Practical No: 01

Aim: Encrypting and Decrypting Data (Hacker Tools & OpenSSL)

Steps:

Step 1: Setting Up the Virtual Machine

Open the D: drive.
Navigate to the SOC folder.
Inside the SOC folder, create a new folder with any name (e.g., MyWork).
Double-click to start the virtual machine: Cybersecurity_Lab_VM_Workstation_2023.
Change the path to the folder you created in Step 3.
Increase the RAM to 4096 MB.
Power on the new Cybersecurity VM (It should appear as the third one).
When prompted, log in with the following credentials:

- Username: SOC analyst
- Password: (will be provided during the session)

Step 2: Working with the Terminal

Note: The '$' symbol indicates a non-root user (not in admin mode).
$ pw
pwd (Print Working Directory) displays your current path.

```
[analyst@secOps ~]$ pwd
/home/analyst
```

$ mkdir zipfiles
$ cd zipfiles/
Creates and navigates into a new directory named 'zipfiles'.
$ echo This is a sample file1 > sample1.txt
$ echo This is a sample file2 > sample2.txt
$ echo This is a sample file3 > sample3.txt
$ echo This is a sample file4 > sample4.txt

```
[analyst@secOps ~]$ mkdir zipfiles
[analyst@secOps ~]$ cd zipfiles/
[analyst@secOps zipfiles]$ echo This is sample file1 >sample1.txt
[analyst@secOps zipfiles]$ echo THis is sample file2 >sample2.txt
[analyst@secOps zipfiles]$ echo THis is sample file3 >sample3.txt
```

The 'echo' command is used to write content into a file. You can also use 'touch' or the GUI to create files.
$ ls -l

```
[analyst@secOps zipfiles]$ ls -l
total 12
-rw-rw-r-- 1 analyst analyst 21 Jul 31 02:57 sample1.txt
-rw-rw-r-- 1 analyst analyst 21 Jul 31 02:58 sample2.txt
-rw-rw-r-- 1 analyst analyst 21 Jul 31 02:58 sample3.txt
```

Lists files with detailed information.

Step 3: Creating Encrypted Zip Files

$ zip -e file1.zip sample*
Password: a
$ zip -e file2.zip sample*
Password: a1
$ zip -e file3.zip sample*
Password: a1@

$ zip -e file4.zip sample*
Password: a1@b
$ zip -e file5.zip sample*
Password: a1@b2
$ zip -e file6.zip sample*
Password: a1@b2!

```
[analyst@secOps zipfiles]$ zip -e file1.zip
Enter password:

zip error: Invalid command arguments (zero length password not allowed)
[analyst@secOps zipfiles]$ zip -e file1.zip sample*
Enter password:
Verify password:
  adding: sample1.txt (stored 0%)
  adding: sample2.txt (stored 0%)
  adding: sample3.txt (stored 0%)
  adding: sample4.txt (stored 0%)
[analyst@secOps zipfiles]$ zip -e file2.zip sample*
Enter password:
Verify password:
  adding: sample1.txt (stored 0%)
  adding: sample2.txt (stored 0%)
  adding: sample3.txt (stored 0%)
```

The '-e' option enables encryption for the zip file.

Step 4: Cracking Zip Passwords with fcrackzip
$ man fcrackzip

```
[analyst@secOps ~]$ man fcrackzip
[analyst@secOps ~]$
```

```
Terminal
File  Edit  View  Search  Terminal  Help
FCRACKZIP(1)                General Commands Manual                FCRACKZIP(1)

NAME
       fcrackzip - a Free/Fast Zip Password Cracker

SYNOPSIS
       fcrackzip   [-bDBchVvplum2]   [--brute-force]  [--dictionary]  [--benchmark]
       [--charset characterset] [--help] [--validate] [--verbose]  [--init-password
       string/path] [--length min-max] [--use-unzip] [--method name] [--modulo r/m]
       file...

DESCRIPTION
       fcrackzip searches each zipfile given for encrypted files and tries to guess
       the  password.  All files must be encrypted with the same password, the more
       files you provide, the better.

   OPTIONS
       -h, --help
              Prints the version number and (hopefully) some helpful insights.

       -v, --verbose
              Each -v makes the program more verbose.
```

Displays the manual/help for the fcrackzip tool.
$ fcrackzip -vul 1-4 file1.zip

```
unknown option
[analyst@secOps zipfiles]$ fcrackzip -vul 1-4 file1.zip
found file 'sample1.txt', (size cp/uc     33/    21, flags 9, chk 1739)
found file 'sample2.txt', (size cp/uc     33/    21, flags 9, chk 1745)
found file 'sample3.txt', (size cp/uc     33/    21, flags 9, chk 174f)
found file 'sample4.txt', (size cp/uc     33/    21, flags 9, chk 180b)


PASSWORD FOUND!!!!: pw == a
```

This command attempts to crack the password using brute force.
Output: pw == a

Step 5: Unzipping Files After Cracking

$ mkdir zip
$ cd zip/
Creates and navigates into a new directory named 'zip'.
$ unzip file1.zip

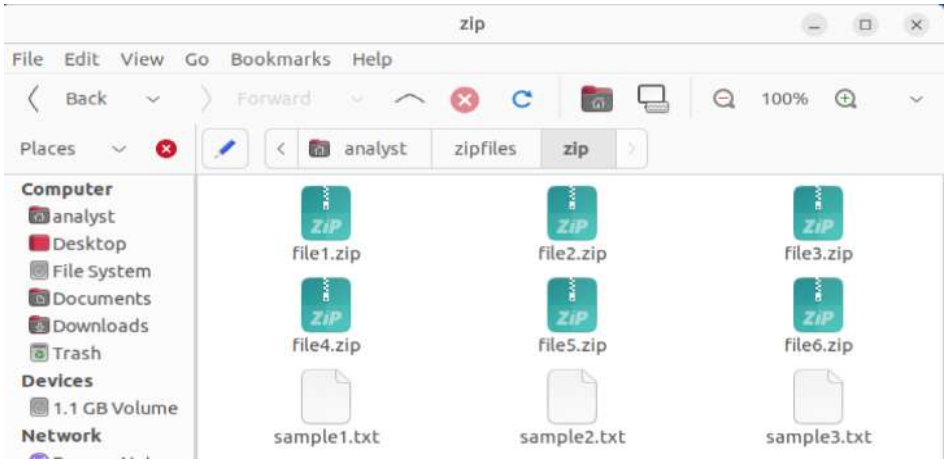
```
[analyst@secOps zipfiles]$ mkdir zip
[analyst@secOps zipfiles]$ cd zip/
[analyst@secOps zip]$ unzip file1.zip
Archive:  file1.zip
[file1.zip] sample1.txt password:
 extracting: sample1.txt
 extracting: sample2.txt
 extracting: sample3.txt
 extracting: sample4.txt
```

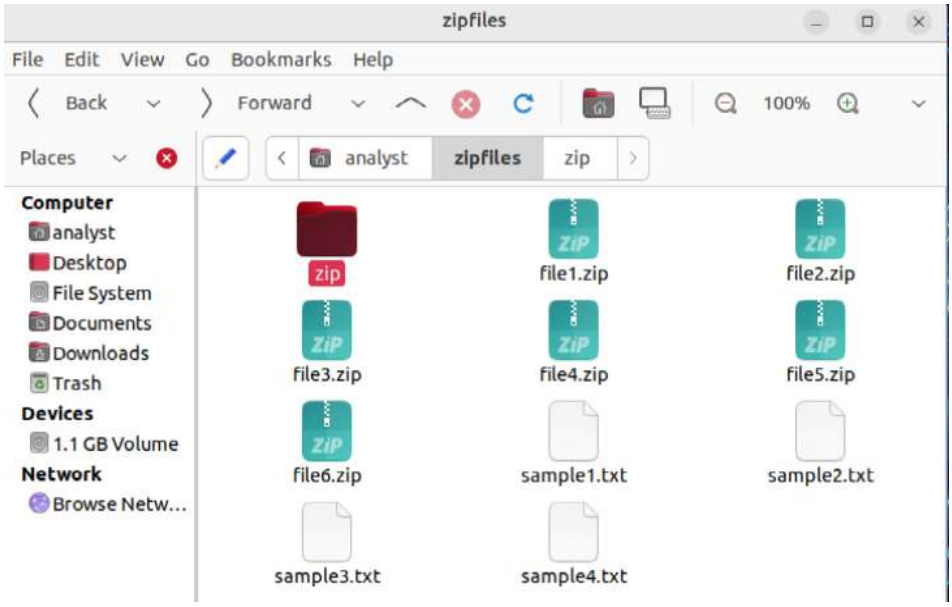
Unzips the cracked file.
$ fcrackzip -vul 1-4 file2.zip

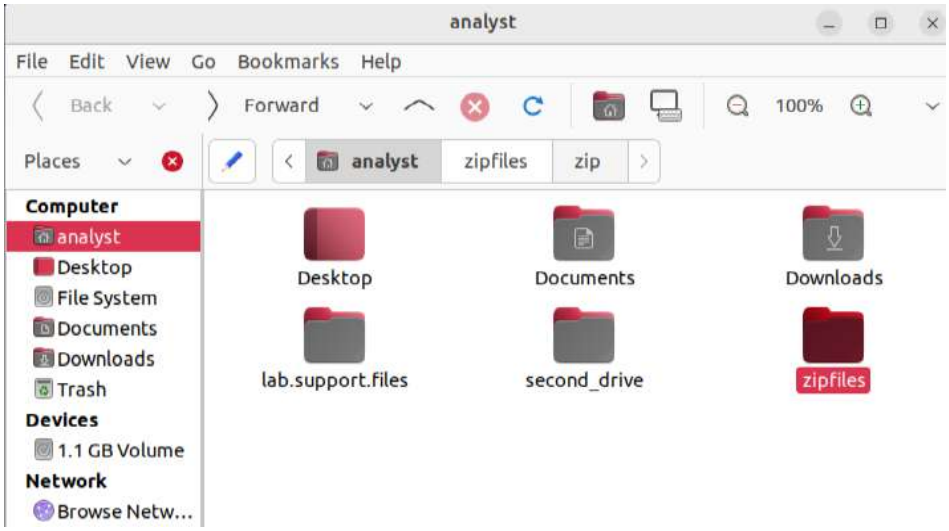
```
[analyst@secOps zip]$ fcrackzip -vul 1-4 file2.zip
found file 'sample1.txt', (size cp/uc     33/    21, flags 9, chk 1739)
found file 'sample2.txt', (size cp/uc     33/    21, flags 9, chk 1745)
found file 'sample3.txt', (size cp/uc     33/    21, flags 9, chk 174f)
found file 'sample4.txt', (size cp/uc     33/    21, flags 9, chk 180b)


PASSWORD FOUND!!!!: pw == a1
```

Output: pw == a1
$ unzip file2.zip

```
[analyst@secOps zip]$ unzip file2.zip
Archive:  file2.zip
[file2.zip] sample1.txt password:
replace sample1.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace sample2.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace sample3.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace sample4.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
```




zipfiles
File Edit View Go Bookmarks Help
Back Forward 100%
Places analyst zipfiles zip
Computer
analyst
Desktop
File System
Documents
Downloads
Trash
Devices
1.1 GB Volume
Network
Browse Netw...
zip
file1.zip
file2.zip
file3.zip
file4.zip
file5.zip
file6.zip
sample1.txt
sample2.txt
sample3.txt
sample4.txt


analyst
File Edit View Go Bookmarks Help
Back Forward 100%
Places analyst zipfiles zip
Computer
analyst
Desktop
File System
Documents
Downloads
Trash
Devices
1.1 GB Volume
Network
Browse Netw...
Desktop
Documents
Downloads
lab.support.files
second_drive
zipfiles

Practical 1B: Encryption and Decryption data using OpenSSL.

Steps:

Step1: Check for the letter _to_gradma.txt in the system

Go to the folder where "letter_to_grandma.txt" is saved. It is saved in folder lab.support.files.

Now we need to open the file in terminal, so for that the following command is

$cd lab.support.files/
$ls
$cat letter_to_grandma.txt

```
[analyst@secOps ~]$ cd lab.support.files/
[analyst@secOps lab.support.files]$ ls
apache_in_epoch.log        instructor             pcaps
applicationX_in_epoch.log  letter_to_grandma.txt  pox
attack_scripts             logstash-tutorial.log  sample.img
confidential.txt           long_commands          sample.img_SHA256.sig
cyops.mn                   malware                scripts
elk_services               mininet_services       SQL_Lab.pcap
h2_dropbear.banner         openssl_lab
[analyst@secOps lab.support.files]$ cat letter_to_grandma.txt
Hi Grandma,
I am writing this letter to thank you for the chocolate chip cookies you sent me. I got th
 I have already eaten half of the box! They are absolutely delicious!

I wish you all the best. Love,
Your cookie-eater grandchild.
```

cd = command navigates to the folder
ls = command lists down everything that exists in the folder
cat = command is used to view the content of the file

Step2: Encrypt the file and verify

```
openssl enc -aes-256-cbc -in letter_to_grandma.txt -out
letter_to_grandma.enc
```

```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -in letter_to_grandma.txt -out message.enc
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

You will be prompted to enter and confirm a password and after entering the password it will save the encrypted file.

```
$ls
$cat message.enc
```

```
[analyst@secOps lab.support.files]$ ls
apache_in_epoch.log        instructor             openssl_lab
applicationX_in_epoch.log  letter_to_grandma.txt  pcaps
attack_scripts             logstash-tutorial.log  pox
confidential.txt           long_commands          sample.img
cyops.mn                   malware                sample.img_SHA256.sig
elk_services               message.enc            scripts
h2_dropbear.banner         mininet_services       SQL_Lab.pcap
[analyst@secOps lab.support.files]$ cat message.enc
Salted__◆◆◆-◆U◆◆>>◆J◆◆@◆j◆◆◆◆u◆◆g◆◆8Bu=l◆◆x◆6u7x◆c`L◆◆~Y◆M◆◆◆
O◆◆N◆◆◆%◆◆◆aad$-w◆◆H◆[◆I[◆
                         ◆w◆◆◆◆;$◆;◆◆◆◆◆◆◆#J◆◆Qp_
D6%
```

Step 3: Decrypt the file and verify

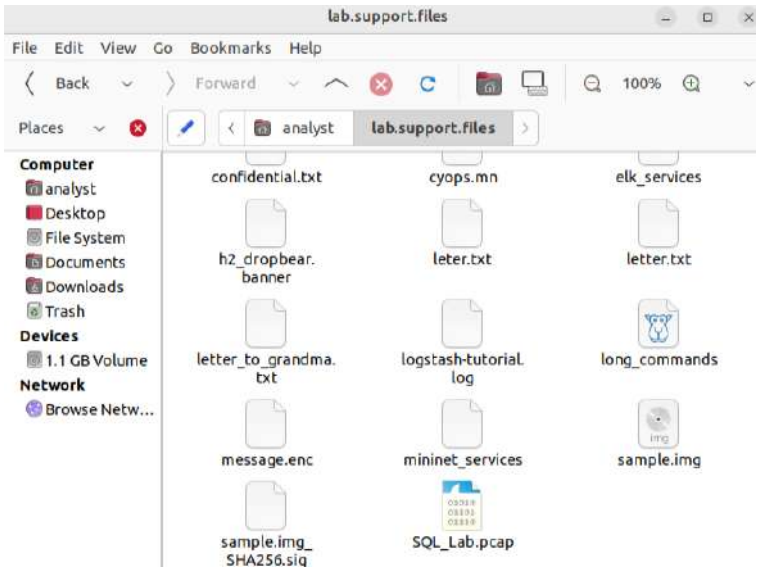
```
openssl enc -aes-256-cbc -d -in letter_to_grandma.enc -out
letter_to_grandma_decrypted.txt
```

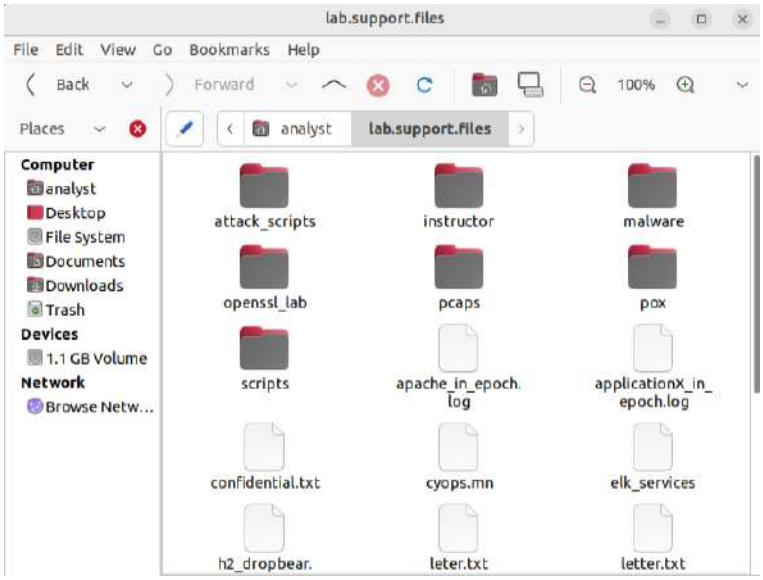
```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -d -in message.enc -out letter.txt
enter AES-256-CBC decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[analyst@secOps lab.support.files]$ ls
apache_in_epoch.log        elk_services           logstash-tutorial.log  openssl_lab            scripts
applicationX_in_epoch.log  h2_dropbear.banner     long_commands          pcaps                  SQL_Lab.pcap
attack_scripts             instructor             malware                pox
confidential.txt           letter_to_grandma.txt  message.enc            sample.img
cyops.mn                   letter.txt             mininet_services       sample.img_SHA256.sig
```

Enter the same password used during encryption.
$cat letter.txt

```
[analyst@secOps lab.support.files]$ cat letter.txt
Hi Grandma,
I am writing this letter to thank you for the chocolate chip cookies you sent me. I got them this morning and
 I have already eaten half of the box! They are absolutely delicious!

