Lab1 - Basic Static Analysis

Issued: June 3rd, 2021
Due: June 10th - 11:59PM

Name: <Kamalesh Ram Chandran Govindaraj>
<kgovinda >

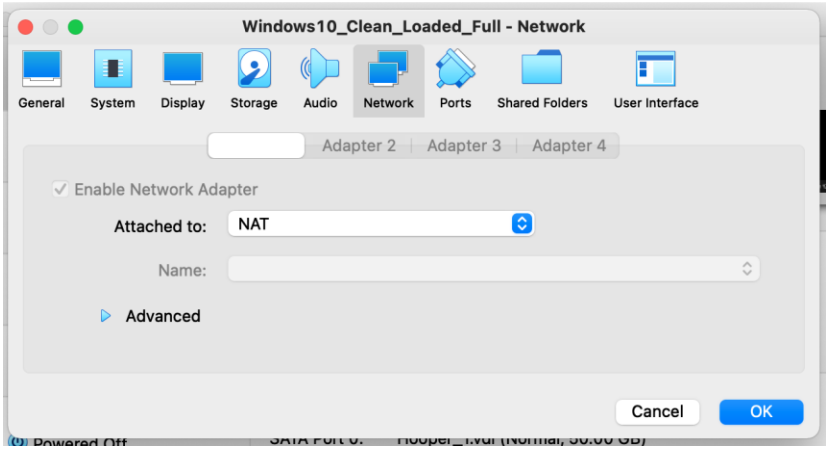
Step 1: Setup VM Infrastructure

Goal - The goal of this activity is for you to understand how to configure a virtual machine (VM) for malware analysis. The critical lesson you will learn is that *attention to detail matters*. Follow these steps carefully and you will be fine! In Step 1, you will go through a basic configuration to setup a Flare VM on . You will not be graded on this portion of the lab. Concepts you will learn in this activity:

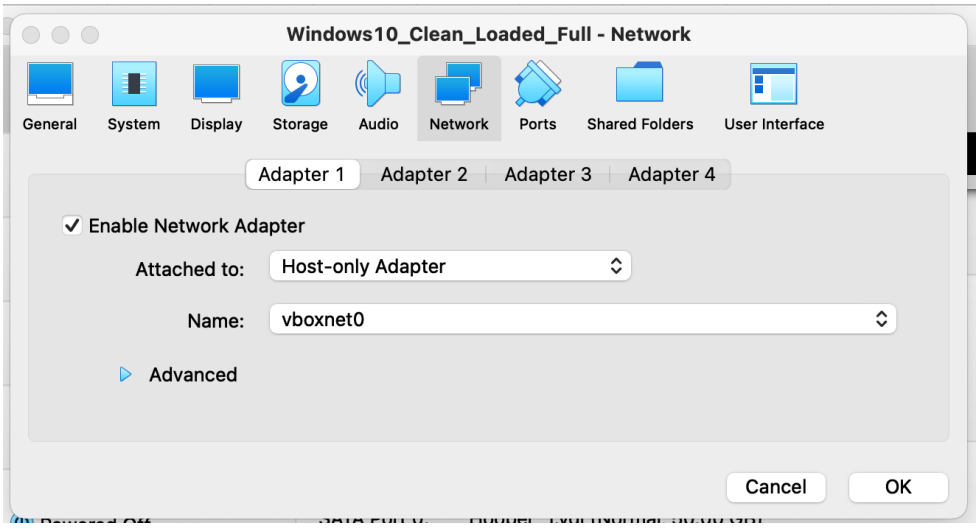
- How to use VirtualBox
- How to configure a VM for malware analysis
- How to enable host-only on VMs

Steps:

