

PROJ 4 : Basic Dynamic Techniques

Prof. Alberto LaCava

CY -640

MALWARE ANALYSIS & DEFENSE

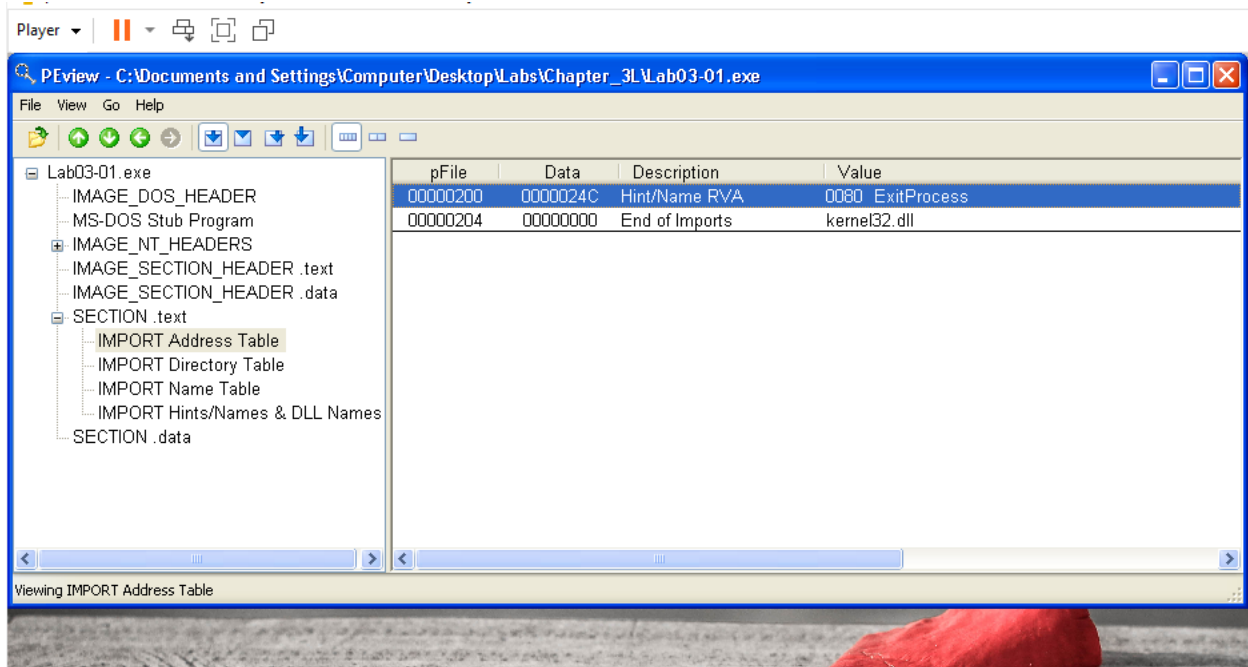
Nikhil Patel

09/20/2022

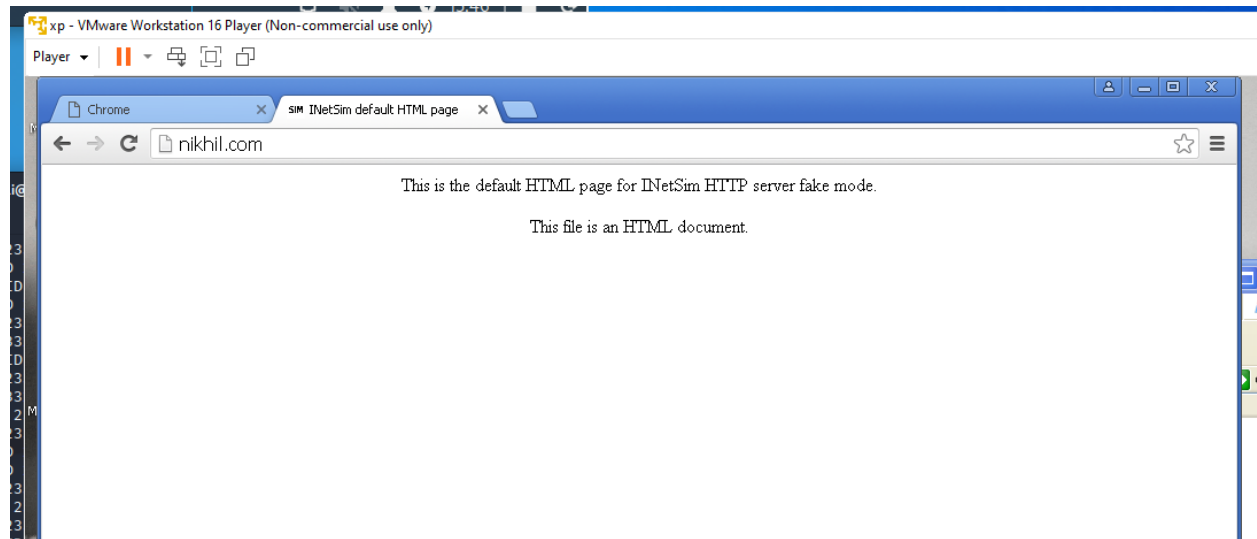
Introduction : In this lab, we will learn about basic dynamic approaches utilizing some basic tools such as PView, Strings, and also setup the connection between virtual machines using INetSim as done in project 3, after which we will monitor the Lab03-01.exe process and finally view the INetSim logs.