I wish you all the best. Love,
Your cookie-eater grandchild.
```

Output:





Practical 1C: Hashing a text file with OpenSSL and verifying hashes.

Steps:

Step1: Navigate to the file location, Change directory to the location containing the target file

```
$cd lab.support.files/
$ls
$cat letter_to_grandma.txt
```

```
[analyst@secOps ~]$ cd lab.support.files/
[analyst@secOps lab.support.files]$ ls
apache_in_epoch.log        instructor             pcaps
applicationX_in_epoch.log  letter_to_grandma.txt  pox
attack_scripts             logstash-tutorial.log  sample.img
confidential.txt           long_commands          sample.img_SHA256.sig
cyops.mn                   malware                scripts
elk_services               mininet_services       SQL_Lab.pcap
h2_dropbear.banner         openssl_lab
[analyst@secOps lab.support.files]$ cat letter_to_grandma.txt
Hi Grandma,
I am writing this letter to thank you for the chocolate chip cookies you sent me
. I got them this morning and I have already eaten half of the box! They are abs
olutely delicious!

I wish you all the best. Love,
Your cookie-eater grandchild.
```

navigates to the lab.support.files directory, lists its contents, and displays the text of letter_to_grandma.txt to verify the file before processing.

Step2: Make changes and verify those changes in those hash values

1. Generate a hash (Example using SHA256)

```
$openssl sha256 letter_to_grandma.txt
This outputs the SHA256 hash value of the file.
```

2. Verify the hashes

```
After making certain changes save it as letter_to_grandpa
$openssl sha512 letter_to_grandma.txt
$openssl sha256 letter_to_grandpa.txt
$openssl sha512 letter_to_grandma.txt
$ls -l
```

```
[analyst@secOps lab.support.files]$ openssl sha256 letter_to_grandma.txt
SHA256(letter_to_grandma.txt)= deff9c9bbece44866796ff6cf21f2612fbb77aa1b2515a900
bafb29be118080b
[analyst@secOps lab.support.files]$ openssl sha512 letter_to_grandma.txt
SHA512(letter_to_grandma.txt)= e2d5bc8161ef34765300379c357ecf5e732cc317f1c62c713
568bb47580fe503e3523e790e85d2cdae2625e8e8db6725a1342a2faa8140c9057d552828ebb93a
[analyst@secOps lab.support.files]$ openssl sha256 letter_to_grandpa.txt
SHA256(letter_to_grandpa.txt)= deff9c9bbece44866796ff6cf21f2612fbb77aa1b2515a900
bafb29be118080b
```

generate SHA256 and SHA512 hashes for letter_to_grandma.txt and show that letter_to_grandpa.txt has an identical SHA256 hash, indicating no changes between the two files

```
[analyst@secOps lab.support.files]$ openssl sha512 letter_to_grandpa.txt
SHA512(letter_to_grandpa.txt)= e2d5bc8161ef34765300379c357ecf5e732cc317f1c62c713
568bb47580fe503e3523e790e85d2cdae2625e8e8db6725a1342a2faa8140c9057d552828ebb93a
[analyst@secOps lab.support.files]$ openssl sha256 letter_to_grandpa.txt
SHA256(letter_to_grandpa.txt)= 43302c4500b7c4b8e574ba27a59d83267812493c029fd054c
9242f3ac73100bc
[analyst@secOps lab.support.files]$ openssl sha512 letter_to_grandpa.txt
SHA512(letter_to_grandpa.txt)= 7c35db79a06aa30ae0f6de33f2322fd419560ee9af9cedeb6
e251f2f1c4e99e0bbe5d2fc32ce501468891150e3be7e288e3e568450812980c9f8288e3103a1d3
[analyst@secOps lab.support.files]$ openssl sha256 letter_to_grandpa.txt
SHA256(letter_to_grandpa.txt)= 663a781df0ee5d88d74b7ec001f79c48b8a1744fd80066881
9c6a3bc1b1ac48f
```

```
[analyst@secOps lab.support.files]$ openssl sha512 letter_to_grandpa.txt
SHA512(letter_to_grandpa.txt)= c63ec8dcfaaa11470c5aace0df81926d8ee9a05a6433a9b0f
442bb7e8b0470b7a31e8eaacf8527892abac6369a19d3df51812d6343740f32b8b157fdf1b2998d
[analyst@secOps lab.support.files]$
```

Multiple SHA256 and SHA512 hash commands are run on letter_to_grandpa.txt, resulting in different hash values each time, which indicates the file contents are being modified between checks

If both match, the file is unchanged; if not, the file has been modified.

```
[analyst@secOps lab.support.files]$ ls -l
total 600
-rw-r--r-- 1 analyst analyst    649 Jan 12  2023 apache_in_epoch.log
-rw-r--r-- 1 analyst analyst    126 Jan 12  2023 applicationX_in_epoch.log
drwxr-xr-x 4 analyst analyst   4096 Jan 12  2023 attack_scripts
-rw-r--r-- 1 analyst analyst    102 Jan 12  2023 confidential.txt
-rw-r--r-- 1 analyst analyst   2871 Jan 12  2023 cyops.mn
-rw-r--r-- 1 analyst analyst     75 Jan 12  2023 elk_services
-rw-r--r-- 1 analyst analyst    373 Jan 12  2023 h2_dropbear.banner
drwxr-xr-x 2 analyst analyst   4096 Jan 12  2023 instructor
-rw-rw-r-- 1 analyst analyst    255 Aug  7 03:06 leter.txt
-rw-rw-r-- 1 analyst analyst    255 Aug  7 03:46 letter_to_grandma.txt
-rw-rw-r-- 1 analyst analyst    253 Aug  7 04:08 letter_to_grandpa.txt
-rw-rw-r-- 1 analyst analyst    255 Aug  7 03:04 letter.txt
-rw-r--r-- 1 analyst analyst  24464 Jan 12  2023 logstash-tutorial.log
-rwxr-x--- 1 analyst analyst    486 Jan 12  2023 long_commands
drwxr-xr-x 2 analyst analyst   4096 Jan 12  2023 malware
-rw-rw-r-- 1 analyst analyst    370 Aug  7 02:50 message.enc
-rwxr-xr-x 1 analyst analyst    172 Jan 12  2023 mininet_services
drwxr-xr-x 2 analyst analyst   4096 Jan 12  2023 openssl_lab
```

lists all files and directories in lab.support.files with detailed information including permissions, size, owner, and modification date.

3. Verify the sample file

```
$openssl sha256 sample.img
$openssl sha256 sample.img_SHA256.sig
$openssl sha512 sample.img
$openssl sha512 sample.img_SHA256.sig
```

```
[analyst@secOps lab.support.files]$ openssl sha256 sample.img
SHA256(sample.img)= c56c4724c26eb0157963c0d62b76422116be31804a39c82fd44ddf0ca501
3e6a
[analyst@secOps lab.support.files]$ openssl sha256 sample.img_SHA256.sig
SHA256(sample.img_SHA256.sig)= 14df2685a92ec27482c4942da143b04e18aec1c27085b0e3b
b87111b7a9e86f3
[analyst@secOps lab.support.files]$ openssl sha512 sample.img
SHA512(sample.img)= ea2833ce3404ed1a44072b148ae84777845794d8ea2f13149ce056d1792d
a4b2e440c992387077cb2a4ec030bcfcfce56ae99f6b68fd0aad3599d9bcfbe4c2b8
[analyst@secOps lab.support.files]$ openssl sha512 sample.img_SHA256.sig
SHA512(sample.img_SHA256.sig)= 71a00d72d30cde99f56f3b5fba344f88c510403fb5eca467c
b1f9170c5ccebbc02957898f2ffe0c57db918d36d7143c28fd2f1134bd692a9d4dfb4e2f852bbb5
[analyst@secOps lab.support.files]$
```

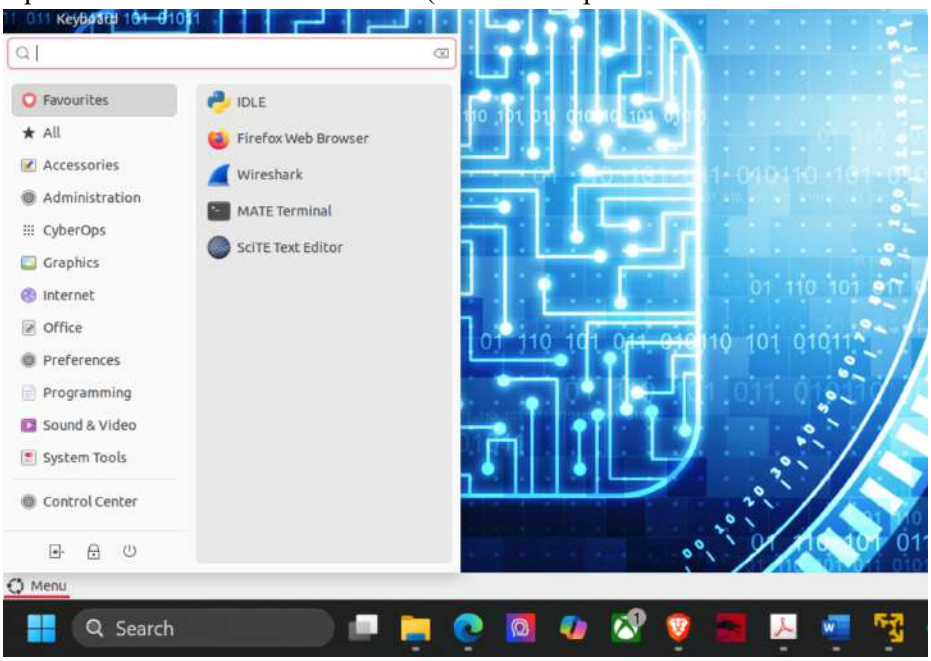
Practical No : 2

Aim: Examining telnet and SSH in wireshark

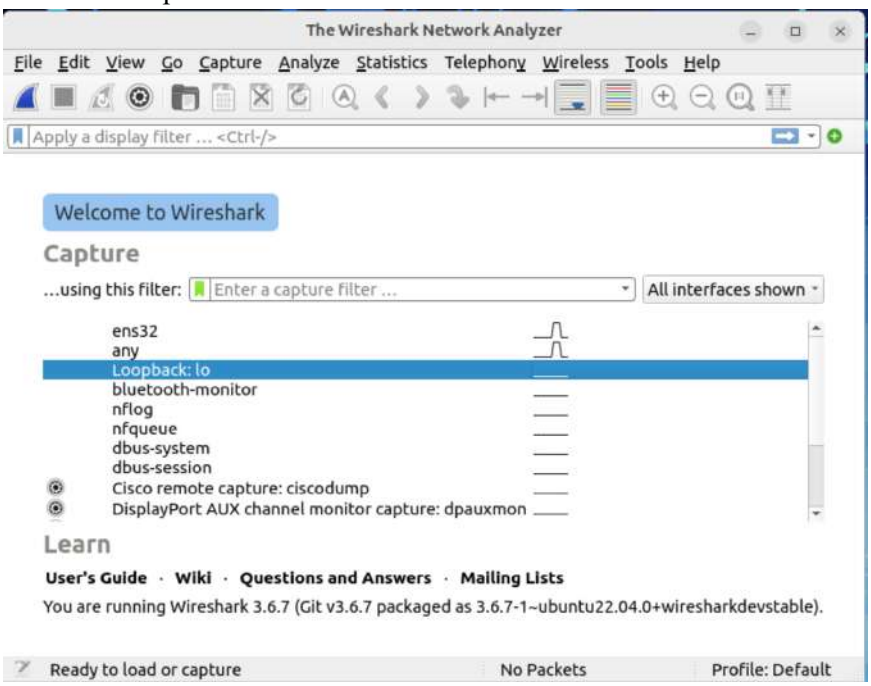
Steps:

Telnet :

1. Open wireshark in virtual machine. (Note: First open wireshark then start telnet connection)



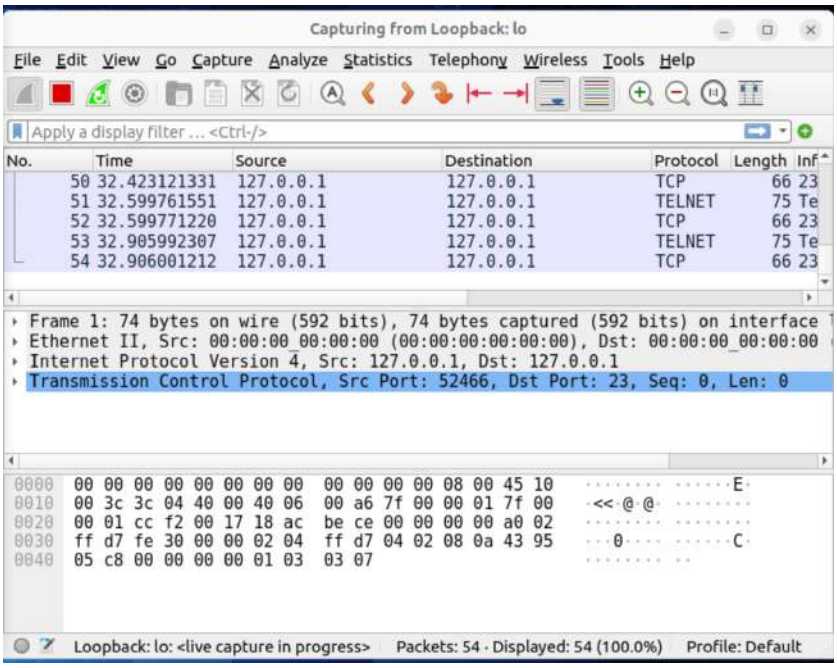
2. Select the loopback:lo interface.



3. Open the terminal and run:
   $telnet localhost

```
[analyst@secOps ~]$ telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Ubuntu 22.04.1 LTS
labvm login:
```

connects to the local machine via telnet, successfully reaching the login prompt for Ubuntu 22.04.1 LTS on localhost.



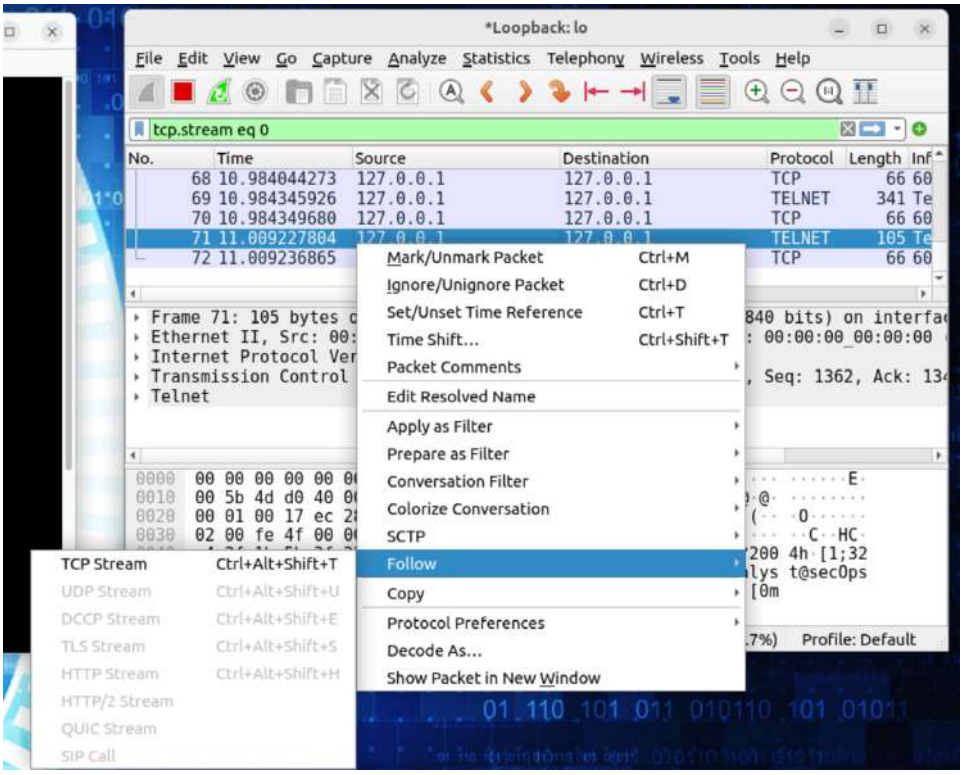
Telnet traffic being recorded on the loopback interface, displaying TCP packets between source port 52466 and destination port 23 (the Telnet port) on 127.0.0.1.

4. Enter the username: analyst and password: cyberops when prompted.