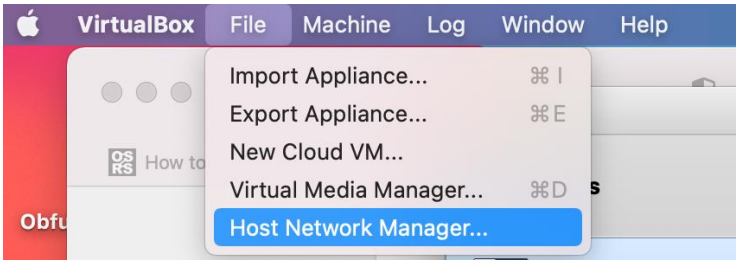
1. Install the latest version of VirtualBox where you plan to run the analysis VM. If you are not familiar with VirtualBox or VMs - find a tutorial on-line and give it a read. It's not a terribly hard technology to understand and use.
2. I have made a location on Google Drive with a Windows 10 VM for VirtualBox that you may download for this exercise. You will find it here: https://drive.google.com/file/d/16mvO7wKO1d14nEXP8Y3A0WoFTxblt7SN/view?usp=sharing
3. Download this VM. It is:
    a. Loaded with Windows10 Pro
    b. Configured so that Windows Defender is turned off
    c. Has a UN/PWD of `cse410/cse410`
    d. Any security questions you come across the answer is always: `cse410`
4. Import the VM into VirtualBox.
5. Right click on the imported VM, select Settings, and then select Network. Make sure that NAT is selected. This will provide you access to the Internet which you will need to configure the VM. We will set this VM to be offline once we are done configuring. To verify the NAT setting you are looking for this screen.



6. Next you will install the Flare VM. I described what this VM is in class. Follow the instructions on this page precisely.
   https://github.com/fireeye/flare-vm
   Note, the VM I provided you has a 50GB HD. The installation notes say you should have 60GB but 50GB worked for me. Let me know otherwise!
7. It will take a while to do the Flare VM install - almost an hour. Welcome to malware analysis. Be patient, and let it finish. To see their pretty desktop, restart the VM. It will have a black theme.
8. This VM is the idealized one upon which to perform Windows malware analysis. However, you need to make one more adjustment. Shutdown the VM. Right click on it and access Settings and then Network. This time, however, set the "Attached to: " value to "Host-only Adapter"



You must remember to do this. It will prohibit access to the Internet for this VM which is what you want once malware is on the VM. Note, you need a network adapter - which in this case is vboxnet0. VirtualBox may not have that at outset. It is trivial to create. You may create one from the **Host Network Manager** in VirtualBox - shown in the figure below.



9. Double check - make sure the VM is set to **Host-only Adapter** and move onto to the lab itself.
10. Last step. And this is **critically important**. You now have a **clean**, Windows 10 Flare VM. Export a version of this as an OVA. You will need this for at least four of the remaining five labs. Each lab you will import this OVA file to start fresh. Make certain you remember where you exported it to!
11. Really make sure you did step 10!

Step 2: Lab Exercise

Exercise 1: Tuning a Windows VM

What You Need: The Internet and the ability to search Github from your laptop - not the VM.
Problem: You are about to run the Windows VM. Windows likes to do lots of annoying things by default. Like run WindowsDefender or run the Firewall or run telemetry services that send information back to Microsoft about your machine's performance. I turned off Windows Defender using a combination of information from the following two sites:

Turn off Windows Denfender https://www.windowscentral.com/how-permanently-disable-windows-defender-antivirus-windows-10#disable_defender_securitycenter
Turn off Tamper Protection
https://www.thewindowsclub.com/how-to-enable-tamper-protection-in-windows-10

You will need to do the following. Search on github (https://github.com) and find PowerShell scripts or Windows batch files that turn off these services. Specifically, fill in the table below with at **least one** URL for a gitbhub site with such scripts.