Procedure :

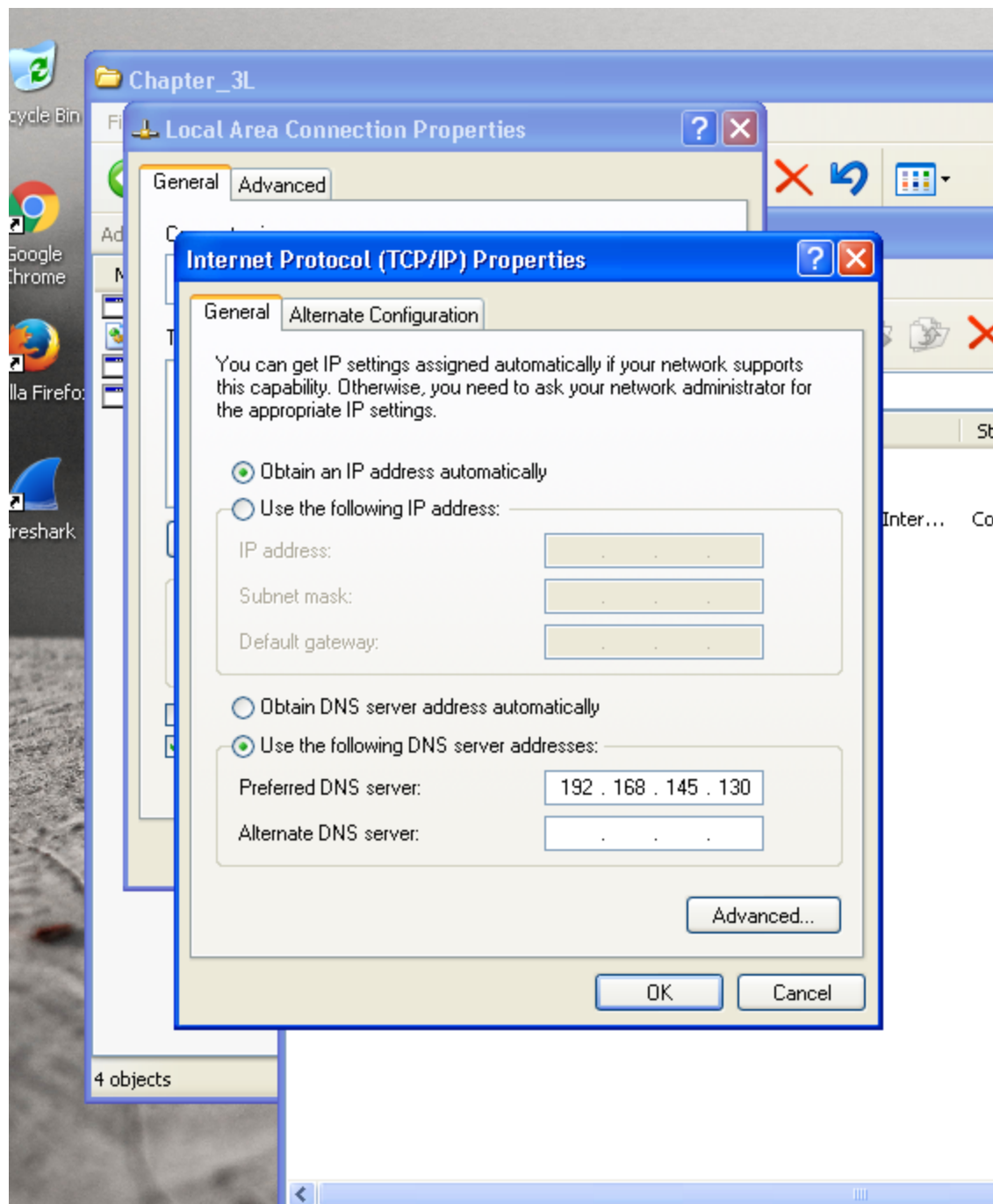
1. Open a virtual machine and open the PView application from the start menu.
2. Go to file and open a file “**Lab03-01.exe**” to check the data value.