```
[analyst@secOps ~]$ telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Ubuntu 22.04.1 LTS
labvm login: analyst
Password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-60-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Oct 13 07:35:26 PM UTC 2025

  System load:  0.0615234375       Processes:              238
  Usage of /:   34.6% of 22.90GB   Users logged in:        1
  Memory usage: 31%                IPv4 address for ens32: 192.168.153.134
  Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.
```

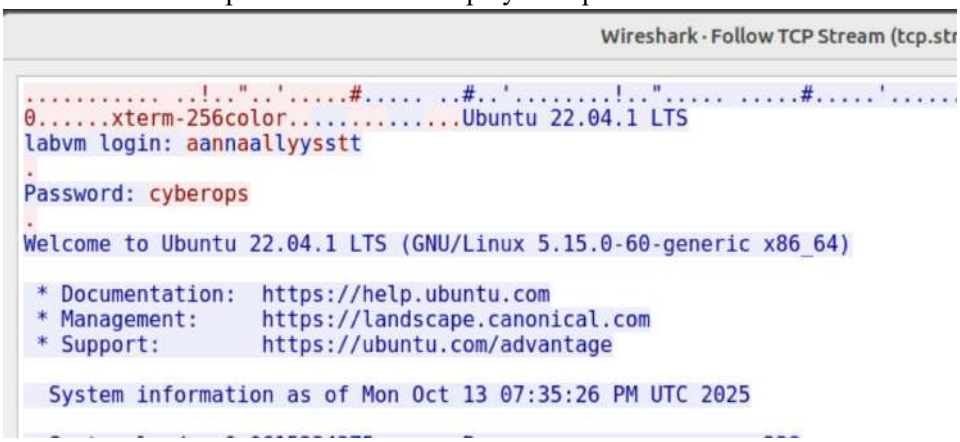
logs into Ubuntu 22.04.1 LTS on the localhost via a Telnet session, providing credentials and viewing system information and support details after a successful login.

5. In the wireshark window, you will see the packets sent and received. Select any packet, right-click and choose Follow -> TCP Stream.



The user right-clicks a Telnet packet in Wireshark and selects "Follow" to analyze the entire TCP stream of the Telnet session between local endpoints.

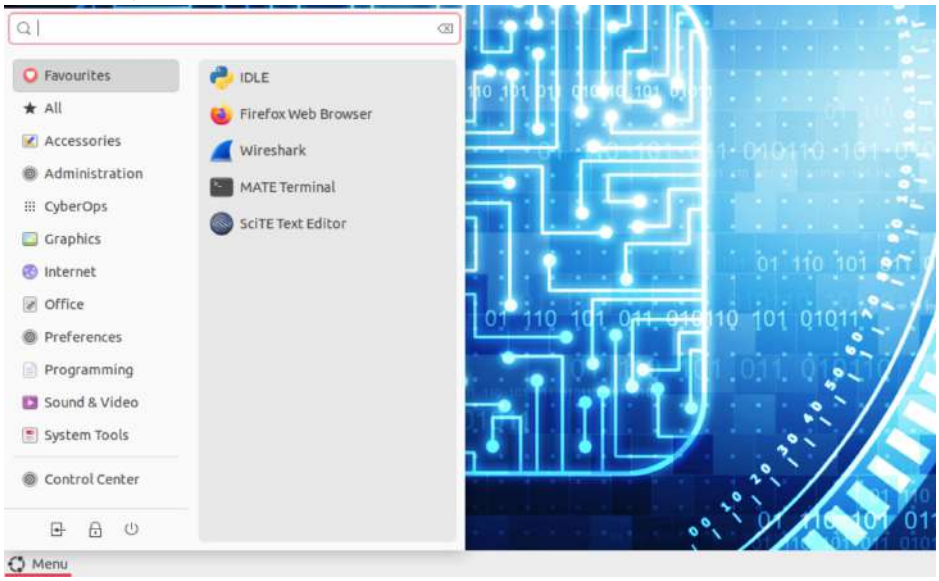
6. The username and password will be displayed in plain text.



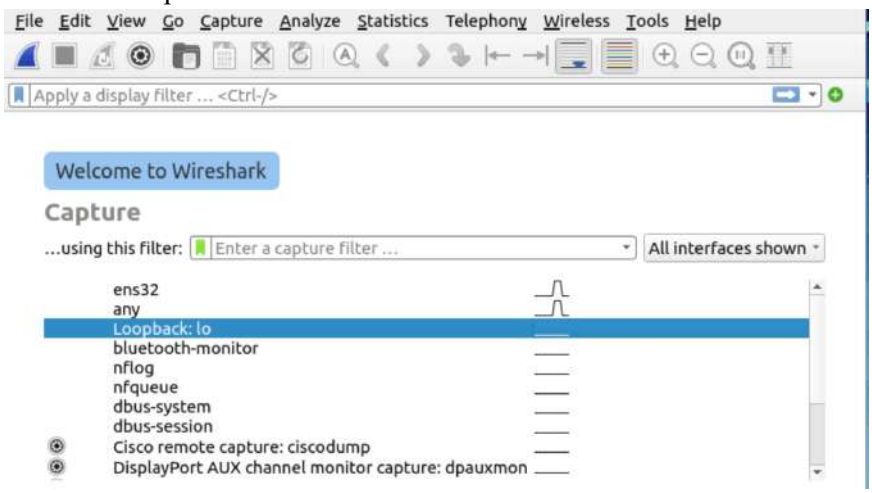
This Wireshark "Follow TCP Stream" window reveals that the Telnet login credentials (username "aannaallyysstt" and password "cyberops") are visible in plaintext during transmission, demonstrating the insecurity of Telnet for sensitive information.

SSH:

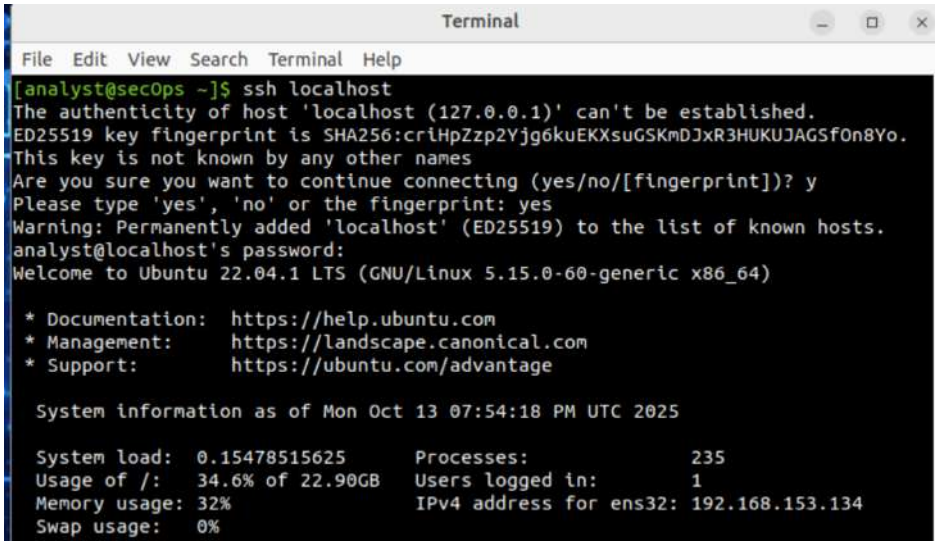
1. Double-click and open wireshark from virtual machine's desktop. (Note: First open wireshark then start ssh connection)



2. Select the loopback:lo interface.



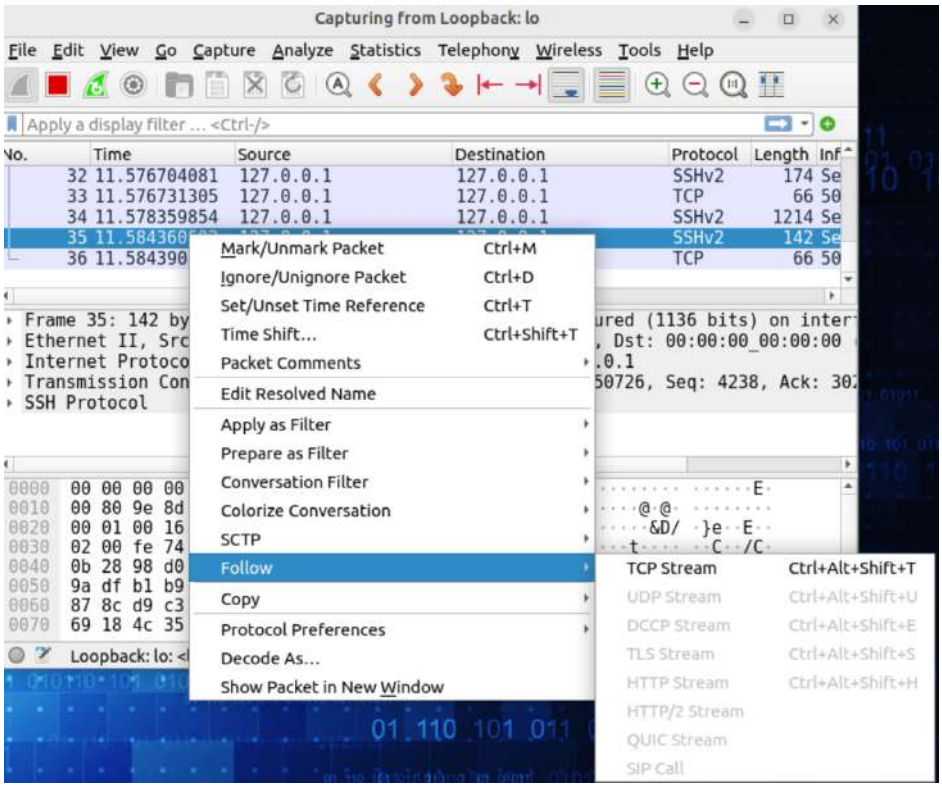
3. Open the terminal and run:
   $ssh localhost



connects to localhost using SSH, verifies the host's fingerprint, enters a password, and securely logs into the Ubuntu 22.04.1 LTS system.

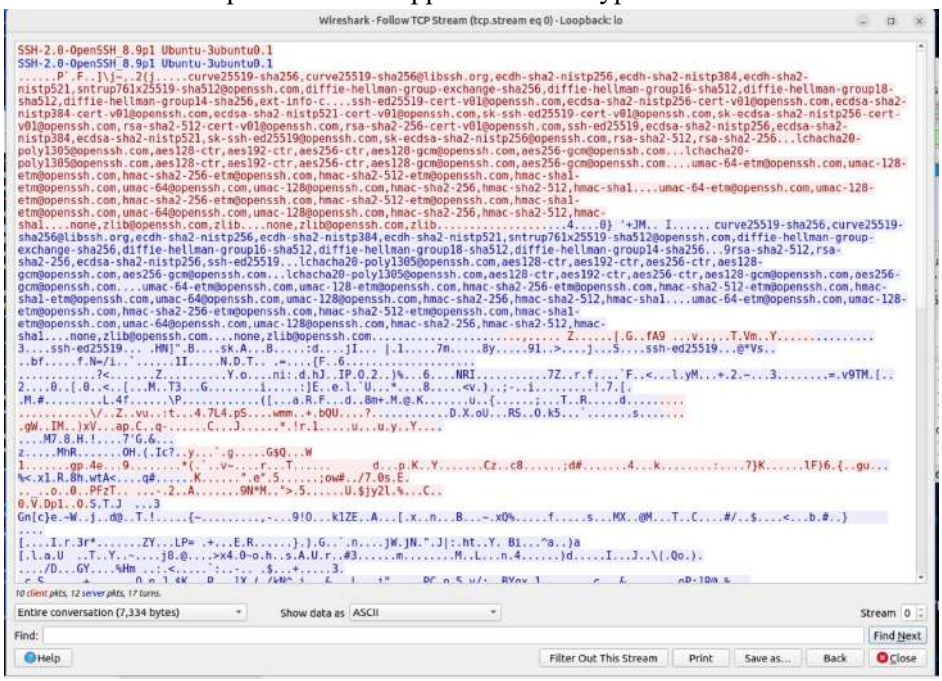
4. Enter the password: cyberops when prompted.

5. In the wireshark window, you will see the packets sent and received. Select any packet, right-click and choose Follow -> TCP Stream.



The user right-clicks on an SSHv2 packet in Wireshark and selects "Follow" to analyze the entire encrypted TCP stream of the SSH session on the loopback interface.

6. The username and password will appear in an encrypted format.



Wireshark shows that SSH session traffic is fully encrypted, so login credentials and sensitive user data remain hidden and cannot be viewed in plaintext.

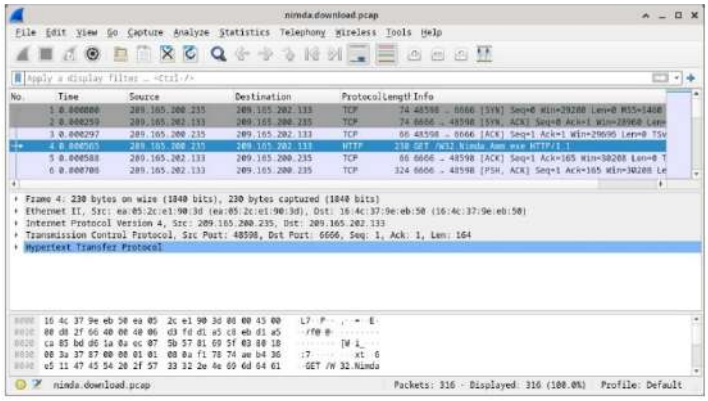
Practical No 3

B. Demonstrate Extract an Executable from a PCAP

Steps:

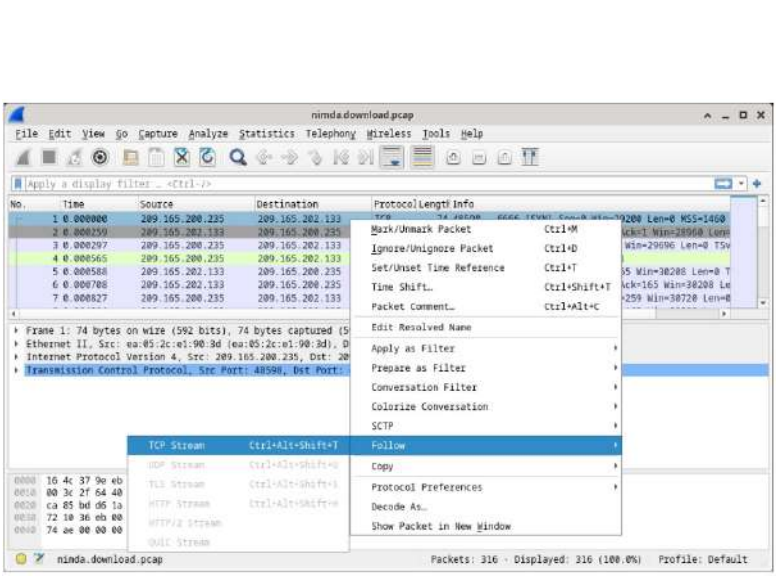
Part 1: Analyze Pre-Captured Logs and Traffic Captures

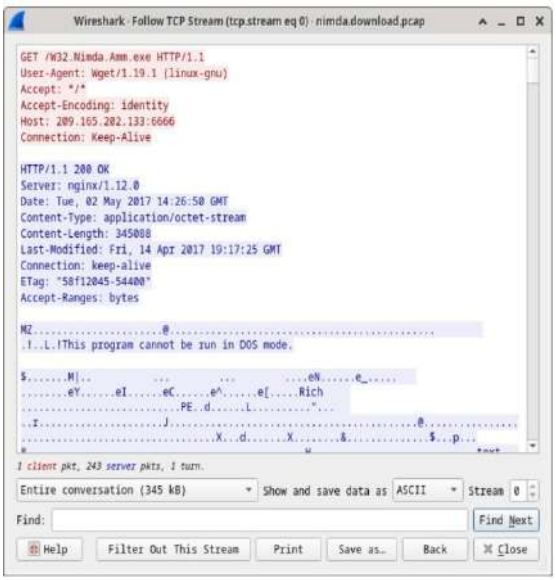
- Open the pcap file in Wireshark.
- Locate the relevant TCP stream transferring the executable, often identified by HTTP GET or FTP data.
- Right-click the packet and select "Follow TCP Stream" to view the file transfer content.
- In the TCP Stream window, set "Show data as" to "Raw" and click "Save as..." to export the raw data as a file.
- Alternatively, go to "File" > "Export Objects" > select the protocol (HTTP/FTP) and save the executable from the displayed list.



Packets one through three are the TCP handshake. The fourth packet shows the request for the malware file. Confirming what was already known, the request was done over HTTP, sent as a GET request.

Because HTTP runs over TCP, it is possible to use Wireshark's Follow TCP Stream feature to rebuild the TCP transaction. Select the first TCP packet in the capture, a SYN packet. Right-click it and Follow TCP stream.



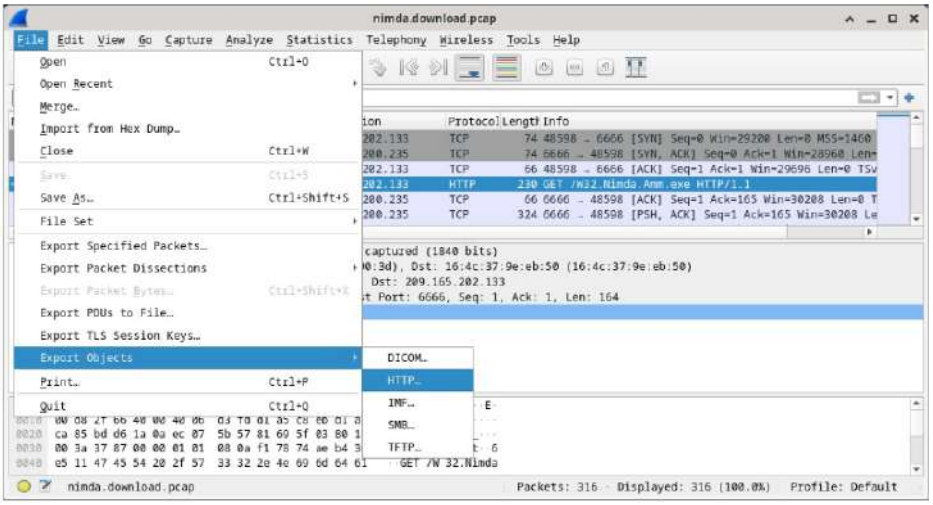


Wireshark displays another window containing the details for the entire selected TCP flow.

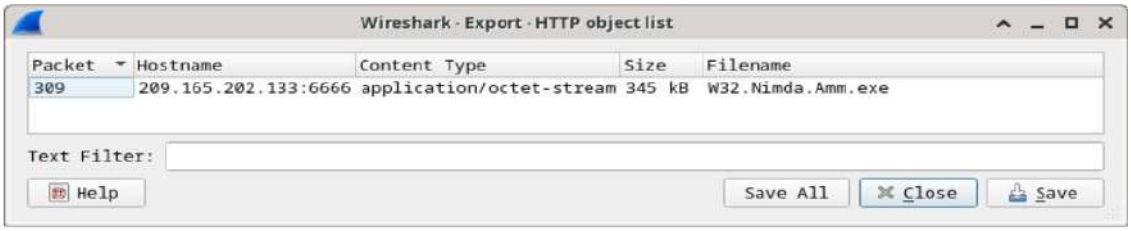
Part 2: Extract Downloaded Files from PCAP

In that fourth packet in the download.pcap file, notice that the HTTP GET request was generated from 209.165.200.235 to 209.165.202.133. The Info column also shows this is in fact the GET request for the file.

With the GET request packet selected, navigate to File > Export Objects > HTTP, from Wireshark's menu.



Wireshark will display all HTTP objects present in the TCP flow that contains the GET request. In this case, only the Nimda.Amm.exe file is present in the capture. It will take a few seconds before the file is displayed.



In the HTTP object list window, select the Nimda.Amm.exe file and click Save As at the bottom of the screen.

Click the left arrow until you see the Home Click Home and then click the analyst folder (not the analyst tab). Save the file there.

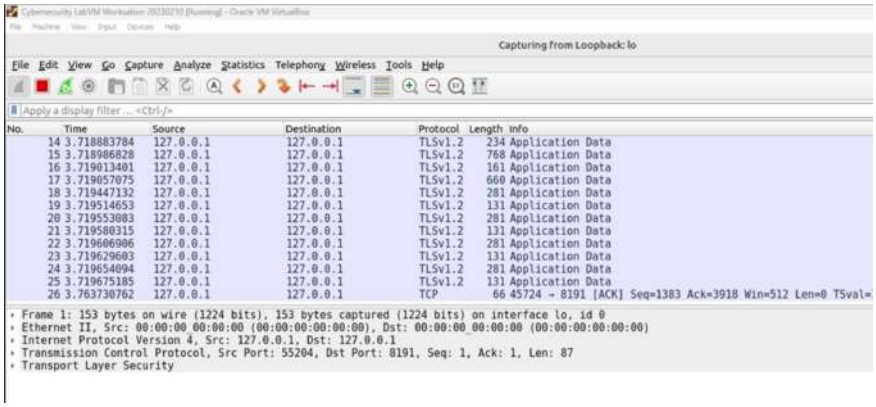
