directions to analyze the malware statically – you will obtain all answers for Part 1 **without running the file.** You may use any of the static analysis tools provided on this VM or any other tools discussed in class (e.g. VirusTotal) to answer the following questions about CMSC426Lab4.exe:

1. What is the MD5 hash of CMSC426Lab4.exe? (3 pts)

73d88fe552e212ab57a008c42591f83e

1. In a few sentences, describe why file hashes (MD5, SHA-1, SHA-256, etc.) are useful to malware analysts. (6 pts)

Hashes are used to keep track of a specific malware sample. Threat reports can be shared.

1. Imagine an antivirus product that has a list of file hashes corresponding to every single malicious file that has been previously seen. The antivirus uses this list of file hashes to identify malware on the filesystem of the computer it is installed on. Describe some weaknesses of this approach. Make sure to give an example of how a malware author could ensure that their malware goes undetected. (8 pts)

The file hashes will change even if one bit is changed in the file, so a malware author could just change some parts of the file, while keeping the malware functional and undetected.

1. How many DLL files does the malware import functions from? (4 pts)

3

Text

Description automatically generated with medium confidence

1. What is the compilation timestamp of this malware sample? (4 pts)

08/22/2011

Graphical user interface, text, application

Description automatically generated

**Part 2: Writing a YARA Rule (40 pts)**

In Part 2 of the lab, you will write a YARA rule to detect CMSC426Lab4.exe and another malicious file that is very similar to it (which has been hidden somewhere on the VM). During this part of the lab, **do not run the malware sample**. You will be able to do the entire part using static analysis tools such as the strings command-line utility and a disassembler such as Ghidra or Ida Pro.

YARA is a pattern-matching tool that is widely used by malware analysts for writing malware signatures. It can detect files based on strings, bytes sequences, file metadata, and other patterns.

Documentation for YARA is linked below:

* <https://yara.readthedocs.io/en/stable/>
* <https://yara.readthedocs.io/en/stable/wusCyararitingrules.html>

To scan a file with a YARA rule, open a command prompt on your VM and run yara64.exe. Here are a few example commands.

To scan the file fileToScan.exe with a YARA rule defined in myRule.yar, run:

* yara64.exe myRule.yar fileToScan.exe

To scan all the files in a folder named folderToScan/ instead of a single file, run:

* yara64.exe -r myRule.yar folderToScan/

In this lab, you will write a YARA rule for CMSC426Lab4.exe containing three strings and a byte sequence. First, use the strings command-line utility to identify strings matching the following descriptions.

1. Provide the string CMSC426Lab4.exe that is a URL: (3 pts)

https://www.thisisthemalwareswebsite.com./start.html

1. Provide the string CMSC426Lab4.exe that is a User Agent: (3 pts)

ThisIsTheEvilUserAgent

1. Provide the string CMSC426Lab4.exe that is a file path: (3 pts)

C:\badfile.exe

Next, you will identify a unique byte sequence in CMSC426Lab4.exe. Often, this can be done by selecting bytes from a unique function in the file (such as a custom encryption routine). In this case, you will select a sequence of bytes from one of the functions that handles much of the malware sample’s networking capabilities. Instructions for this part of the lab will be provided for Ghidra, but you are welcome to use Ida Pro instead if you wish.

Run Ghidra and select File -> New Project. Create a non-shared project with a name of your choice. Next, select File -> Import File. Navigate to CMSC426Lab4.exe, select it, and hit OK.

Double click CMSC426Lab4.exe under your project to open it. Ghidra will begin analyzing it; you may choose all the default analysis options when prompted.

1. At what address is the Windows API function InternetOpenA called inside of CMSC426Lab4.exe? Ghidra provides this address as a hexadecimal number – please leave your answer in this format. (4 pts)

00407698

Switch to the listing view in Ghidra, showing the call to InternetOpenA. Find the bytes representing the machine code for the five assembly instructions starting with CALL InternetOpenA. This should be approximately two dozen bytes.

1. What are the bytes corresponding to these five instructions? Leave them in hexadecimal, separated by spaces (6 pts).

49 6e 74 65 72 6e 65 74 4f 70 65 6e 41 00

Write a YARA rule that matches files which contain the three strings from questions 1-3 and the byte sequence from question 5. Remember that in YARA syntax, any strings containing backslashes must be escaped by adding an additional backslash! (eg. \\ instead of just \)

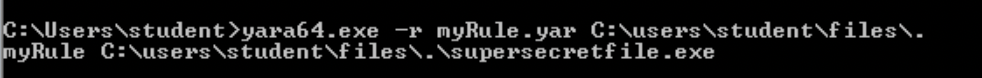
1. Provide a screenshot of your YARA rule below (10 pts)

Text

Description automatically generated

Scan the folder C:\users\student\Files\ using your YARA rule. One of the files in this folder is very similar to CMSC426Lab4.exe and it should be detected by your rule.

7. Provide a screenshot of your rule detecting the similar file in C:\users\student\Files\. It should not detect any other files. (8 pts)



8. What is the MD5 hash of the similar file that your rule detected in System32? (3 pts)

cf1ce84cafdf6465fc9ac0548716119e

**Setup for Dynamic Analysis**

Once you have completed CMparts 1 and 2, shut down the Lab4 VM. You will need to configure its networking settings in VirtualBox to prepare for running the malware safely. Go to the network settings for your VM (Settings -> Network). Make sure that Adapter 1 is enabled, that it is attached to Host-only Adapter, and that its name matches the name of the host network interface that you used in the previous labs (e.g. VirtualBox Host-Only Ethernet Adapter)

Graphical user interface, text, application

Description automatically generated

Once you are finished configuring the network settings, boot the Lab4 VM.