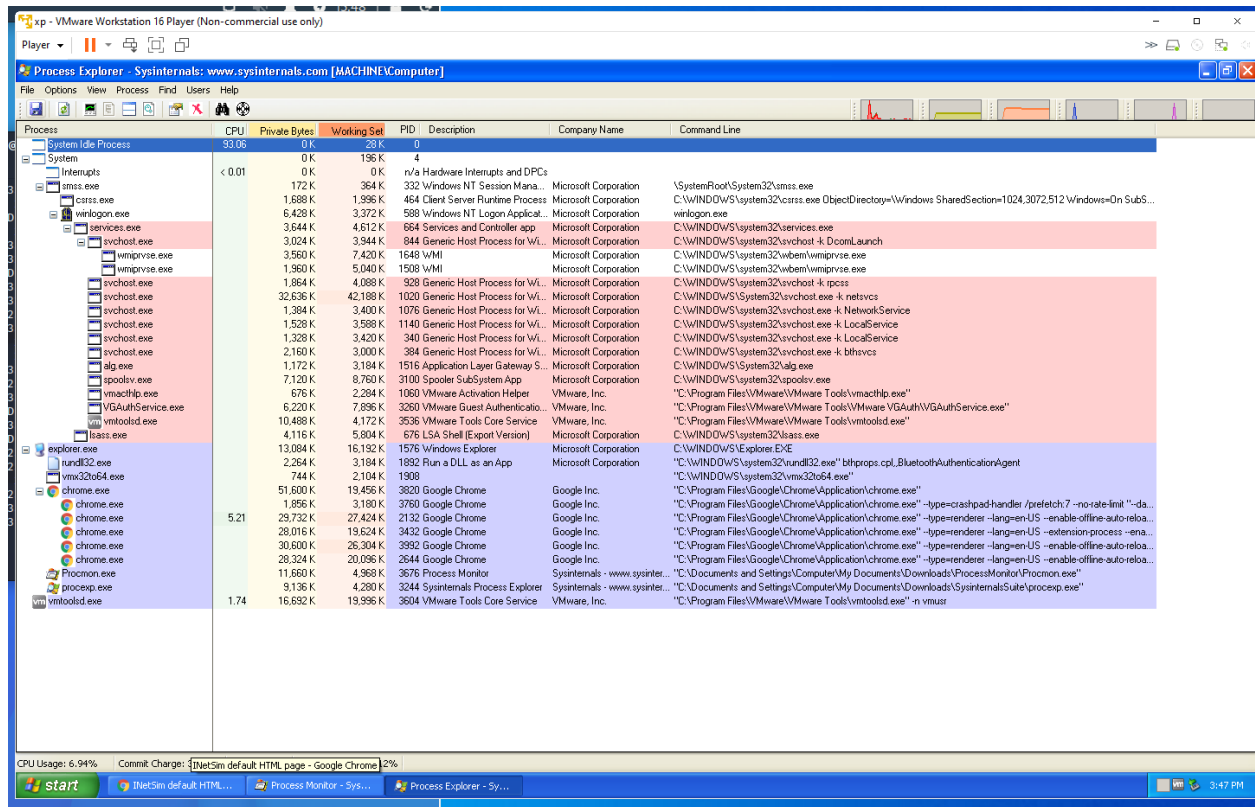
3. Now open the **strings** in the command prompt.
4. Examine the “**Lab03-01.exe**” and find the string above the advpack and take a screenshot.
5. Now prepare for **Dynamic Analysis** :
 - a. Set up **INetSim** to simulate the internet :
 - i. Go to the browser and “**http:NIKHIL.com**” if you see the default html and you are going in the right direction.



- b. Setting the **DNS server** :
 - i. Go to the *Network Connections* and right click on the *LAN* and click *properties*.
 - ii. Double click on *internet protocol (TCP/IP)*.