Return to your terminal window and ensure the file was saved. Change directory to the /home/analyst folder and list the files in the folder using the ls -l: cd /home/analyst

The file command gives information on the file type. Use the file command to learn a little more about the malware, as show below: file W32.Nimda.Amm.exe

C. Demonstrate a practical for Exploring DNS Traffic

Part 1: Capture DNS traffic

Start Wireshark. Select an active interface with traffic for packet capture.

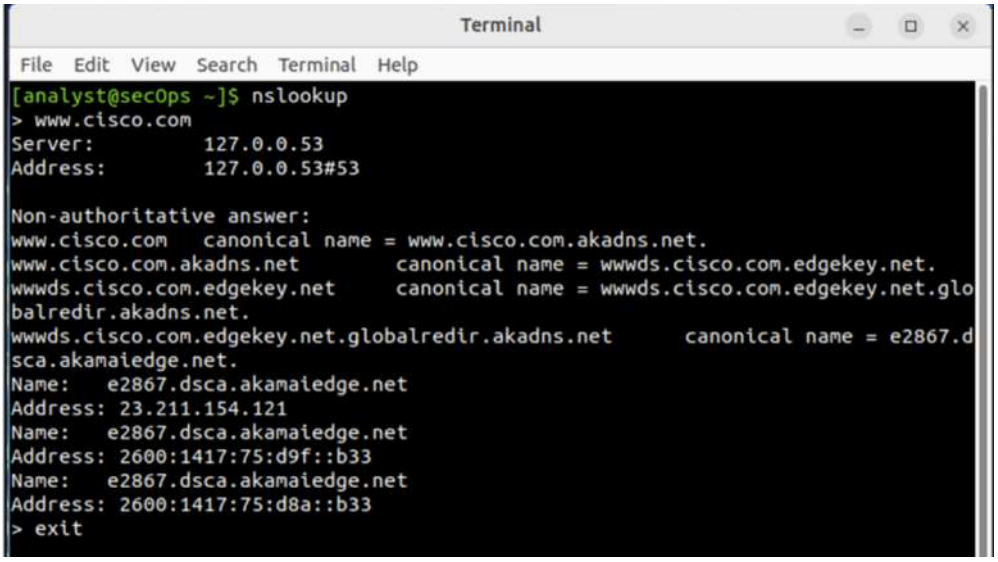


At a command prompt or terminal, type nslookup enter the interactive mode.

Enter the domain name of a website. The domain name www.cisco.com is used in this example.

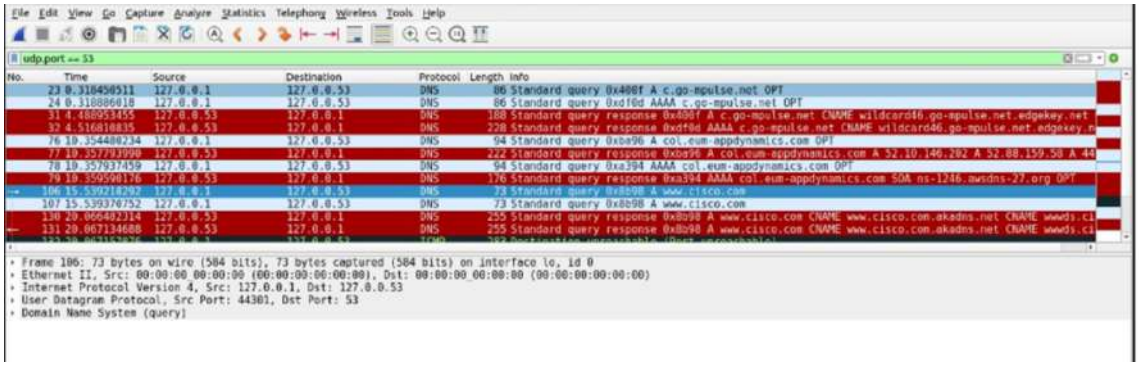
Type exit when finished. Close the command prompt.

Click Stop capturing packets to stop the Wireshark capture.



Part 2: Explore DNS Query Traffic

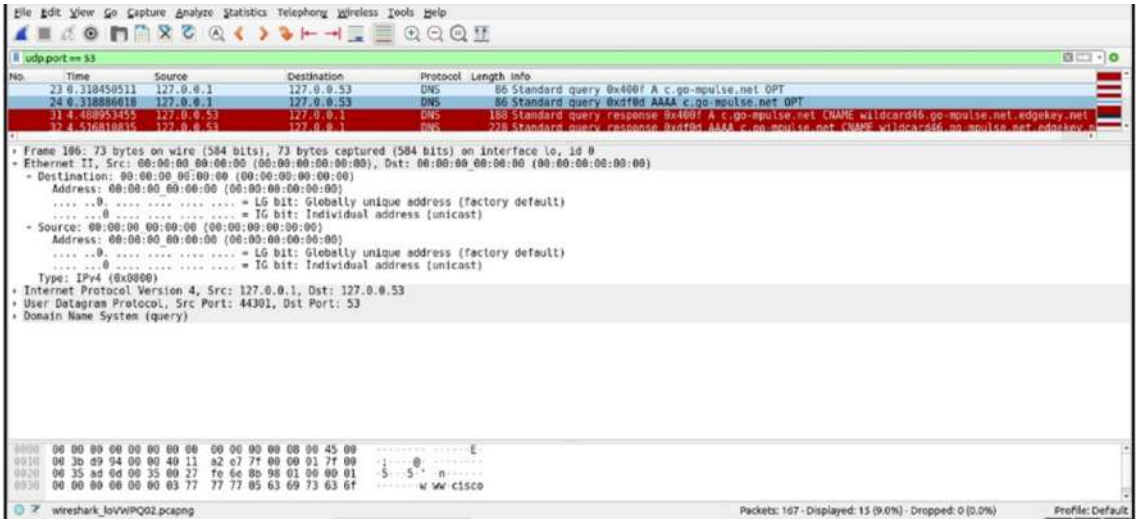
Observe the traffic captured in the Wireshark Packet List pane. Enter udp.port == 53 in the filter box and click the arrow (or press enter) to display only DNS packets.



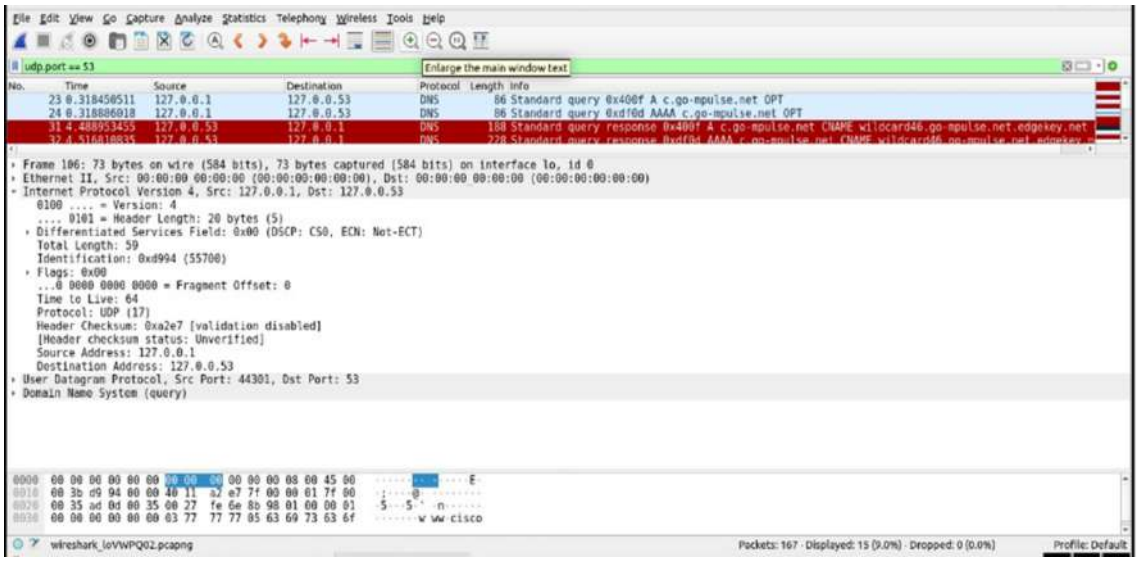
Select the DNS packet contains Standard query and A www.cisco.com in the Info column.

In the Packet Details pane, notice this packet has Ethernet II, Internet Protocol Version 4, User Datagram Protocol and Domain Name System (query).

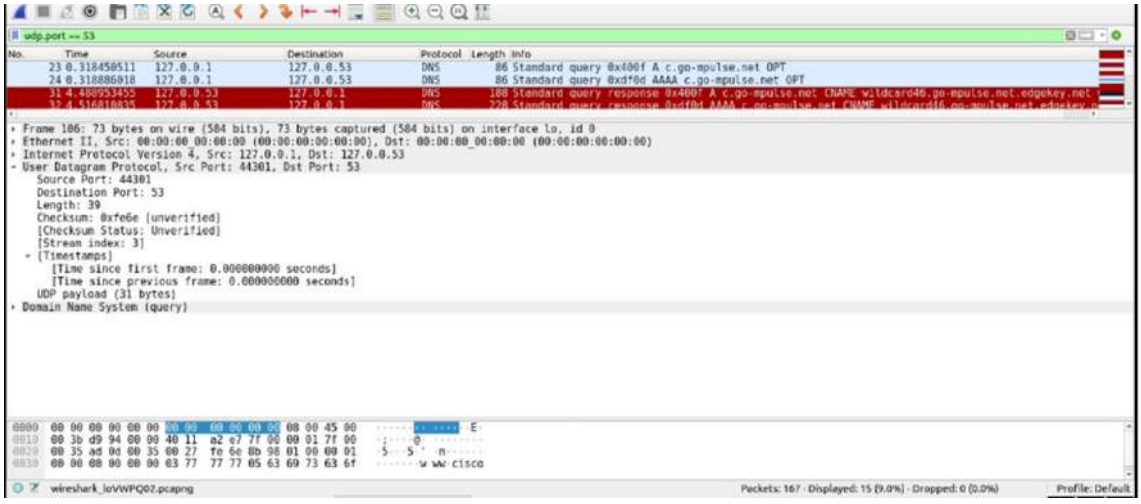
Expand Ethernet II to view the details. Observe the source and destination fields.



Expand Internet Protocol Version 4. Observe the source and destination IPv4 addresses.

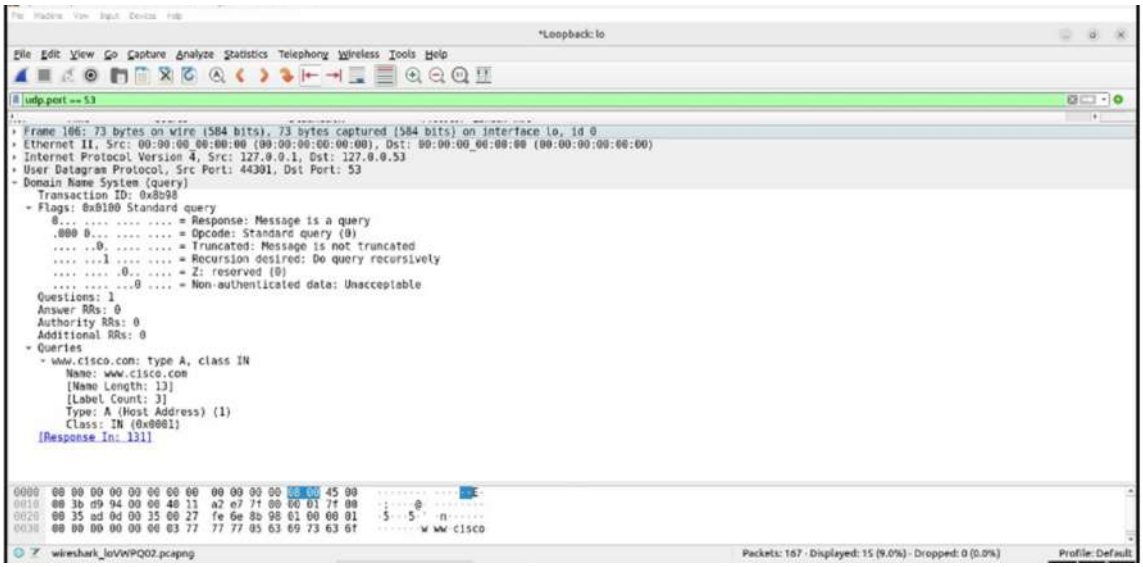


Expand the User Datagram Protocol. Observe the source and destination ports.



Expand Domain Name System (query) in the Packet Details pane. Then expand the Flags and Queries.

Observe the results. The flag is set to do the query recursively to query for the IP address to www.cisco.com.



Practical No 4

A. Using Wireshark to Examine HTTP and HTTPS Traffic

Part 1: Capture and View HTTP Traffic

Step 1: Open a terminal and start tcpdump

While in the terminal application, enter the command:
$sudo tcpdump –i enp0s3 –s 0 –w httpdump.pcap

Enter the password cyberops for the user analyst when prompted.

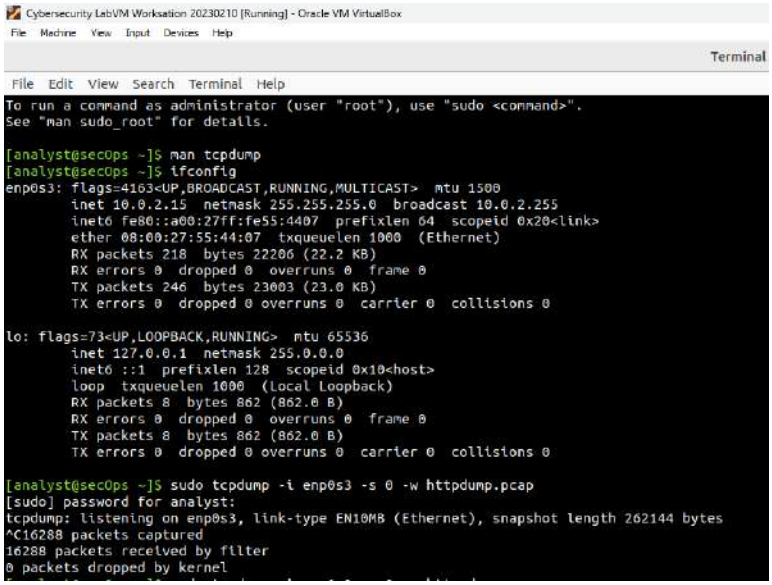
This command starts tcpdump and records network traffic on the enp0s3 interface.

The **-i** command option allows you to specify the interface. If not specified, the tcpdump will capture all traffic on all interfaces.

The **-s** command option specifies the length of the snapshot for each packet. You should limit snaplen to the smallest number that will capture the protocol information in which you are interested. Setting snaplen to 0 sets it to the default of 262144, for backwards compatibility with recent older versions of tcpdump.

The **-w** command option is used to write the result of the tcpdump command to a file. Adding the extension.pcap ensures that operating systems and applications will be able to read to file. All recorded traffic will be printed to the file httpdump.pcap in the home directory of the user analyst.

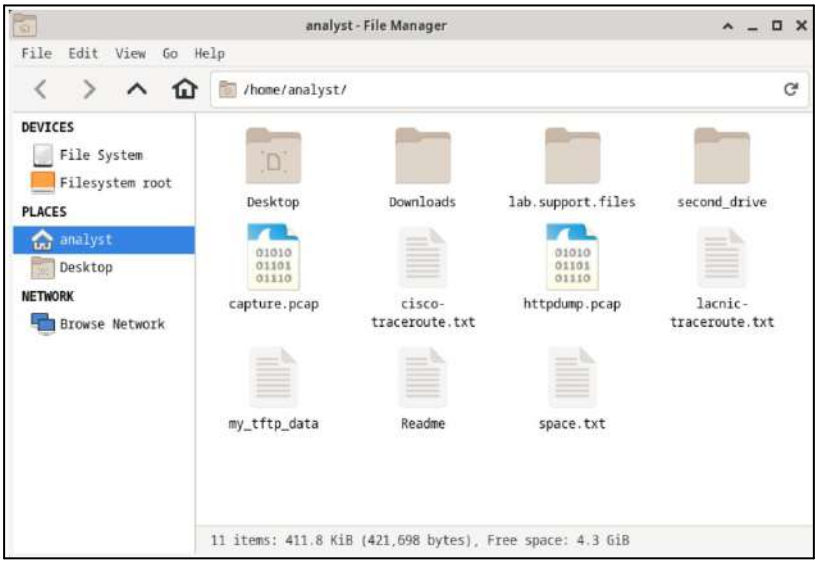
Use the man pages for tcpdump to determine the usage of the -s and -w command options.



- Open a web browser from the launch bar within the CyberOps Workstation VM. Navigate to http://www.altoromutual.com/login.jsp
- Because this website uses HTTP, the traffic is not encrypted. Click the Password field to see the warning pop up.
- Enter a username of Admin with a password of Admin and click Login.
  Close the web browser.
- Return to the terminal window where tcpdump is running. Enter CTRL+C to stop the packet capture.

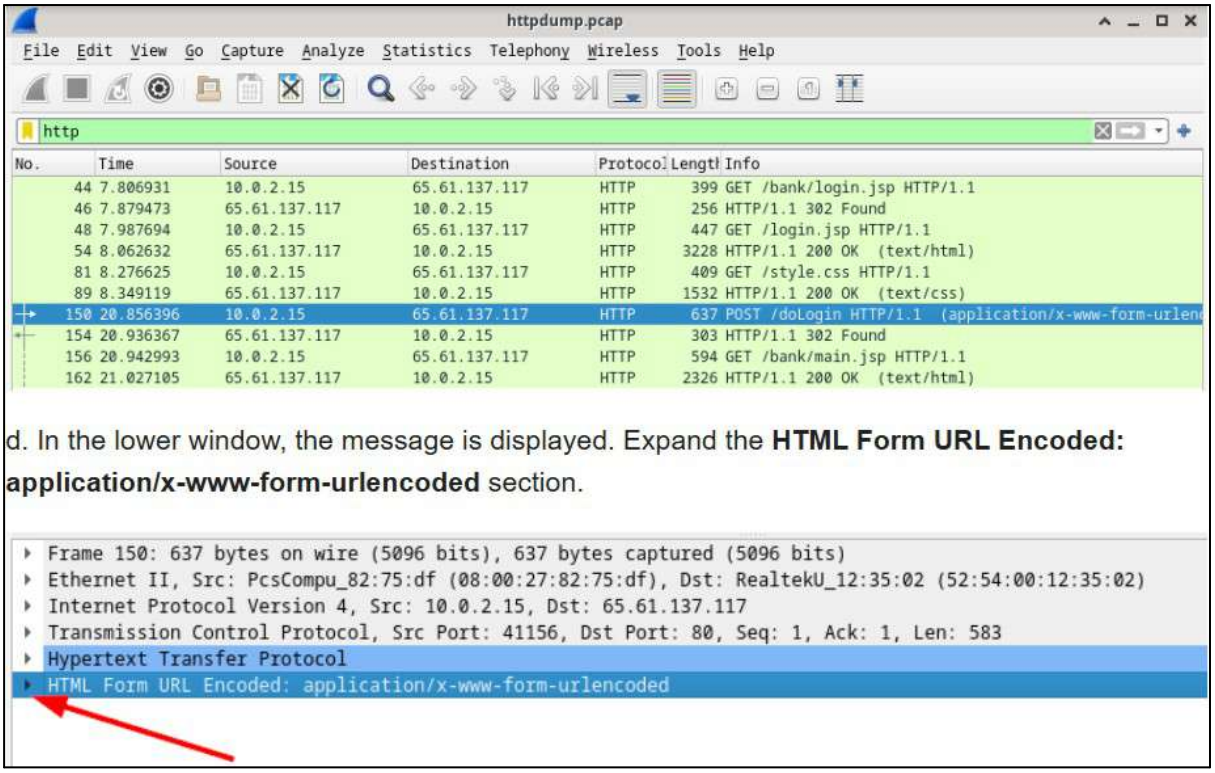
Step 2: View the HTTP Traffic

Click the File Manager icon on the desktop and browse to the home folder for the user analyst. Double-click the httpdump.pcap file, in the Open With dialog box scroll down to Wireshark and then click Open.



In the Wireshark application, filter for http and click Apply.

Browse through the different HTTP messages and select the POST message.



d. In the lower window, the message is displayed. Expand the **HTML Form URL Encoded: application/x-www-form-urlencoded** section.

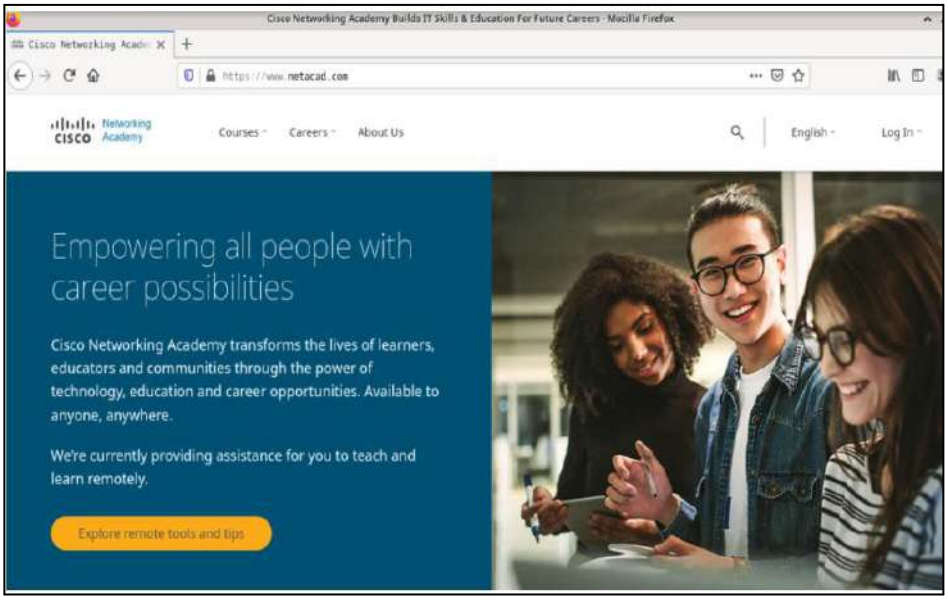
Part 2: Capture and View HTTPS Traffic

Step 1: Start tcpdump within a terminal

- While in the terminal application, enter the command:
  $sudo tcpdump –i enp0s3 –s 0 –w httpsdump.pcap
- Enter the password cyberops for the user analyst when prompted.

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap
[sudo] password for analyst:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C28892 packets captured
28892 packets received by filter
0 packets dropped by kernel
```