On your Lab4 VM, go to “Control Panel” -> “Network and Internet” -> “Network and Sharing Center” -> “Local Area Connection”. Select “Properties” -> “Internet Protocol Version 4 (TCP/IPv4)” -> “Properties”. Set your VM’s IP address and DNS server to the following values:

Graphical user interface, application

Description automatically generated

Once you have configured your VM’s network settings, **take a snapshot of your VM in VirtualBox**. You need to do this to ensure that you can restore your VM to a state prior to when it was infected by the malware. This step is **NOT OPTIONAL**.

**Part 3: Writing a Snort Rule (35 pts)**

In Part 3 of the lab you will write a Snort rule to detect the network traffic of CMSC426Lab4.exe. Snort is a network-based intrusion detection system.

First, you will use FakeNet-NG and Wireshark to capture the malware’s network traffic. FakeNet-NG is a tool that intercepts and redirects network traffic. This will prevent the malware from interacting with your host machine, and it will trick the malware into believing that it is connected to the internet. Since many types of malware expect to be connected to the internet in order to receive instructions from a C&C server, this tool allows analysts to observe further behavior from the malware than if the VM were fully disconnected from the network.

First, ensure that you have **taken a snapshot of your VM** with the network settings configured as specified by the earlier directions. You must take a snapshot prior to running FakeNet-NG. It is not easy to restore your VM’s prior network settings after you have run it. When you are ready, run FakeNet-NG (and allow it through the firewall, if prompted). To test that it is working correctly, ping google.com from the command line. It should respond with the IP address 192.168.56.10, showing that FakeNet-NG is intercepting and redirecting the ICMP traffic.

Text

Description automatically generated

Next, run WireShark and begin capturing network traffic on the Npcap Loopback Adpter. Leave FakeNet-NG and WireShark running so that you can capture the malware’s network traffic.

With FakeNet-NG and WireShark running, open an **Administrator** command prompt and run cd C:\Users\student\ to change to the directory in which CMSC426Lab4.exe is located. Then run CMSC426Lab4.exe in the Administrator command prompt.

The malware should repeatedly produce the same network traffic at short intervals. In Wireshark, identify traffic sent by CMSC426Lab4.exe. Right click one of these packets and select Follow TCP Stream (Note: It is not a DNS request/response).

1. Provide a screenshot showing the output of the Follow TCP Stream in WireShark. The User-Agent HTTP header should show the user agent string that you found in Part, and the Host HTTP header in the screenshot should be based on the URL you found in Part 2. (10 pts)  
  
Graphical user interface, text, application, email

Description automatically generated

Once you have found the malware’s network traffic in WireShark, go to your VM’s network settings and set the network adapter to “Not Attached”. This will disconnect your VM from the network. Then, you may close Wireshark and FakeNet-NG.

FakeNet-NG saves .pcap files containing the redirected network traffic to C:\Users\student\Desktop\fakenet\_logs\. You will use Snort to detect network traffic in this .pcap that was produced by the malware sample.

This is the best resource for writing Snort rules: (you should be able to write the rule for question 2 using just this video, except for the part that uses the flow keyword)

* <https://www.youtube.com/watch?v=8T8XVoNqMbc>

If you want some more information about Snort, these are also helpful

* <https://www.rapid7.com/blog/post/2016/12/09/understanding-and-configuring-snort-rules/>
* <https://coralogix.com/blog/writing-effective-snort-rules-for-the-sta/>
* <https://resources.infosecinstitute.com/topic/snort-rules-workshop-part-one/>

Note that Snort 2.9.14.1 is installed on the Lab4 VM, so some syntax for Snort 3 may not work.

Create a new file named local.rules inside the folder C:\Snort\rules\. You should write a snort rule for detecting the malware’s network traffic inside of this file. Follow these instructions when writing your snort rule:

* Your rule should match HTTP traffic containing the user agent string from Part 2
  + You do not need to match the entire user agent string
  + Just up to (not including) the first semicolon
* Your rule should use the flow keyword to match only outbound traffic
* Choose a unique Snort ID (sid) for your rule that is greater than 1000000
* Your rule should also contain a message (msg) about the detected network traffic.
* For readability, please make a multi-line Snort rule, with each line ending with a \ (see the Rapid7 resource linked above for more details)

Before running your rule, create an empty folder named snort\_dynamicrules inside of C:\Snort\lib\. The Snort config file requires this folder, but I forgot to make it before releasing the VM.

To run your Snort rule, use the following command in the command prompt:

* C:\Snort\bin\snort.exe -c C:\Snort\etc\snort.conf -r [path to .pcap file] -l C:\Snort\log

Alerts from your rule will be written to C:\Snort\log\alert.ids

2. Provide a screenshot of your Snort rule below (15 pts)

Text

Description automatically generated

3. Provide a screenshot of your alert.ids file, showing that it has matched traffic originating from the malware sample (10 pts)  
  
Text, letter

Description automatically generated