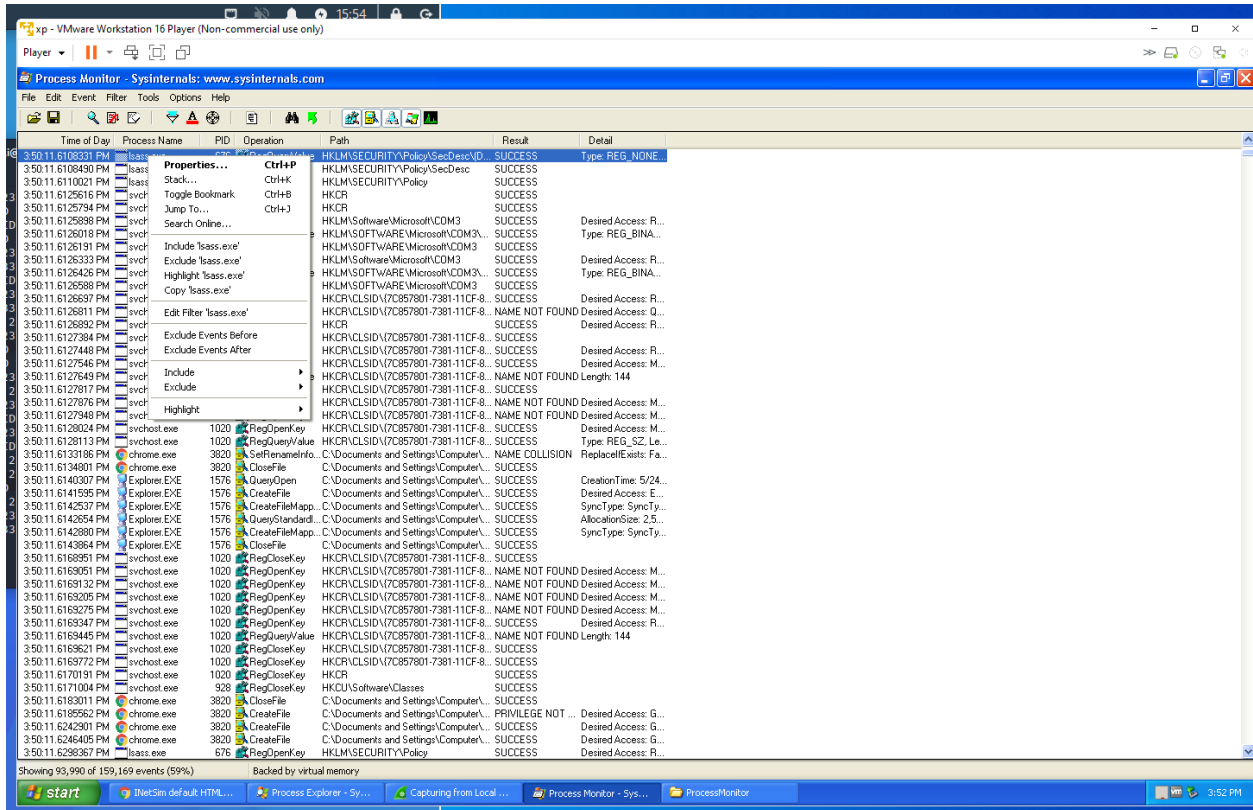


- c. Run the *Process Explorer* from the start menu or desktop.



- d. Run the *wireshark* and click *start* to capture the packets from the interface which goes to the linux machine.
- e. Start *Process monitor* and double click on *Procmon.exe* in the folder you unzipped process monitor.

6. Now we will excluding harmless process such *Isass.exe*, *svchost.exe*, *vmtoolsd.exe*, ..., *setup.exe*. “Repeat until all the processes are hidden.”



7. Now run the Lab03-01.exe by double clicking the file.

- Go to Process Explorer, click view in the lower panel view handlers. You will see the WinVMX32 mutant. Click on view “lower panel view”, DLLs.

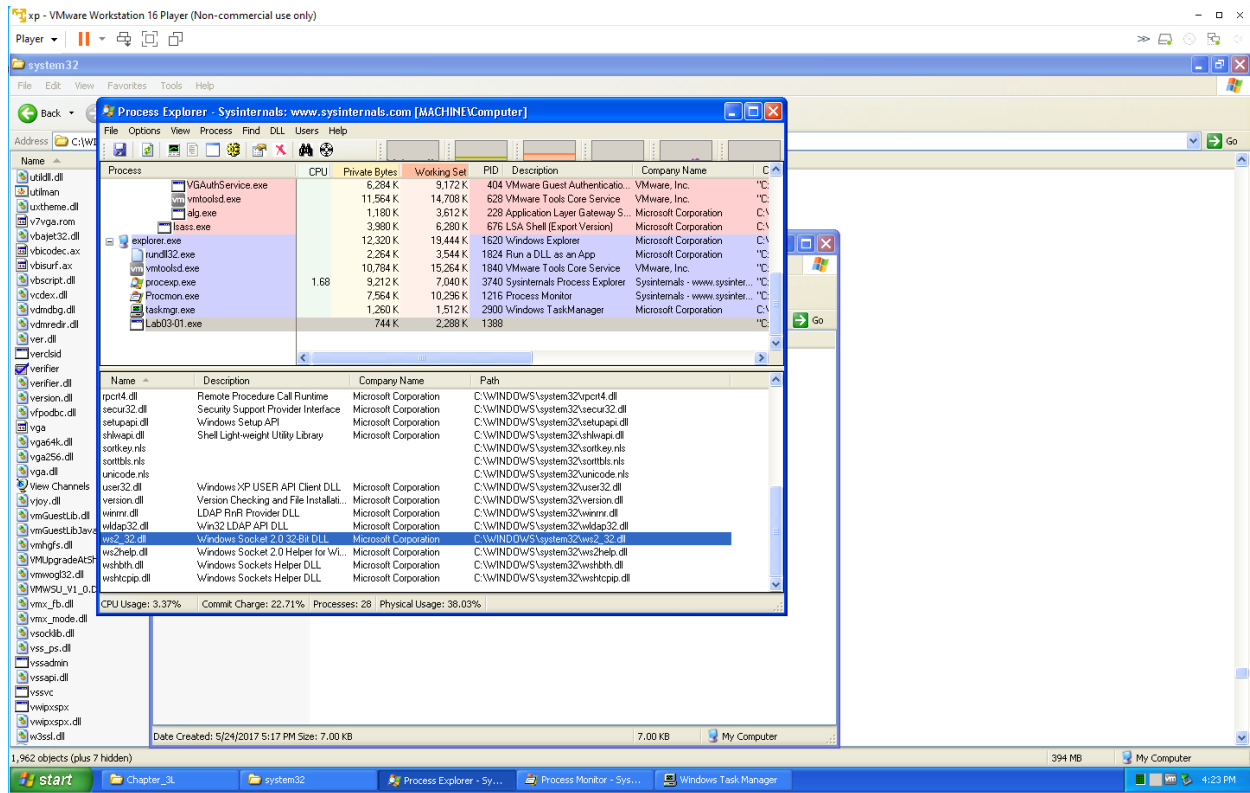
The screenshot shows the Process Explorer window from Sysinternals. The top pane displays a list of running processes with columns for Process, CPU, Private Bytes, Working Set, PID, Description, and Company Name. The bottom pane shows a tree view of system objects, including File, Key, KeyedEvent, Mutant, Semaphore, Thread, and WindowStation. The 'Mutant' object is highlighted, showing its name as '\BaseNamedObjects\WinVMX32'.