- This command will start tcpdump and record network traffic on the enp0s3 interface of the Linux workstation. If your interface is different than enp0s3, please modify it when using the above command.
- All recorded traffic will be printed to the file httpsdump.pcap in the home directory of the user analyst.
- Open a web browser from the launch bar within the CyberOps Workstation VM. Navigate to www.netacad.com.
- Click Log in.

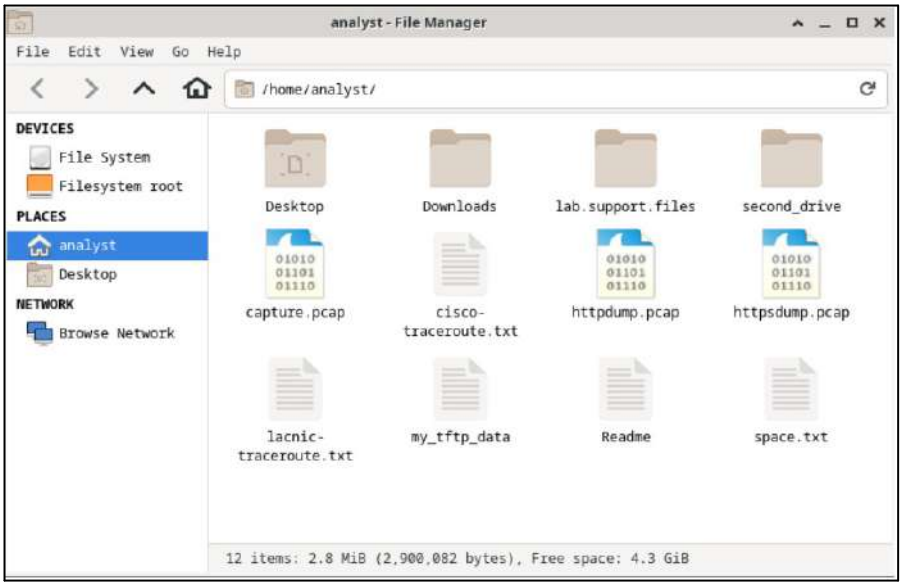


- Enter in your NetAcad username and password. Click Next.

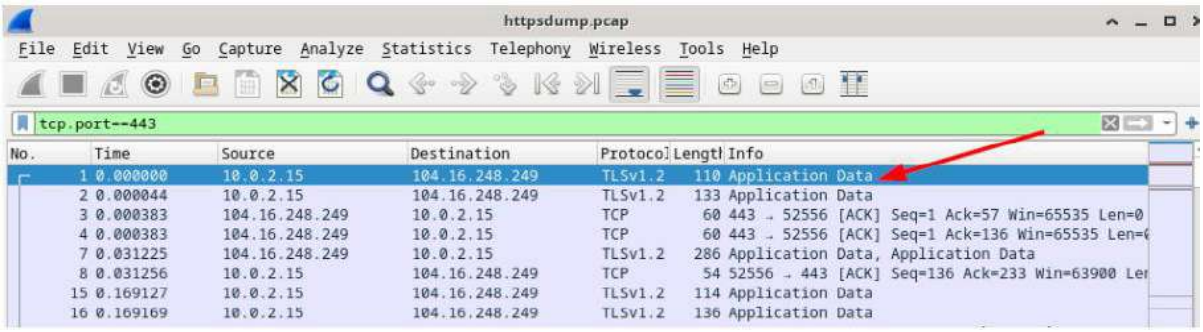
- Close the web browser in the VM.
- Return to the terminal window where tcpdump is running. Enter CTRL+C to stop the packet capture.

Step 2: View the HTTPS capture

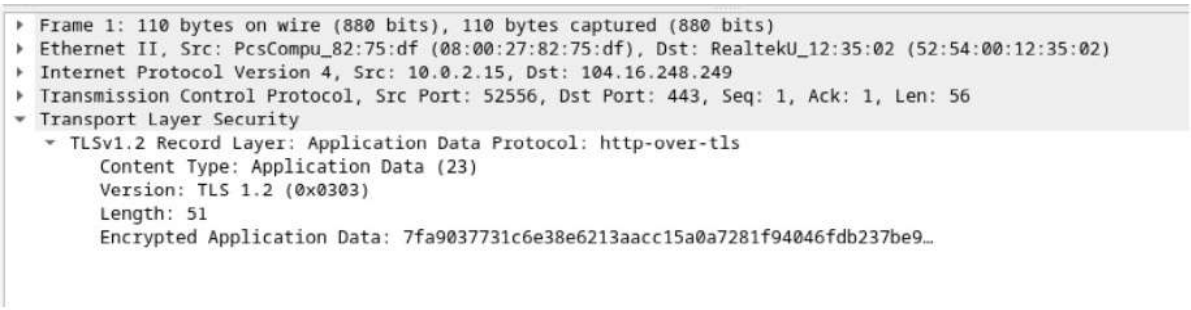
- Click the Filesystem icon on the desktop and browse to the home folder for the user analyst. Open the httpsdump.pcap file.



- In the Wireshark application, expand the capture window vertically and then filter by HTTPS traffic via port 443.
- Enter tcp.port==443 as a filter, and click Apply.
- Browse through the different HTTPS messages and select an Application Data message.



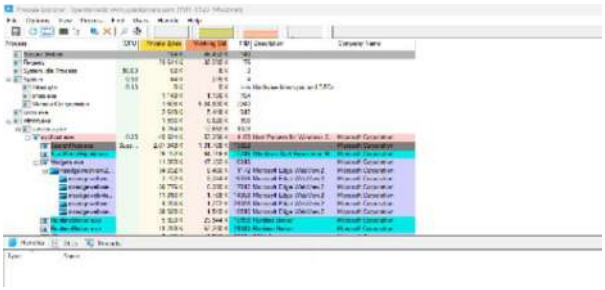
- Completely expand the Secure Sockets Layer section.
- Click the Encrypted Application Data.



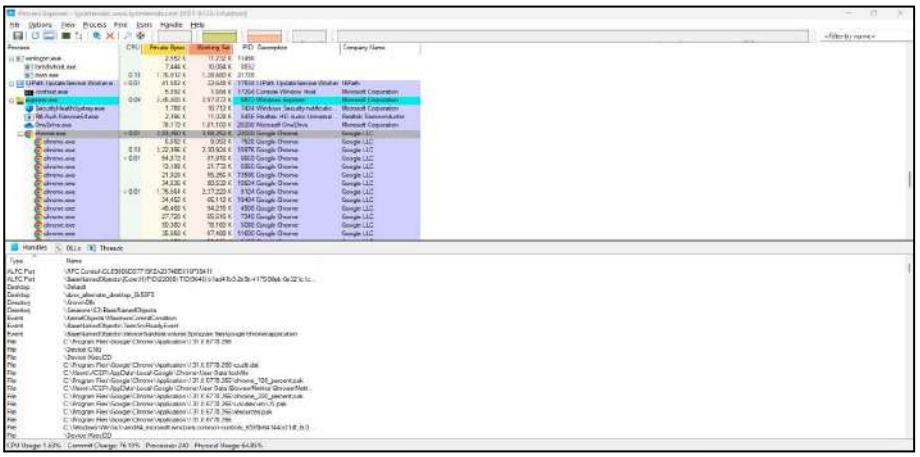
B. Exploring Processes, Threads, Handles, and Windows Registry

Part 1: Exploring Processes

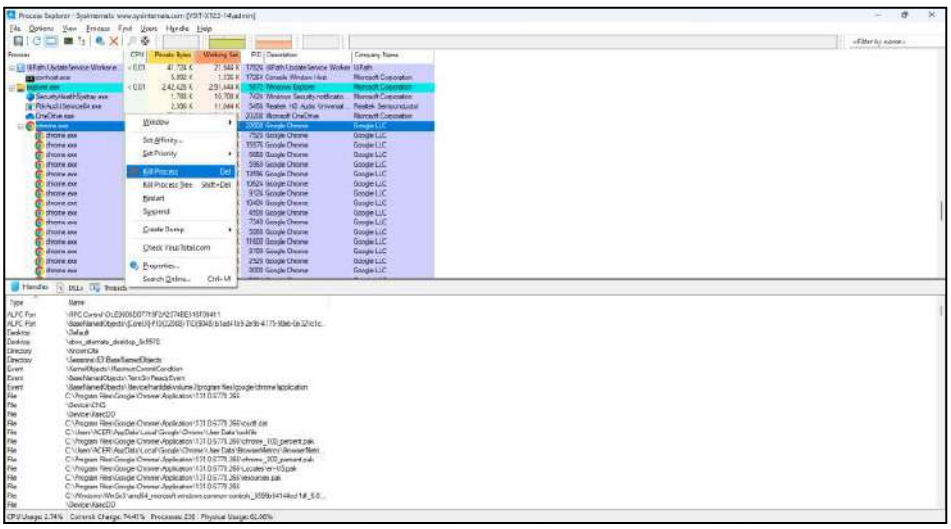
- Navigate to the SysinternalsSuite folder with all the extracted files.
- Open procexp.exe. Accept the Process Explorer License Agreement when prompted.
- The Process Explorer displays a list of currently active processes.