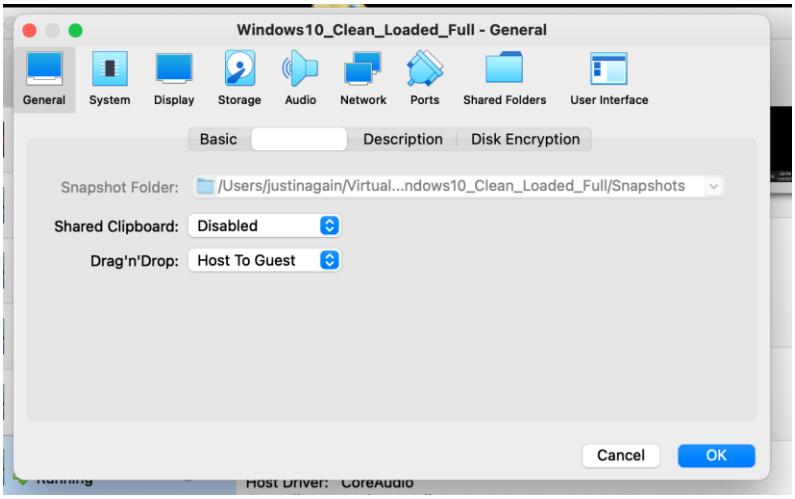
Score	**/ 2 pts**
Service	URL where script was found
Defender	Debloat-Windows-10/disable-windows-defender.ps1 at master · W4RH4WK/Debloat-Windows-10 · GitHub
Telemetry	Win10-Initial-Setup-Script/Win10.psm1 at master · Disassembler0/Win10-Initial-Setup-Script · GitHub

One note about the github site you find - and partly the reason I have you do this exercise. Whatever it is, these types of scripts serve a dual purpose. In one respect - they help us with our lab or may legitimately help a company turn off Windows Defender if it has purchased another anti-virus software like McAfee. However, if a machine is compromised, these are the types of payloads sent by a command and control (C2) server to its clients in order to make machines less secure and more vulnerable.

Exercise 2: Packing

What You Need:

- The Flare VM you configured.
- Make certain the Flare VM is in **Host-only Adapter** mode so there is no Internet connection.
- Make certain the Flare VM is setup to allow copying from host to client. Here, right click on the VM and go to **Setting > General > Advanced** and enable **Host to Guest** Drag n' Drop. Like this:



- You will need to download the malware samples from this location, a file named secure.zip: https://drive.google.com/file/d/12NWNhkacfdxsb0059pVlouSxpidmkRAnYwb/view?usp=sharing
- Once downloaded to your machine, drag and drop the encrypted zip file onto the Flare VM desktop.
- Be sure to then:
    - Delete the zip file from your machine once copied. Empty your recycle bin!
    - Shutdown the Flare VM
    - Turn off **Host to Guest** Drag n' Drop
    - Restart the Flare VM
    - Again - we need to be careful!
- You will need the folder secure.zip which is on the desktop. This holds the malware. Unzip this folder. The password is: Password123!

Problem: The folder with malware samples is packed by different packers. You will need to use a tool on the FlareVM to examine each of the files listed in the table below and fill in the content. If you click the Windows icon and search on the name - you will find that it is installed - we discussed it in class. In the second column I am looking for the name of the packer. In the third column I want the entropy of the CODE or .text section. Hint, you can use a resource described in Lecture to find the entropy of the CODE or .text section. In the last column, decide whether the CODE or .text section is simply compressed or compressed and encrypted.

Answer:

Score	**/ 4 pts**		
File	Packer	Entropy of CODE or .text Section	Compressed or Compressed and Encrypted
cd73d1cadd1a57f85ecb3ad41ae346e042d4b919dd8745cacc6f2cc79636b71d	tElock v0.98	7.992	Compressed/Encrypted
c14b5a2cf442a3750cec9e5b0166ac264a737afe56c6221279c1285edd156d21	[PyInstaller v.3.6	6.666	Compressed

The c14b5a2cf442a3750cec9e5b0166ac264a737afe56c6221279c1285edd156d21 packer is interesting. In the context of Windows malware, search the Internet and explain what the packer does and what type of malware it is (like Virus, Trojan, Worm, etc.)

Answer:

Score	**/ 2 pts**
PyInstaller bundles a Python application and all its dependencies into a single package. The user can run the packaged app without installing a Python interpreter or any modules. If a infected code or module is packed, when the original program executes it will remain hidden like a trojan horse.	