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
VGAuthService.exe		6,284 K	9,172 K	404	VMware Guest Authentication...	VMware, Inc.
vmtoolsd.exe	1.74	11,576 K	14,716 K	628	VMware Tools Core Service	VMware, Inc.
alg.exe		1,180 K	3,612 K	228	Application Layer Gateway S...	Microsoft Corporation
lsass.exe		4,012 K	6,288 K	676	LSA Shell (Export Version)	Microsoft Corporation
explorer.exe		12,448 K	19,472 K	1620	Windows Explorer	Microsoft Corporation
rundll32.exe		2,264 K	3,544 K	1824	Run a DLL as an App	Microsoft Corporation
vmtoolsd.exe		7,528 K	11,784 K	1840	VMware Tools Core Service	VMware, Inc.
procexp.exe		9,220 K	6,816 K	3740	Sysinternals Process Explorer	Sysinternals - www.sysinter...
Procmon.exe		7,556 K	10,176 K	1216	Process Monitor	Sysinternals - www.sysinter...
taskmgr.exe		1,260 K	1,508 K	2900	Windows TaskManager	Microsoft Corporation
Lab03-01.exe		744 K	2,288 K	1388		

Type	Name
File	\Device\Tcp
Key	HKLM
Key	HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\Protocol_Catalog9
Key	HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\NameSpace_Catalog5
Key	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage
Key	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters
Key	HKLM\SYSTEM\ControlSet001\Services\NetBT\Parameters\Interfaces
Key	HKLM\SYSTEM\ControlSet001\Services\NetBT\Parameters
KeyedEvent	\KernelObjects\CritSecOutOfMemoryEvent
Mutant	\BaseNamedObjects\WinVMX32
Semaphore	\BaseNamedObjects\shell.{A48F1A32-A340-11D1-BC6B-00A0C90312E1}
Thread	Lab03-01.exe(1388): 280
Thread	Lab03-01.exe(1388): 280
WindowStation	\Windows\WindowStations\WinSta0
WindowStation	\Windows\WindowStations\WinSta0

CPU Usage: 1.74% Commit Charge: 22.59% Processes: 29 Physical Usage: 37.91%

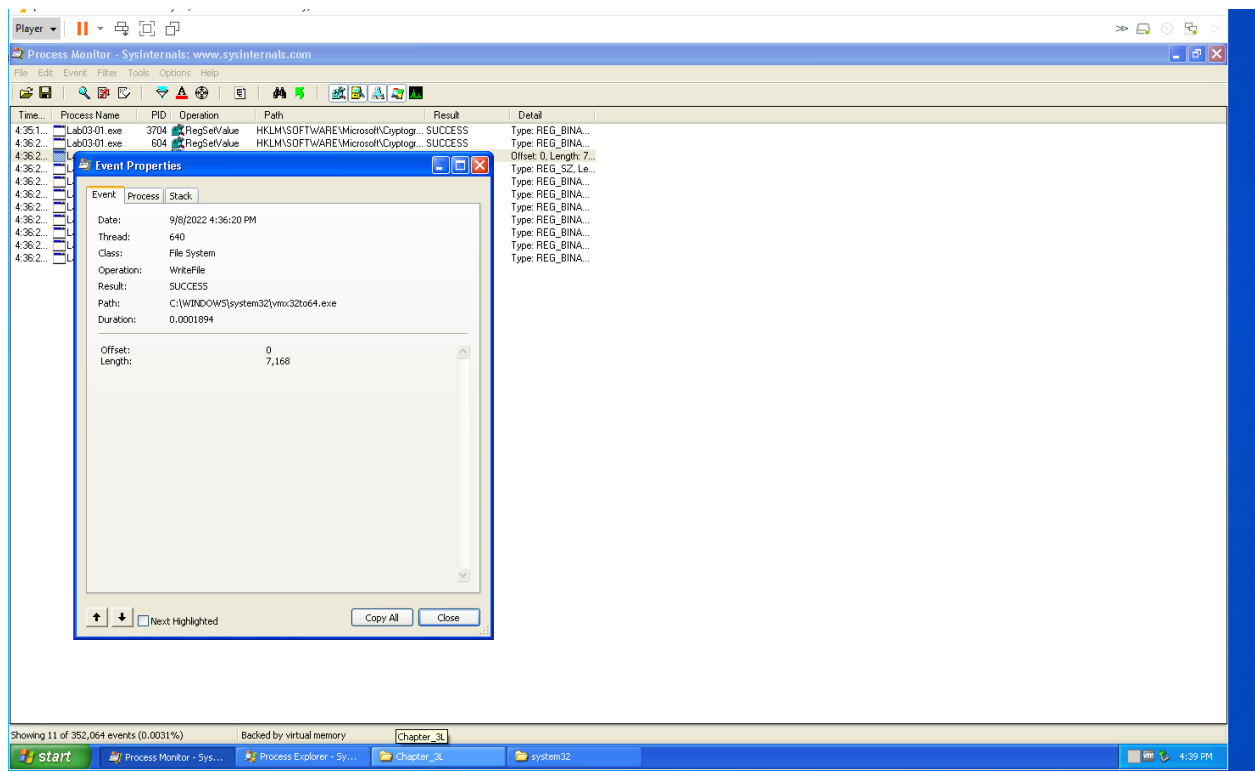
9. Search for ws2_32.dll and wshtcpip.dll and take a screenshot.