- To locate the web browser process, drag the Find Window's Process icon into the opened web browser window. Google Chrome was used in this example.



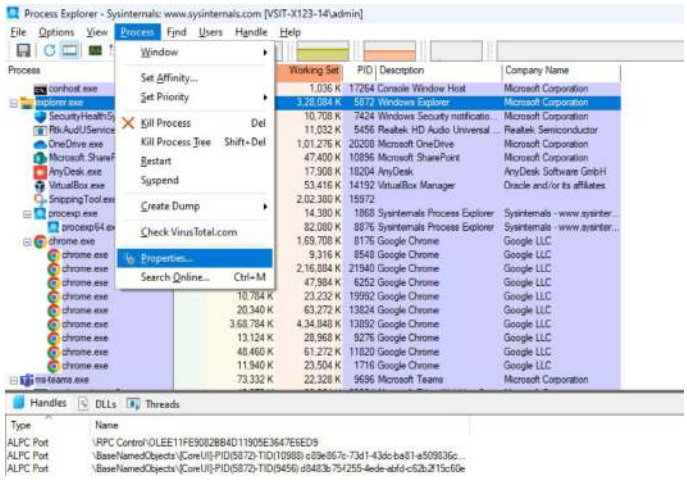
- The Google Chrome process can be terminated in the Process Explorer. Right-click the selected process and select Kill Process. Click OK to continue.



Part 2: Exploring Threads and Handles

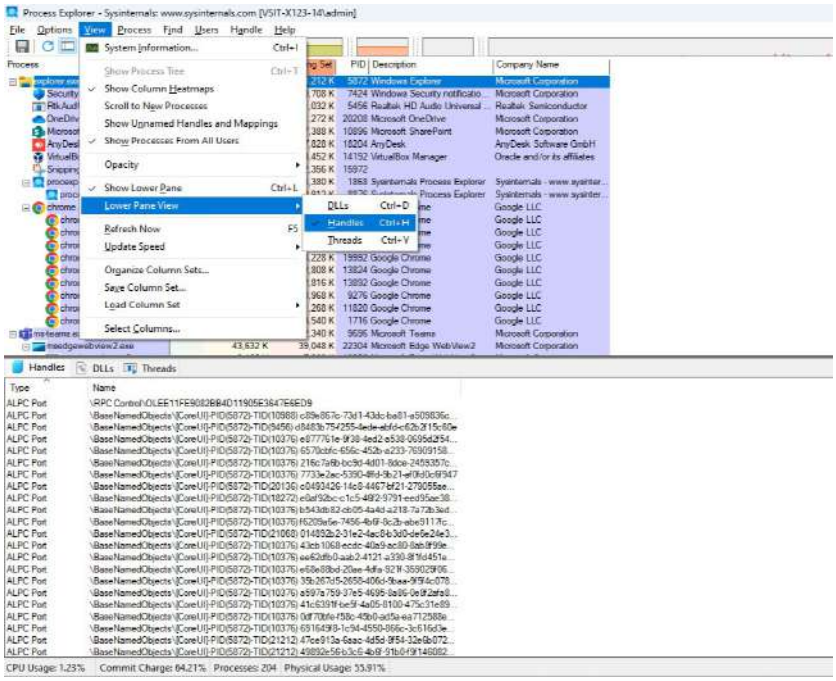
Step 1: Explore Threads

- Open a command prompt.
- In Process Explorer window, right-click and Select Properties…. Click the Threads tab to view the active threads for the conhost.exe process. Click OK to continue if prompted by a warning dialog box.
- Examine the details of the thread.



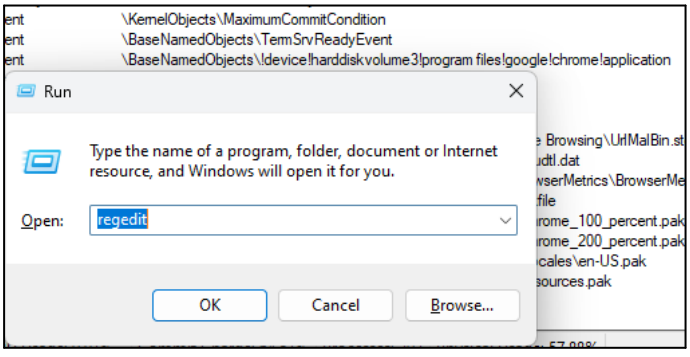
Step 2: Explore Handles

- In the Process Explorer, click View > select Lower Pane View > Handles to view the handles associated with the conhost.exe process.



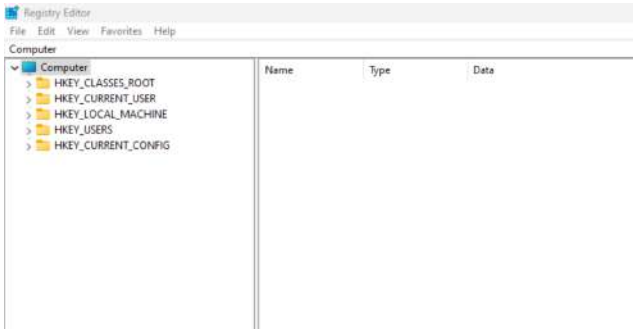
Step 3: Exploring Windows Registry

- The Windows Registry is a hierarchical database that stores most of the operating systems and desktop environment configuration settings.
- To access the Windows Registry, click Start > Search for regedit and select Registry Editor. Click Yes when asked to allow this app to make changes.

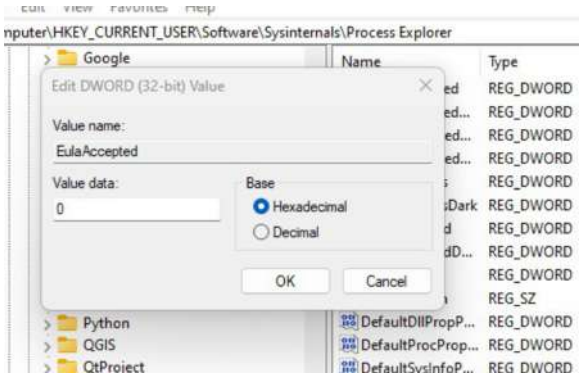


The Registry Editor has five hives. These hives are at the top level of the registry.

- HKEY_LOCAL_MACHINE and HKEY_CLASSES_ROOT hold machine-wide settings, file associations, and application data.
- HKEY_CURRENT_USER and HKEY_USERS store user-specific settings, with HKEY_CURRENT_USER reflecting the active user.
- HKEY_CURRENT_CONFIG provides current hardware config details used at bootup.



- In a previous step, you had accepted the EULA for Process Explorer. Navigate to the EulaAccepted registry key for Process Explorer.
- Click to select Process Explorer in HKEY_CURRENT_USER > Software > Sysinternals > Process Explorer. Scroll down to locate the key EulaAccepted. Currently, the value for the registry key EulaAccepted is 0x00000001(1).
- Double-click EulaAccepted registry key. Currently the value data is set to 1. The value of 1 indicates that the EULA has been accepted by the user.
- Change the 1 to 0 for Value data. The value of 0 indicates that the EULA was not accepted. Click OK to continue.



- Open the Process Explorer. Navigate to the folder where you have downloaded SysInternals. Open the folder SysInternalsSuite > Open procexp.exe.
- When you open the Process Explorer, what did you see?

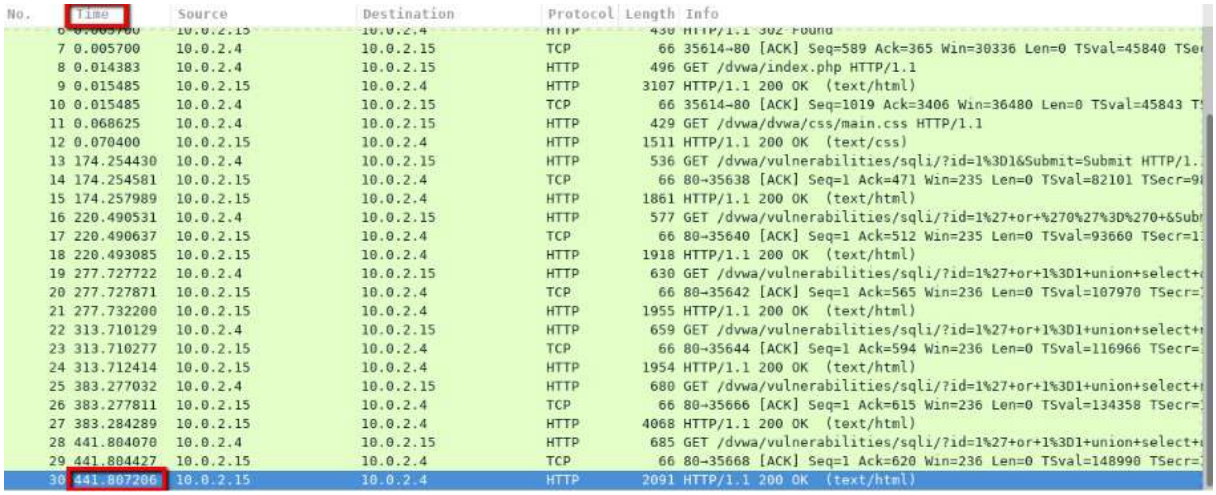
Practical No 5

Aim : Perform a practical to Attack on a MySQL Database by using PCAP file.

Steps :

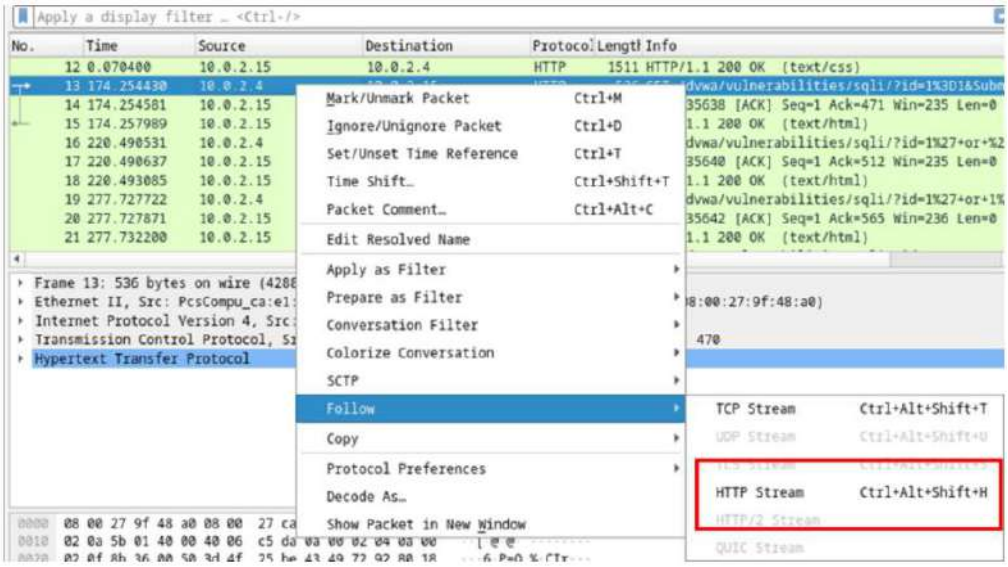
Part 1: Open Wireshark and load the PCAP file.

- Start the CyberOps Workstation VM.
- Click Applications > CyberOPS > Wireshark on the desktop and browse to the Wireshark application.
- In the Wireshark application, click Open in the middle of the application under Files.
- Browse through the /home/analyst/ directory and search for lab.support.files. In the lab.support.files directory and open the SQL_Lab.pcap file.
- The PCAP file opens within Wireshark and displays the captured network traffic. This capture file extends over an 8-minute (441 second) period, the duration of this SQL injection attack.

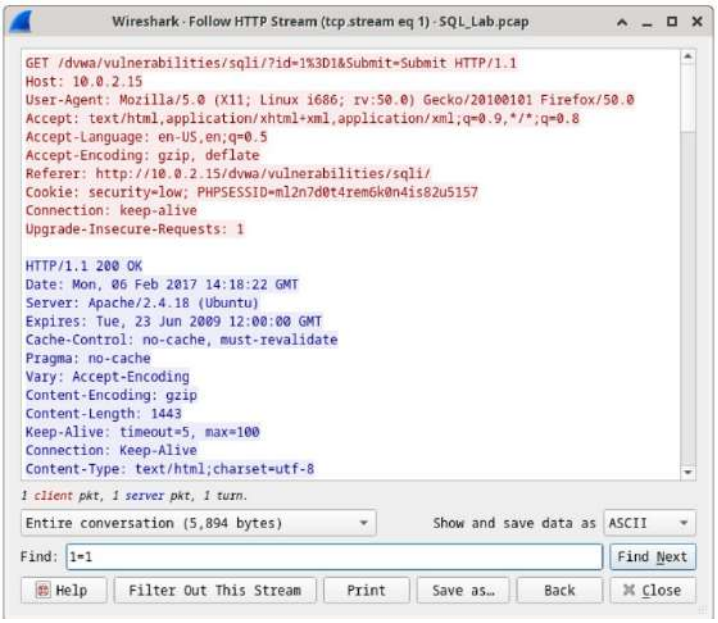


Part 2: View the SQL Injection Attack.

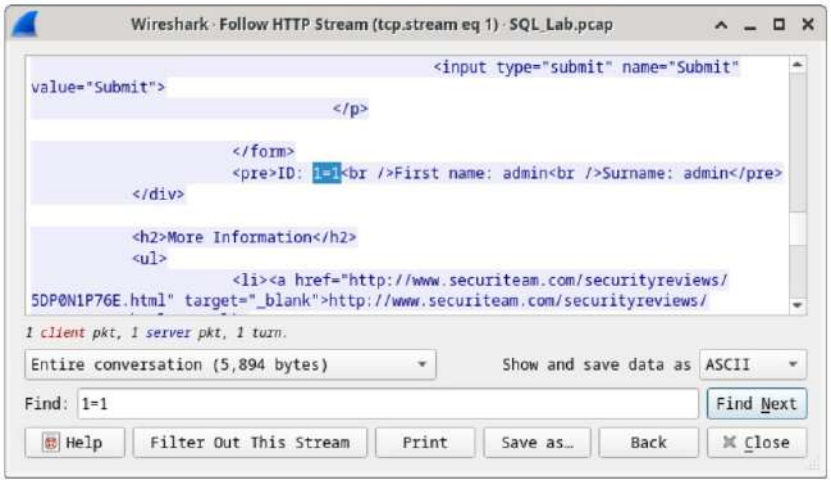
- Within the Wireshark capture, right-click line 13 and select Follow > HTTP Stream. Line 13 was chosen because it is a GET HTTP request. This will be very helpful in following the data stream as the application layers sees it and leads up to the query testing for the SQL injection.



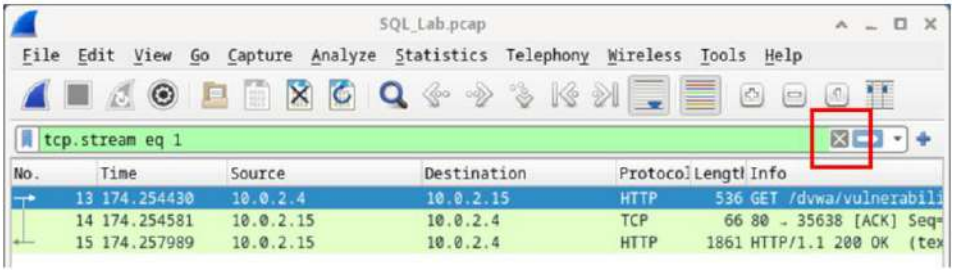
- The source traffic is shown in red. The source has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.
- In the Find field, enter 1=1. Click Find Next.



- The attacker has entered a query (1=1) into a UserID search box on the target 10.0.2.15 to see if the application is vulnerable to SQL injection. Instead of the application responding with a login failure message, it responded with a record from a database. The attacker has verified they can input an SQL command, and the database will respond. The search string 1=1 creates an SQL statement that will be always true. In the example, it does not matter what is entered into the field, it will always be true.



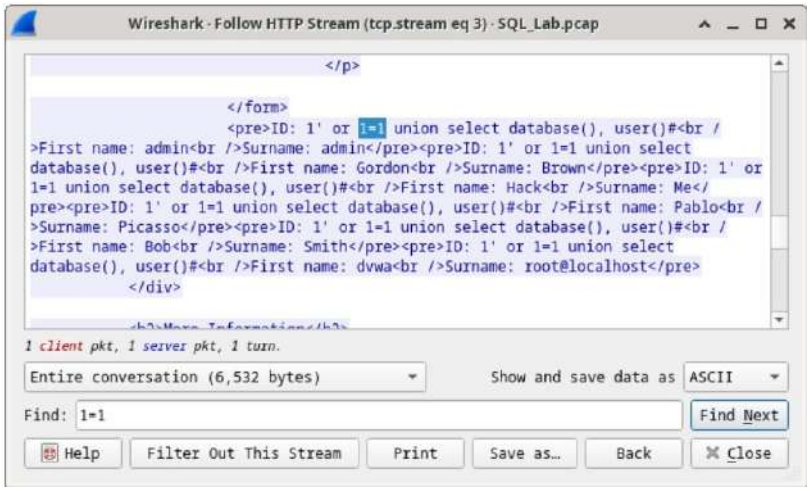
- Close the Follow HTTP Stream window.
- Click Clear display filter to display the entire Wireshark conversation.