Exercise 3: Manual Examination of Malware with PEStudio

What You Need:

- Same as for Exercise 2 except now you will need PEStudio which is also installed on the machine (as part of Flare VM).

In class we learned about PEStudio which allows you to do a deep dive on files that conform to the Microsoft PE standard. The tool also blends other tools we learned about, such as

VirusTotal, to enable you to access whether a given file is malware. This problem will have you focus on the Import section of a file from the secure zip file. Open PEStudio and load the file:

8fcf6f749ef0697bc195e551ec7781ce37c67c2e70644291a8cd17b8aff853dd

Note, it takes a while for PEStudio to examine the file. Be patient.

Answer the following set of questions:

1. What are the libraries and which one is blacklisted:

Scoe	**/ 2 pts**
Libraries	Blacklisted (Y/N)
Wsock32.dll	Y
Kernel32.dll	N
User32.dll	N
Advapi32.dll	N
Oleaut32.dll	N

2. The blacklisted DLL imports a host of methods. Based on what is being imported - what do you think this malware is doing? That is, when it runs what type of process might it create. Note - we won't know for sure until later lectures when we decompile the code!

Score	**/ 2 pts**
The DLL imports methods such as send, select, socket, recv, listen, inet_addr and inet_ntoa. These imports are used in network for creating and sending message between client and server. If this dll is sending and receiving communication from a remote server then it can be used to prompt a remote shell or send payload to infect the system through a back door.	

Once you are done with this question you may turn off the FlareVM.

Exercise 4: Identifying Trends Across a Large Sample

What You Need:

- You will need a zip file with malware JSON summaries located on Google Drive. I shared it with folks, you will find it here:

https://drive.google.com/file/d/1_0WbbNzmqIh2w6kawjBdexQCVaZZX2Yy/view?usp=sharing

- There are 2,295 JSON reports on WindowsEXE malware samples provided by VirusTotal. You should use python3 to examine these files.

Problem: This problem will take you more time. So far, we have manually looked at individual malware samples and understood characteristics within them. However, you need to consider large sets of data. This set of JSON reports is one such example. The reports list out, among a lot of other information, the scanners used by VirusTotal and whether they detected the file described by the report as malware or did not. Here is a snippet below of the JSON where the scans section begins - detected:true means it flagged it as malware detected:false means it did not:


```
▼ scans:
  ▼ Bkav:
      detected:                    true
      version:                     "1.3.0.9899"
      result:                      "HW32.Packed."
      update:                      "20200506"
  ▼ DrWeb:
      detected:                    true
      version:                     "7.0.46.3050"
      result:                      "Win32.VirLock.10"
      update:                      "20200506"
  ▼ MicroWorld-eScan:
      detected:                    true
      version:                     "14.0.409.0"
      result:                      "Win32.Virlock.Gen.1"
      update:                      "20200506"
  ▼ FireEye:
      detected:                    true
      version:                     "32.31.0.0"
      result:                      "Generic.mg.151dceb6a965f670"
      update:                      "20200316"
```


What we are interested in is the best and worst malware detection tools. So, you should:

1. Create a python program that iterates over all the JSONs in the directory.
2. Count the number of times a scan tool identifies a report as malware across the entire set of reports. So, you should have output like

Bkav 2286
MicroWorld-eScan 2295
...

3. Number two seems hard. Here is a hint. Each time you process a report, place the mention of a DLL in a List. Then when you are done processing all the fies, use a Counter to get the count of objects in the list. Google will have examples. There are dozens of other ways to do this.
4. Then sort the values.

In the table below list the **complete** set of malware detections by company, ordered from largest to smallest. I should see the entire list in the answer section - one company per row.