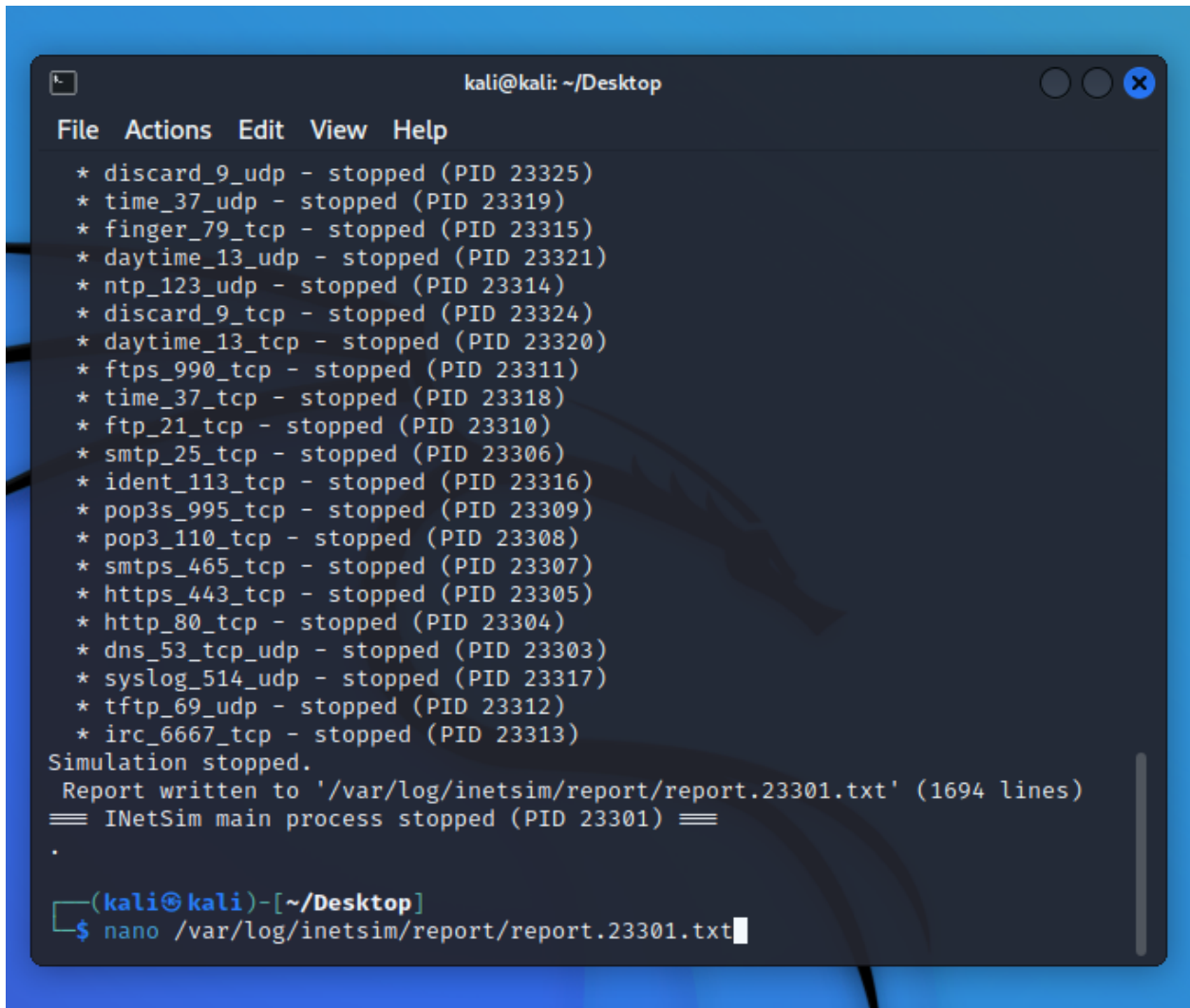
10. Now viewing the malicious Process's Events in the process monitor.

- Go to the filter and process name "**Lab03-01.exe**" included.
- Add two more filters :
 - Operation of RegSetValue
 - Operation of WriteFile
- Click ok, so that you will end up with two events.