Part 3: The SQL Injection Attack continues

- Within the Wireshark capture, right-click line 19, and click Follow > HTTP Stream.
- In the Find field, enter 1=1. Click Find Next.

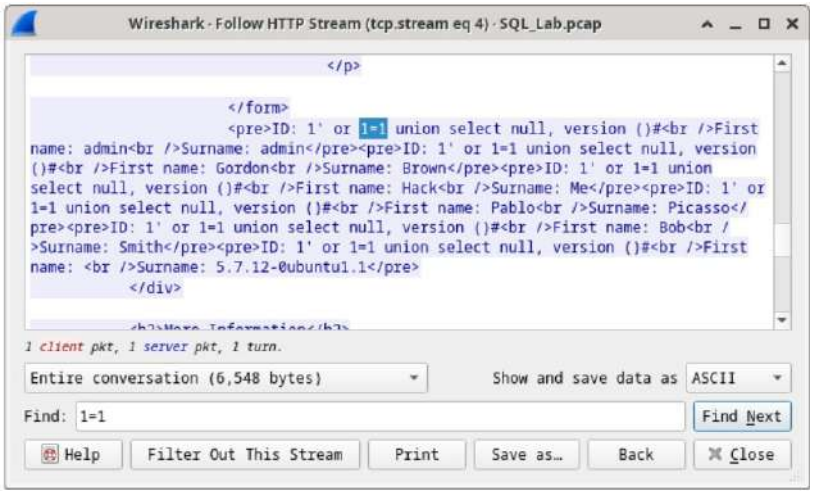
- The attacker has entered a query (1' or 1=1 union select database(), user()#) into a UserID search box on the target 10.0.2.15. Instead of the application responding with a login failure message, it responded with the following information:



- The database name is dvwa and the database user is root@localhost. There are also multiple user accounts being displayed.
- Close the Follow HTTP Stream window.
- Click Clear display filter to display the entire Wireshark conversation

Part 4: The SQL Injection Attack provides system information.

- Within the Wireshark capture, right-click line 22 and select Follow > HTTP Stream. In red, the source traffic is shown and is sending the GET request to host 10.0.2.15. In blue,
- Click Find Next.
- The attacker has entered a query (1' or 1=1 union select null, version ()#) into a UserID search box on the target 10.0.2.15 to locate the version identifier. Notice how the version identifier is at the end of the output right before the </pre>. closing HTML code.

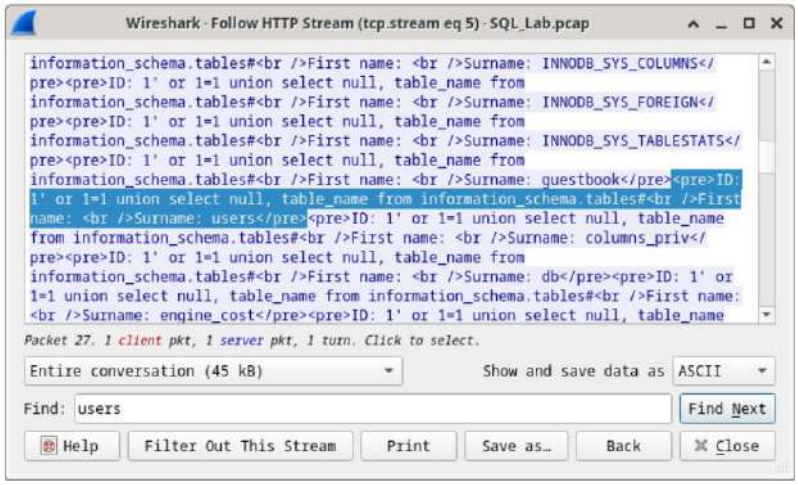


- Close the Follow HTTP Stream window.
- Click Clear display filter to display the entire Wireshark conversation.

Part 5: The SQL Injection Attack and Table Information

- Within the Wireshark capture, right-click on line 25 and select Follow > HTTP Stream.
  The source is shown in red. It has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.
- In the Find field, enter users. Click Find Next.

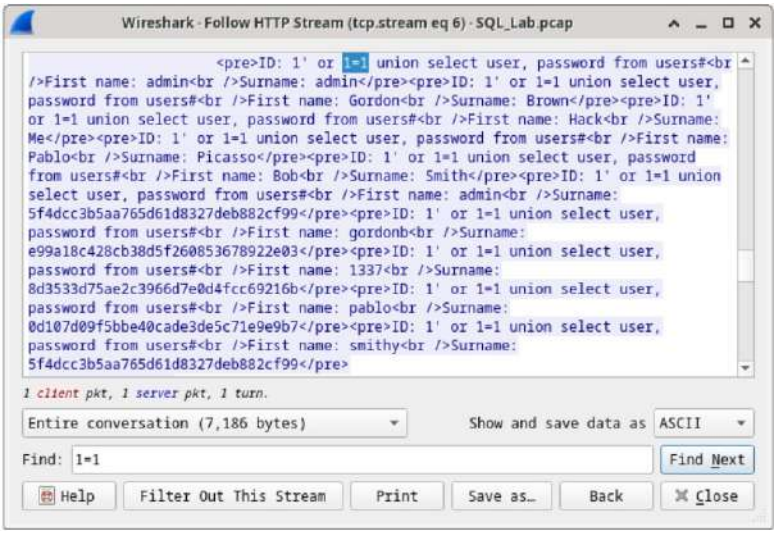
- The attacker has entered a query (1'or 1=1 union select null, table_name from information_schema.tables#) into a UserID search box on the target 10.0.2.15 to view all the tables in the database. This provides a huge output of many tables, as the attacker specified "null" without any further specifications.



- Close the Follow HTTP Stream window.
- Click Clear display filter to display the entire Wireshark conversation.

Part 6: The SQL Injection Attack Concludes

- Within the Wireshark capture, right-click line 28 and select Follow > HTTP Stream. The source is shown in red. It has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.
- Click Find and type in 1=1. Search for this entry. When the text is located, click Cancel in the Find text search box.
- The attacker has entered a query (1'or 1=1 union select user, password from users#) into a UserID search box on the target 10.0.2.15 to pull usernames and password hashes!



Practical No 6

Aim: Create your own syslog Server

Steps :

Step 1: To check whether rsyslog services already running or not use above
$sudo systemctl status rsyslog

```
[analyst@secOps ~]$ sudo systemctl status rsyslog
[sudo] password for analyst:
● rsyslog.service - System Logging Service
     Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor prese>
     Active: active (running) since Mon 2025-10-13 18:27:19 UTC; 1h 54min ago
TriggeredBy: ● syslog.socket
       Docs: man:rsyslogd(8)
             man:rsyslog.conf(5)
             https://www.rsyslog.com/doc/
   Main PID: 551 (rsyslogd)
      Tasks: 4 (limit: 2287)
     Memory: 3.3M
        CPU: 57ms
     CGroup: /system.slice/rsyslog.service
             └─551 /usr/sbin/rsyslogd -n -iNONE

Oct 13 18:27:19 labvm systemd[1]: Starting System Logging Service...
Oct 13 18:27:19 labvm rsyslogd[551]: imuxsock: Acquired UNIX socket '/run/syste>
Oct 13 18:27:19 labvm rsyslogd[551]: rsyslogd's groupid changed to 113
Oct 13 18:27:19 labvm rsyslogd[551]: rsyslogd's userid changed to 107
Oct 13 18:27:19 labvm rsyslogd[551]: [origin software="rsyslogd" swVersion="8.2>
Oct 13 18:27:19 labvm systemd[1]: Started System Logging Service.
lines 1-20/20 (END)
```

The rsyslog service is actively running, confirming that the system logging service started successfully and is capturing system logs.

Step 2: In case not installed or running, install rsyslog using the following commands:
$sudo apt-get update
$sudo apt-get install rsyslog

Step 3: Open rsyslog configuration file
$sudo nano /etc/rsyslog.conf

```
[analyst@secOps ~]$ sudo nano /etc/rsyslog.conf
```

Step 4: Uncomment above four lines that enable udp and tcp port binding:

```
module(load="imuxsock") # provides support for local system logging
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")
```

Step 5: Add template right before GLOBAL DIRECTIVES section.
$template remote-incoming-logs,"/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
. ?remote-incoming-logs

```
# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")

$template remote-incoming-logs,"/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
*.* ?remote-incoming-logs
# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")

###########################
#### GLOBAL DIRECTIVES ####
###########################

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
```

This rsyslog configuration snippet enables TCP syslog reception on port 514, sets a template for remote logs, and allows kernel logging support.

Step 6: Save and restart rsyslog service:
$sudo systemctl restart rsyslog

```
[analyst@secOps ~]$ sudo systemctl restart rsyslog
```

Step 7: Confirmed that rsyslog service is listening on configured ports
$ss tunelp | grep 514

```
[analyst@secOps ~]$ ss -tunelp | grep 514
udp   UNCONN 0      0                                  0.0.0.0:514       0.0.0.0:*
   ino:92782 sk:6 cgroup:/system.slice/rsyslog.service <->
udp   UNCONN 0      0                                     [::]:514          [::]:*
   ino:92783 sk:a cgroup:/system.slice/rsyslog.service v6only:1 <->
tcp   LISTEN 0      25                                 0.0.0.0:514       0.0.0.0:*
   ino:92786 sk:e cgroup:/system.slice/rsyslog.service <->
tcp   LISTEN 0      25                                    [::]:514          [::]:*
   ino:92787 sk:11 cgroup:/system.slice/rsyslog.service v6only:1 <->
```

The output confirms that rsyslog is listening for syslog messages on bothTCP and UDP port 514, supporting IPv4 and IPv6 connections.

Step 8: Allow rsyslog firewall port rules
$sudo ufw allow 514/tcp
$sudo ufw allow 514/udp

```
[analyst@secOps ~]$ sudo ufw allow 514/tcp
Rules updated
Rules updated (v6)
[analyst@secOps ~]$ sudo ufw allow 514/udp
Rules updated
Rules updated (v6)
```

Step 9: To verify configuration, run the following command:
$sudo rsyslogd -N1 -f /etc/rsyslog.conf

```
[analyst@secOps ~]$ sudo rsyslogd -N1 -f /etc/rsyslog.conf
rsyslogd: version 8.2112.0, config validation run (level 1), master config /etc/
rsyslog.conf
rsyslogd: End of config validation run. Bye.
```

This command checks the rsyslog configuration file for errors, confirming syntax validity without actually starting the service.

Practical No : 7

Aim: Configure your Linux system to send syslog messages to a syslog server and Read them.

Steps:
Step 1: Install and configure rsyslog server first for that please refer practical no 5.

Step 2: Open your system and install rsyslog using the following commands
$sudo apt-get update
$sudo apt-get install rsyslog

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.2 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.4 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [45.1 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [119 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [176 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [217 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [930 kB]
Fetched 66.0 MB in 5min 49s (189 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
781 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install rsyslog
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libestr0 libfastjson4 liblognorm5
Suggested packages:
  rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-doc rsyslog-openssl | rsyslog-gnutls rsyslog-gssapi rs
The following NEW packages will be installed:
  libestr0 libfastjson4 liblognorm5 rsyslog
0 upgraded, 4 newly installed, 0 to remove and 781 not upgraded.
Need to get 829 kB of archives.
After this operation, 2,280 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kali.download/kali kali-rolling/main amd64 libestr0 amd64 0.1.11-1 [9,204 B]
Get:2 http://kali.download/kali kali-rolling/main amd64 libfastjson4 amd64 1.2304.0-1 [28.9 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 liblognorm5 amd64 2.0.6-4 [67.2 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 rsyslog amd64 8.2302.0-1 [723 kB]
Fetched 829 kB in 23s (36.1 kB/s)
Selecting previously unselected package libestr0:amd64.
```

Step 3: Open rsyslog configuration file
$sudo nano /etc/rsyslog.conf

```
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html


################
#### MODULES ####
################

module(load="imuxsock") # provides support for local system logging
module(load="imklog")   # provides kernel logging support
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
```

Step 4: Add below lines at the end of the file
@192.168.137.50:514
. @@192.168.137.50:514

```
kern.*                          -/var/log/kern.log
mail.*                          -/var/log/mail.log
user.*                          -/var/log/user.log

#
# Emergencies are sent to everybody logged in.
#
*.emerg                         :omusrmsg:*
@192.168.137.50:514
*.*@@192.168.137.50:514
```

Step 5: For the end add these following variables in case when the rsyslog server goes down.
@ActionQueueFileName queue
@ActionQueueMaxDiskSpace 1g
@ActionQueueSaveOnShutdown on
@ActionQueueType LinkedList
@ActionResumeRetryCount -1

Step 6: Then Save and exit the file

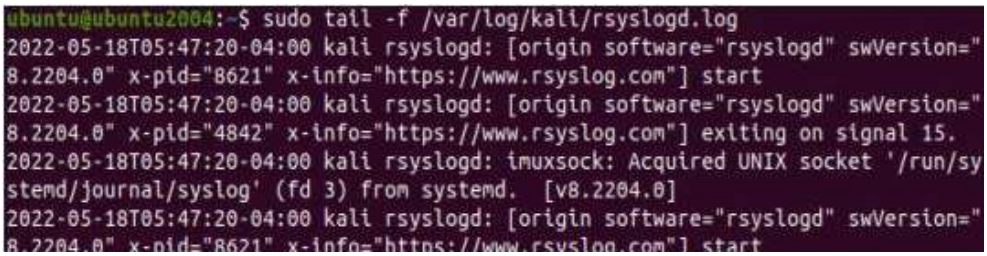
Step 7: restart the rsyslog service
$sudo systemctl restart rsyslog

Verify the logs After the configuration is completed on the client machine, we want to verify that everything went well.

Step 8: Go to your Rsyslog server to verify the logs from your client machine
$ls /var/log/

In my case, the directory named kali is the name of my client machine which I am currently using. We will enter this directory and see something like this:

Step 9: To check logs use the following command: Let's for example inspect rsyslogd.log.
"sudo tail -f /var/log/kali/rsyslogd.log"



Practical No : 8

Aim: Install and Run Splunk on Linux.

Steps:

Step1: Download Splunk Installer
$cd /tmp && wget https://download.splunk.com/products/splunk/releases/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb

```
ubuntu@ubuntu:~$ cd /tmp && wget https://download.splunk.com/products/splunk
/releases/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb
--2023-06-25 08:21:23--  https://download.splunk.com/products/splunk/release
s/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb
Resolving download.splunk.com (download.splunk.com)... 18.66.53.32, 18.66.53
.89, 18.66.53.94, ...
Connecting to download.splunk.com (download.splunk.com)|18.66.53.32|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 263297630 (251M) [binary/octet-stream]
Saving to: 'splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb'

.deb                 5%[>                  ]  13.27M  1.24MB/s    eta eb
splunk-7.1.1-8f0ea 100%[=================>] 251.10M  2.02MB/s    in 3m 10s

2023-06-25 08:24:34 (1.32 MB/s) - 'splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64
.deb' saved [263297630/263297630]
```

Step 2: Install Splunk
$sudo dpkg -i splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb

```
ubuntu@ubuntu:/tmp$ sudo dpkg -i splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.d
eb
[sudo] password for ubuntu:
Selecting previously unselected package splunk.
(Reading database ... 175043 files and directories currently installed.)
Preparing to unpack splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb ...
Unpacking splunk (7.1.1) ...
Setting up splunk (7.1.1) ...
complete
```

Step 3: Enable the Splunk to start at boot

Press enter key till you reach to the end of the agreement, then you have to accept the license agreement by typing "y". Then you have to enter the initial admin password and use this password to access the web portal.