Score	**/ 4 pts**
('APEX', 2285), ('Microsoft', 2272), ('FireEye', 2271), ('Endgame', 2270), ('Invincea', 2269), ('Fortinet', 2263), ('GData', 2260), ('Rising', 2260), ('MicroWorld-eScan', 2255), ('ESET-NOD32', 2255), ('MAX', 2254), ('BitDefender', 2253), ('Ad-Aware', 2253), ('McAfee', 2251), ('SentinelOne', 2248), ('Cybereason', 2248), ('CrowdStrike', 2247), ('Antiy-AVL', 2247), ('K7GW', 2245), ('K7AntiVirus', 2244), ('Emsisoft', 2240), ('ZoneAlarm', 2240), ('Kaspersky', 2237), ('AhnLab-V3', 2237), ('Acronis', 2237), ('Sophos', 2233), ('AVG', 2233), ('Cyren', 2232), ('Sangfor', 2224), ('Avira', 2220), ('DrWeb', 2219), ('VBA32', 2219), ('Arcabit', 2201), ('Qihoo-360', 2187), ('BitDefenderTheta', 2186), ('VIPRE', 2183), ('Avast', 2182), ('Comodo', 2182), ('NANO-Antivirus', 2178), ('F-Secure', 2174), ('F-Prot', 2173), ('MaxSecure', 2166), ('Bkav', 2157),	

```
('Zillya', 2147),
('Ikarus', 2144),
('ClamAV', 2135),
('Trapmine', 2135),
('Tencent', 2125),
('Panda', 2105),
('McAfee-GW-Edition', 2088),
('ALYac', 2066),
('TrendMicro', 2016),
('TrendMicro-HouseCall', 2011),
('TotalDefense', 1918),
('TACHYON', 1911),
('eGambit', 1824),
('Malwarebytes', 1717),
('Zoner', 1640),
('Baidu', 1615),
('SUPERAntiSpyware', 1534),
('Symantec', 1529),
('Yandex', 965),
('Jiangmin', 669),
('CAT-QuickHeal', 580),
('Webroot', 445),
('ViRobot', 356),
('CMC', 205),
('Kingsoft', 71),
('Paloalto', 30),
('Alibaba', 22),
('AegisLab', 15),
('Avast-Mobile', 0)
```

Take a look at Symantec (now called NortonLifeLock). How did it do at detecting the malware? Give your assessment of Symantec's performance.

Score	**/ 2 pts**
Symantec has detected 66.6% correctly of the 2295 reports. Since the definitions changes from company to company on viruses and database also differs so few company provide better result.	

Last thing - delete the WindowsVM where the malware was run completely!

Exercise 6: Encryption

What You Need: For this you may use the Kali VM. Be sure to use `python3.9`.

In class we discussed basic encryption techniques. XOR encryption was one such technique. It is basic but malware developers can use it to hide strings from the `strings` utility, and it is still used quite often! In this exercise you will author your own tool to encrypt a string and to then decrypt a string. This activity is inspired by this page:

https://www.codementor.io/@arpitbhayani/deciphering-single-byte-xor-ciphertext-17mtwlzh30

An excellent introduction to the topic. Please implement the logic within - the `single_byte_xor` method - and then decrypt the following string. I used the same key that was used in the article itself - `82`.

Input: `b'\x1f=$7r=<r&=r\x131&;$;&+r`s'`

Answer: Your answer for this problem should be in the following format:

Score	**/ 2 pts**
Unencrypted Text	b'Move on to Activity 2!'

The prior activity had you create an encryption method and decrypt a string. Here is an encrypted string, XOR encrypted. However, this time you do not have the key. You must identify what key encrypted the encrypted string - the code you implemented earlier will take you halfway there:

```
\x00\x1c\x1d\x07t\x1d\x07t\x00\x1c\x11t\x15\x1a\x07\x03\x11\x06t\x00\x1bt\x11\x0c\x11\x06\x17\x1d\x07\x11t\x1b\x1a\x11xt\x18\x15\x16tfz
```

Answer: Your answer for this problem should be in the following format:

Score	**/ 4 pts**
ey Used for Encryption	84
Unencrypted Text	b'THIS IS THE ANSWER TO EXERCISE ONE, LAB 2.'

Final Note

Once you are done, delete this VM and remove it completely (deleting all underlying files).