11. Double click the event ending with vm32to64.exe and expand the details and then take screenshot.



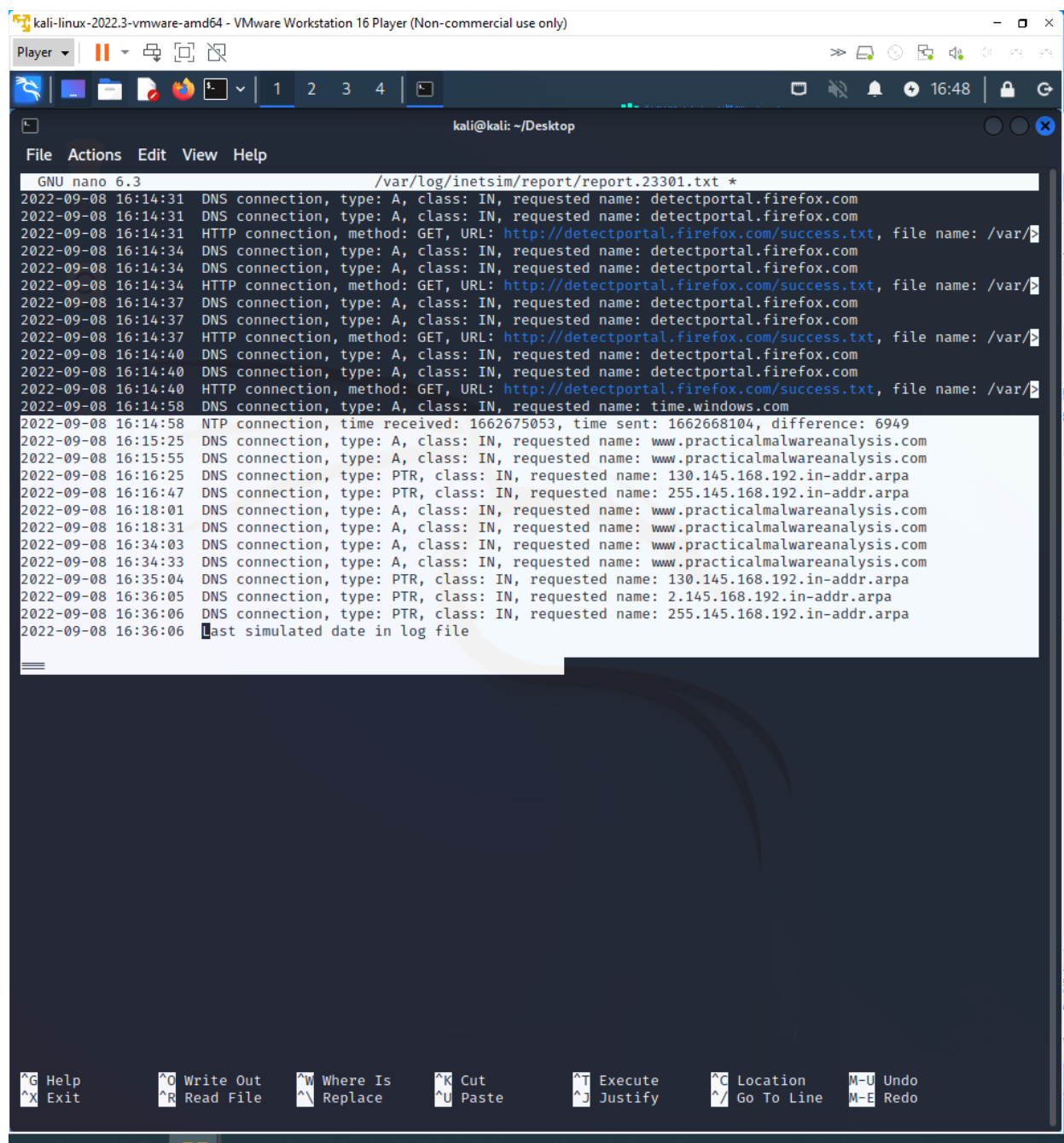
12. Now we will view INetSim logs in kali linux machine, press ctrl + c in the terminal where INetSim was running.
13. Now type ***nano /var/log/inetsim/report/report.3384.txt***.



The screenshot shows a terminal window titled 'kali@kali: ~/Desktop'. The window contains the output of an INetSim simulation. The output lists 20 services that have stopped, each with its name and PID. The services are: discard_9_udp (PID 23325), time_37_udp (PID 23319), finger_79_tcp (PID 23315), daytime_13_udp (PID 23321), ntp_123_udp (PID 23314), discard_9_tcp (PID 23324), daytime_13_tcp (PID 23320), ftps_990_tcp (PID 23311), time_37_tcp (PID 23318), ftp_21_tcp (PID 23310), smtp_25_tcp (PID 23306), ident_113_tcp (PID 23316), pop3s_995_tcp (PID 23309), pop3_110_tcp (PID 23308), smtps_465_tcp (PID 23307), https_443_tcp (PID 23305), http_80_tcp (PID 23304), dns_53_tcp_udp (PID 23303), syslog_514_udp (PID 23317), tftp_69_udp (PID 23312), and irc_6667_tcp (PID 23313). Below the list, it says 'Simulation stopped.' and 'Report written to '/var/log/inetsim/report/report.23301.txt' (1694 lines)'. It also shows 'INetSim main process stopped (PID 23301)'. At the bottom, the terminal prompt is '(kali@kali)-[~/Desktop]' and the command '\$ nano /var/log/inetsim/report/report.23301.txt' is entered.

```
kali@kali: ~/Desktop
File Actions Edit View Help
* discard_9_udp - stopped (PID 23325)
* time_37_udp - stopped (PID 23319)
* finger_79_tcp - stopped (PID 23315)
* daytime_13_udp - stopped (PID 23321)
* ntp_123_udp - stopped (PID 23314)
* discard_9_tcp - stopped (PID 23324)
* daytime_13_tcp - stopped (PID 23320)
* ftps_990_tcp - stopped (PID 23311)
* time_37_tcp - stopped (PID 23318)
* ftp_21_tcp - stopped (PID 23310)
* smtp_25_tcp - stopped (PID 23306)
* ident_113_tcp - stopped (PID 23316)
* pop3s_995_tcp - stopped (PID 23309)
* pop3_110_tcp - stopped (PID 23308)
* smtps_465_tcp - stopped (PID 23307)
* https_443_tcp - stopped (PID 23305)
* http_80_tcp - stopped (PID 23304)
* dns_53_tcp_udp - stopped (PID 23303)
* syslog_514_udp - stopped (PID 23317)
* tftp_69_udp - stopped (PID 23312)
* irc_6667_tcp - stopped (PID 23313)
Simulation stopped.
Report written to '/var/log/inetsim/report/report.23301.txt' (1694 lines)
== INetSim main process stopped (PID 23301) ==
.
(kali@kali)-[~/Desktop]
$ nano /var/log/inetsim/report/report.23301.txt
```