```
ubuntu@ubuntu:/tmp$ sudo /opt/splunk/bin/splunk enable boot-start
SPLUNK SOFTWARE LICENSE AGREEMENT

THIS SPLUNK SOFTWARE LICENSE AGREEMENT ("AGREEMENT") GOVERNS THE LICENSING,
INSTALLATION AND USE OF SPLUNK SOFTWARE. BY DOWNLOADING AND/OR INSTALLING SP
LUNK
SOFTWARE: (A) YOU ARE INDICATING THAT YOU HAVE READ AND UNDERSTAND THIS
AGREEMENT, AND AGREE TO BE LEGALLY BOUND BY IT ON BEHALF OF THE COMPANY,
GOVERNMENT, OR OTHER ENTITY FOR WHICH YOU ARE ACTING (FOR EXAMPLE, AS AN
EMPLOYEE OR GOVERNMENT OFFICIAL) OR, IF THERE IS NO COMPANY, GOVERNMENT OR O
THER
ENTITY FOR WHICH YOU ARE ACTING, ON BEHALF OF YOURSELF AS AN INDIVIDUAL; AND
 (B)
YOU REPRESENT AND WARRANT THAT YOU HAVE THE AUTHORITY TO ACT ON BEHALF OF AN
Do you agree with this license? [y/n]: y
```

```
   * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
Copying '/opt/splunk/etc/openldap/ldap.conf.default' to '/opt/splunk/etc/ope
nldap/ldap.conf'.
Generating RSA private key, 2048 bit long modulus
.............................................+++
.........+++
e is 65537 (0x10001)
writing RSA key

Generating RSA private key, 2048 bit long modulus
.....+++
.........................+++
e is 65537 (0x10001)
writing RSA key

Moving '/opt/splunk/share/splunk/search_mrsparkle/modules.new' to '/opt/splu
nk/share/splunk/search_mrsparkle/modules'.
Init script installed at /etc/init.d/splunk.
Init script is configured to run at boot.
```

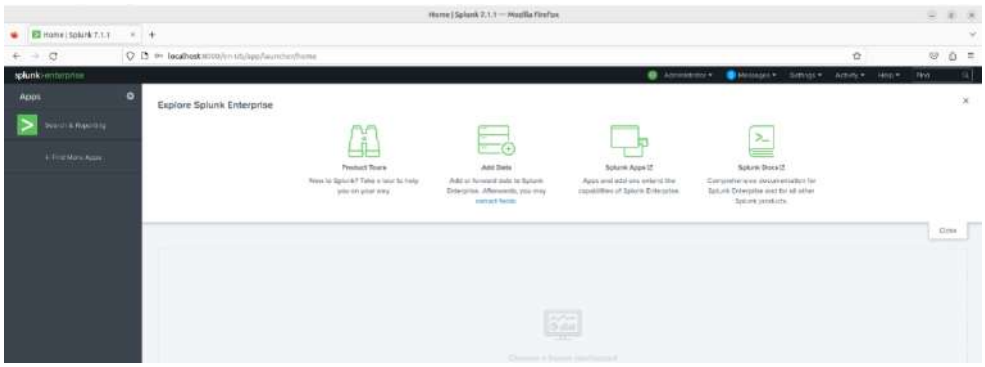
Step 4: Start the Splunk service
$sudo service splunk start

```
ubuntu@ubuntu:/tmp$ sudo service splunk start
ubuntu@ubuntu:/tmp$
```

Step 5: Check splunk service Status
$sudo service splunk status

```
● splunk.service - LSB: Start splunk
     Loaded: loaded (/etc/init.d/splunk; generated)
     Active: active (running) since Sun 2023-06-25 08:46:44 PDT; 46s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 4025 ExecStart=/etc/init.d/splunk start (code=exited, status=0>
      Tasks: 166 (limit: 2214)
     Memory: 487.4M
     CGroup: /system.slice/splunk.service
             ├─4088 splunkd -p 8089 start
             ├─4089 [splunkd pid=4088] splunkd -p 8089 start [process-runne>
             ├─4100 mongod --dbpath=/opt/splunk/var/lib/splunk/kvstore/mong>
             ├─4172 /opt/splunk/bin/python -O /opt/splunk/lib/python2.7/sit>
             ├─4174 /opt/splunk/bin/splunkd instrument-resource-usage -p 80>
             ├─4408 [splunkd pid=4088] [search-launcher]
             └─4409 [splunkd pid=4088] [search-launcher] [process-runner]

Jun 25 08:46:31 ubuntu splunk[4026]:         All installed files intact.
Jun 25 08:46:31 ubuntu splunk[4026]:         Done
Jun 25 08:46:31 ubuntu splunk[4026]: All preliminary checks passed.
Jun 25 08:46:31 ubuntu splunk[4026]: Starting splunk server daemon (splunkd>
Jun 25 08:46:31 ubuntu splunk[4026]: Done
Jun 25 08:46:44 ubuntu splunk[4026]: Waiting for web server at http://127.0>
Jun 25 08:46:44 ubuntu splunk[4026]: If you get stuck, we're here to help.
lines 1-23
```

Step 6: Splunk will be started at port 8000. You can access the application via URL “http://localhost:8000/“. To logged in into the app enter username as “admin” then enter your password. In my case the password is “admin!123”



Practical No : 9

Aim: Install and Configure ELK on Linux.

Steps:

Step 1: write the below command and update and install the jdk
$sudo apt update

```
ubuntu@ubuntu:~/Desktop$ sudo apt update
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [
111 kB]
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable/main i386 Packages [8
1.3 kB]
Hit:5 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:7 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 206 kB in 3s (63.5 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
144 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Install java
$sudo apt install default-jre

```
ubuntu@ubuntu:~/Desktop$ sudo apt install default-jre
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java default-jre-headless fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni openjdk-11-jre
  openjdk-11-jre-headless
Suggested packages:
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java default-jre default-jre-headless fonts-dejavu-extra
  java-common libatk-wrapper-java libatk-wrapper-java-jni openjdk-11-jre
  openjdk-11-jre-headless
0 upgraded, 9 newly installed, 0 to remove and 144 not upgraded.
```

Step 2: check the java version by this command
$java –version

```
ubuntu@ubuntu:~/Desktop$ java --version
openjdk 11.0.19 2023-04-18
OpenJDK Runtime Environment (build 11.0.19+7-post-Ubuntu-0ubuntu120.04.1)
OpenJDK 64-Bit Server VM (build 11.0.19+7-post-Ubuntu-0ubuntu120.04.1, mixed moc
e, sharing)
ubuntu@ubuntu:~/Desktop$
```

Part 2: Install and Configure the Elasticsearch Elastic Search
Elasticsearch store logs coming from external sources and offers real-time distributed search and analytics with the RESTful web interface.

Step 1: Download and install the GPG signing key.
$curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –

```
ubuntu@ubuntu:~/Desktop$ curl -fsSL https://artifacts.elastic.co/GPG-KEY-elastic
search | sudo apt-key add -
OK
```

Step 2: Set up the Elasticsearch repository on your system by running the below command.
$echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list

```
ubuntu@ubuntu:~/Desktop$ echo "deb https://artifacts.elastic.co/packages/7.x/apt
 stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
```

Step 3: Update the repository cache and then install the Elasticsearch package.
$sudo apt update

```
ubuntu@ubuntu:~/Desktop$ sudo apt update
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [
111 kB]
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable/main i386 Packages [8
1.3 kB]
Hit:5 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:7 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 206 kB in 3s (63.5 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
144 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

$sudo apt install elasticsearch

```
ubuntu@ubuntu:~/Desktop$ sudo apt install elasticsearch
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 144 not upgraded.
Need to get 317 MB of archives.
After this operation, 530 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsea
rch amd64 7.17.10 [317 MB]
Fetched 47.1 MB in 1min 20s (587 kB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 189994 files and directories currently installed.)
Preparing to unpack .../elasticsearch_7.17.10_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (7.17.10) ...
Setting up elasticsearch (7.17.10) ...
### NOT starting on installation, please execute the following statements to con
figure elasticsearch service to start automatically using systemd
```

Step 4: Edit the Elasticsearch configuration file to set the cluster name for Graylog set up.
$sudo nano /etc/elasticsearch/elasticsearch.yml
Uncomment network.host:localhost http.port:9200

```
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: localhost
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
```

Step 5: Next, start the Elasticsearch service with the systemctl. Give Elasticsearch little time to start up otherwise, you can get errors about not being able to connect to it.
$sudo systemctl start elasticsearch

```
ubuntu@ubuntu:~/Desktop$ sudo systemctl start elasticsearch
ubuntu@ubuntu:~/Desktop$
```

Step 6: Now, run the below command. It will enable Elasticsearch to start every time your server boots.
$sudo systemctl enable elasticsearch

```
ubuntu@ubuntu:~/Desktop$ sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/
systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.servic
e → /lib/systemd/system/elasticsearch.service.
ubuntu@ubuntu:~/Desktop$
```

Step 7: You will then test whether your Elasticsearch service is running. Do it by sending an HTTP request:
$curl -X GET "localhost:9200"

```
ubuntu@ubuntu:~/Desktop$ curl -X GET "localhost:9200"
{
  "name" : "ubuntu",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "dnVrXuFqQluC0sVRlKxZ3w",
  "version" : {
    "number" : "7.17.10",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "fecd68e3150eda0c307ab9a9d7557f5d5fd71349",
    "build_date" : "2023-04-23T05:33:18.138275597Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Practical No : 10

Aim: Install and Configure ELK on Linux

Steps:

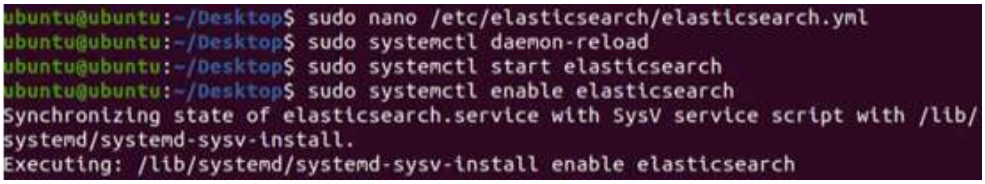
Step 1: Install Java and Els

Step 2:Install Java and Els (Practical 8)

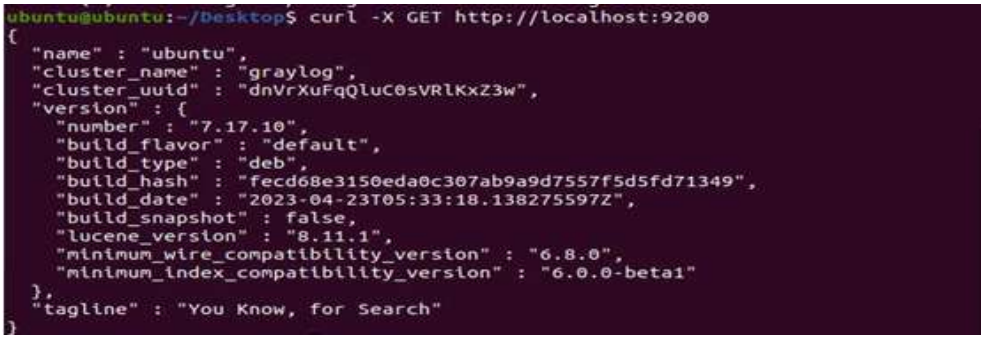
Step 3: Edit the Elasticsearch configuration file to set the cluster name for Graylog set up.
$sudo nano /etc/elasticsearch/elasticsearch.yml

Step 4: Set the cluster name as graylog, as shown below. Then, uncomment the line and below add this line "action.auto_create_index: false" **then save.**

Step 5: Start the Elasticsearch service to read the new configurations.
$sudo systemctl daemon-reload
$sudo systemctl start elasticsearch
$sudo systemctl enable elasticsearch



Step 6: Elastic search should be now listening on port 9200. Use the curl command to check the Elasticsearch's response.
$curl -X GET http://localhost:9200



Step 7: Install MongoDB

MongoDB acts as a database for storing Graylog's configuration. Graylog requires MongoDB v3.6, 4.0 or 4.2. Unfortunately, MongoDB's official repository doesn't have the required MongoDB versions for Ubuntu 20.04. So, we will install MongoDB v3.6 from the Ubuntu base repository.

Step 8: Update the system
$sudo apt update

```
ubuntu@ubuntu:~/Desktop$ sudo apt update
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Fetched 222 kB in 4s (51.6 kB/s)
Reading package lists... Done
```

$sudo apt install -y mongodb-server

```
ubuntu@ubuntu:~/Desktop$ sudo apt install -y mongodb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
mongodb-server is already the newest version (1:3.6.9+really3.6.8+90~g8e540c0b6d
-0ubuntu5.3).
0 upgraded, 0 newly installed, 0 to remove and 144 not upgraded.
```

Step 9: Start the MongoDB and enable it on the system start-up.
$sudo systemctl start mongodb
$sudo systemctl enable mongodb

```
ubuntu@ubuntu:~/Desktop$ sudo systemctl enable mongodb
Synchronizing state of mongodb.service with SysV service script with /lib/system
d/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mongodb
```

Step 10: Install GrayLog Server
GrayLog Server reads data from Elasticsearch for search queries comes from users and then displays it for them through the Graylog web interface.

Step 11: Download and install the Graylog 3.3 repository configuration package.
$wget https://packages.graylog2.org/repo/packages/graylog-4.2- repository_latest.deb

```
ubuntu@ubuntu:~/Desktop$ wget https://packages.graylog2.org/repo/packages/graylo
g-4.2-repository_latest.deb
--2023-06-25 23:16:39--  https://packages.graylog2.org/repo/packages/graylog-4.2
-repository_latest.deb
Resolving packages.graylog2.org (packages.graylog2.org)... 104.21.88.209, 172.67
.153.95, 2606:4700:3035::ac43:995f, ...
Connecting to packages.graylog2.org (packages.graylog2.org)|104.21.88.209|:443..
. connected.
HTTP request sent, awaiting response... 302 Found
```

$sudo dpkg -i graylog-3.3-repository_latest.deb

```
ubuntu@ubuntu:~/Desktop$ sudo dpkg -i graylog-4.2-repository_latest.deb
Selecting previously unselected package graylog-4.2-repository.
(Reading database ... 191185 files and directories currently installed.)
Preparing to unpack graylog-4.2-repository_latest.deb ...
Unpacking graylog-4.2-repository (1-4) ...
Setting up graylog-4.2-repository (1-4) ...
```

Step 12: Update the repository cache.
$sudo apt update

```
ubuntu@ubuntu:~/Desktop$ sudo apt update
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadat
a [59.9 kB]
```

Step 13: Install the Graylog server using the following command.
$sudo apt install -y graylog-server

```
ubuntu@ubuntu:~/Desktop$ sudo apt install -y graylog-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  graylog-server
0 upgraded, 1 newly installed, 0 to remove and 144 not upgraded.
Need to get 197 MB of archives.
After this operation, 218 MB of additional disk space will be used.
Ign:1 https://packages.graylog2.org/repo/debian stable/4.2 amd64 graylog-server
```

Step 14: You must set a secret to secure the user passwords. Use the pwgen command to generate the secret.
$pwgen -N 1 -s 96

```
ubuntu@ubuntu:~/Desktop$ sudo apt install pwgen
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  pwgen
0 upgraded, 1 newly installed, 0 to remove and 144 not upgraded.
Need to get 18.1 kB of archives.
After this operation, 52.2 kB of additional disk space will be used.
```

```
ubuntu@ubuntu:~/Desktop$ pwgen -N 1 -s 96
dHhrek7amsHYKJ4l0IKuJC6wOPbVZ0nCY7Ea4fPBTzQT5xWmrSpnvHY6Q1ePeBVFs8R2mNEH18RDRqDi
W4DxLvL4xb38DOe8
```

Step 15: sudo gedit /etc/graylog/server/server.conf edit the conf file and put Then, place the secret like below.
sudo nano /etc/graylog/server/server.conf

```
# You MUST set a secret to secure/pepper the stored user passwords here. Use at>
# Generate one by using for example: pwgen -N 1 -s 96
# ATTENTION: This value must be the same on all Graylog nodes in the cluster.
# Changing this value after installation will render all user sessions and encr>
password_secret = dHhrek7amsHYKJ4l0IKuJC6wOPbVZ0nCY7Ea4fPBTzQT5xWmrSpnvHY6Q1ePe>
```

Step 16: Now, generate a hash (sha256) password for the root user (not to be confused with the system user, the root user of graylog is admin).

You will need this password to login to the Graylog web interface. Admin's password can't be changed using the web interface. So, you must edit this variable to set.

Replace password with the choice of your password. Put this command in terminal "echo -n password | sha256sum"

```
ubuntu@ubuntu:~/Desktop$ echo -n yourpassword | sha256sum
e3c652f0ba0b4801205814f8b6bc49672c4c74e25b497770bb89b22cdeb4e951  -
```

Step 17: Edit the server.conf file again.in terminal
$sudo nano /etc/graylog/server/server.conf

```
# You MUST set a secret to secure/pepper the stored user passwords here. Use at>
# Generate one by using for example: pwgen -N 1 -s 96
# ATTENTION: This value must be the same on all Graylog nodes in the cluster.
# Changing this value after installation will render all user sessions and encr>
password_secret = dHhrek7amsHYKJ4l0IKuJC6wOPbVZ0nCY7Ea4fPBTzQT5xWmrSpnvHY6Q1ePe>
```

Part 18: Setup Graylog web interface
From version Graylog 2.x, the web interface is being served directly by the Graylog server.
Step 1: Enable the Graylog web interface by editing the server.conf file.
"sudo gedit /etc/graylog/server/server.conf"

Put http_bind_address = 192.168.0.10:9000
http_external_uri = http://public_ip:9000/

```
# Default: 127.0.0.1:9000
http_bind_address = 192.168.186.129:9000
#http_bind_address = [2001:db8::1]:9000

#### HTTP publish URI
#
# The HTTP URI of this Graylog node which is used to communicate with the othe
```

Step 19: Start and enable the Graylog service.
Place the below command
$sudo systemctl daemon-reload
$sudo systemctl start graylog-server
$sudo systemctl enable graylog-server

```
ubuntu@ubuntu:~/Desktop$ sudo systemctl enable  graylog-server
Synchronizing state of graylog-server.service with SysV service script with /lib
/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable graylog-server
Created symlink /etc/systemd/system/multi-user.target.wants/graylog-server.servi
ce → /lib/systemd/system/graylog-server.service.
```

Step 20: Keep looking Graylog server startup logs. This log will be useful for you to troubleshoot Graylog in case of any issues.
$sudo tail -f /var/log/graylog-server/server.log

Step 21: On the successful start of the Graylog server, you should get the following message in the log file. You will able to see the log file.

2020-08-03T16:03:06.326-04:00 INFO [ServerBootstrap] Graylog server up and running.

Access Graylog

The Graylog web interface will now be listening on port 9000. Open your browser and point it to.

“http://ip.add.re.ss:9000” type in browser.