14. Now we will see whether a network packet has been captured or not.
15. Go to Wireshark and click stop capture.
16. In the filter option type frame contains practicalmalwareanalysis, and hit enter.
17. Check for TCP stream in protocol and double click to follow the tcp stream.
18. At last take a screenshot of the Stream you follow.



The screenshot shows a Kali Linux virtual machine running in VMware Workstation 16. The terminal window is open to the nano text editor, editing the file `/var/log/inetsim/report/report.23301.txt`. The file contains a log of network connections. The log entries are as follows:

```
GNU nano 6.3 /var/log/inetsim/report/report.23301.txt *
2022-09-08 16:14:31 DNS connection, type: A, class: IN, requested name: detectportal.firefox.com
2022-09-08 16:14:31 DNS connection, type: A, class: IN, requested name: detectportal.firefox.com
2022-09-08 16:14:31 HTTP connection, method: GET, URL: http://detectportal.firefox.com/success.txt, file name: /var/
2022-09-08 16:14:34 DNS connection, type: A, class: IN, requested name: detectportal.firefox.com
2022-09-08 16:14:34 HTTP connection, method: GET, URL: http://detectportal.firefox.com/success.txt, file name: /var/
2022-09-08 16:14:37 DNS connection, type: A, class: IN, requested name: detectportal.firefox.com
2022-09-08 16:14:37 HTTP connection, method: GET, URL: http://detectportal.firefox.com/success.txt, file name: /var/
2022-09-08 16:14:40 DNS connection, type: A, class: IN, requested name: detectportal.firefox.com
2022-09-08 16:14:40 HTTP connection, method: GET, URL: http://detectportal.firefox.com/success.txt, file name: /var/
2022-09-08 16:14:58 DNS connection, type: A, class: IN, requested name: time.windows.com
2022-09-08 16:14:58 NTP connection, time received: 1662675053, time sent: 1662668104, difference: 6949
2022-09-08 16:15:25 DNS connection, type: A, class: IN, requested name: www.practicalmalwareanalysis.com
2022-09-08 16:15:55 DNS connection, type: A, class: IN, requested name: www.practicalmalwareanalysis.com
2022-09-08 16:16:25 DNS connection, type: PTR, class: IN, requested name: 130.145.168.192.in-addr.arpa
2022-09-08 16:16:47 DNS connection, type: PTR, class: IN, requested name: 255.145.168.192.in-addr.arpa
2022-09-08 16:18:01 DNS connection, type: A, class: IN, requested name: www.practicalmalwareanalysis.com
2022-09-08 16:18:31 DNS connection, type: A, class: IN, requested name: www.practicalmalwareanalysis.com
2022-09-08 16:34:03 DNS connection, type: A, class: IN, requested name: www.practicalmalwareanalysis.com
2022-09-08 16:34:33 DNS connection, type: A, class: IN, requested name: www.practicalmalwareanalysis.com
2022-09-08 16:35:04 DNS connection, type: PTR, class: IN, requested name: 130.145.168.192.in-addr.arpa
2022-09-08 16:36:05 DNS connection, type: PTR, class: IN, requested name: 2.145.168.192.in-addr.arpa
2022-09-08 16:36:06 DNS connection, type: PTR, class: IN, requested name: 255.145.168.192.in-addr.arpa
2022-09-08 16:36:06 Last simulated date in log file
```

The terminal window has a menu bar with File, Actions, Edit, View, and Help. The status bar at the bottom shows various keyboard shortcuts for nano editor functions like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, and Redo.

Conclusion : To conclude the lab, we learned about dynamic techniques and used tools like PEvent to acquire the data value of the current project file we were working on. In addition, we learnt how to use the process monitor to examine a malicious process in the events and how to add a new process to an event to receive the logs in INetSim.