

**University of Mumbai**



**PRACTICAL JOURNAL – ELECTIVE II**

**PSIT4P3d  
Security Operations Center**

**SUBMITTED  
BY**

**SAWANT SHUBHAM SUDHIR**

**SEAT NO: 40392**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR QUALIFYING M.Sc. (I.T.) PART-II  
(SEMESTER – IV) EXAMINATION**

**2022-2023**

**DEPARTMENT OF INFORMATION TECHNOLOGY  
3RD FLOOR, DR. SHANKAR DAYALSHARMA BHAVAN,  
VIDYANAGRI, SANTACRUZ(EAST),  
MUMBAI – 400098.**

**University of Mumbai**



## Department of Information Technology

### Certificate

This is to certify that **Mr. SAWANT SHUBHAM SUDHIR** Seat No. **40392** studying in **Master of Science in Information Technology Part II Semester IV** has satisfactorily completed the Practical of **PSIT4P3d Security Operations Center** as prescribed by University of Mumbai, during the academic year **2022-23**.

\_\_\_\_\_  
Signature  
Subject-In-Charge

\_\_\_\_\_  
Signature  
Head of the Department

\_\_\_\_\_  
Signature  
External Examiner

College Seal: \_\_\_\_\_

Date: \_\_\_\_\_

**INDEX**

<b>Sr. No</b>	<b>Topic</b>	<b>Page No.</b>	<b>Date</b>	<b>Sign</b>
<b>1</b>	Encrypting and Decrypting Data Using OpenSSL	<b>4</b>	/ /2023	
<b>2</b>	Demonstrate the use of Snort and Firewall Rules	<b>6</b>	/ /2023	
<b>3</b>	Demonstrate Extract an Executable from a PCAP	<b>11</b>	/ /2023	
<b>4</b>	Demonstrate Analysis of DNS Traffic	<b>14</b>	/ /2023	
<b>5</b>	Create your own syslog Server	<b>20</b>	/ /2023	
<b>6</b>	Configure your Linux system to send syslog messages to a syslog server and Read them	<b>22</b>	/ /2023	
<b>7</b>	Install and Run Splunk on Linux	<b>24</b>	/ /2023	
<b>8</b>	Install and Configure ELK on Linux	<b>27</b>	/ /2023	
<b>9</b>	Install and Configure GrayLog on Linux	<b>30</b>	/ /2023	
<b>10</b>	Demonstrate Conversion of Data into a Universal Format	<b>35</b>	/ /2023	

**Practical No.1**

**Aim: Encrypting and Decrypting Data Using OpenSSL.****Encrypting Messages with OpenSSL**

OpenSSL can be used as a standalone tool for encryption. While many encryption algorithms can be used, this lab focuses on AES. To use AES to encrypt a text file directly from the command line using OpenSSL, follow the steps below:

**Step 1: Encrypting a Text File**

a. Log into CyberOPS Workstation VM.

b. Open a terminal window.

c. Because the text file to be encrypted is in the /home/analyst/lab.support.files/ directory, change to that directory:

```
[analyst@secOps ~]$ cd ./lab.support.files/
```

```
[analyst@secOps lab.support.files]$
```

d. Type the command below to list the contents of the encrypted letter\_to\_grandma.txt text file on the screen:

```
[analyst@secOps lab.support.files]$ cat letter_to_grandma.txt
```

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

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

e. From the same terminal window, issue the command below to encrypt the text file. The command will use AES-256 to encrypt the text file and save the encrypted version as message.enc. OpenSSL will ask for a password and for password confirmation. Provide the password as requested and be sure to remember the password.

```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -in letter_to_grandma.txt -out message.enc
```

```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -in letter_to_grandma.txt -o
ut message.enc
Enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
[analyst@secOps lab.support.files]$
```

f. cat message.enc

```
[analyst@secOps lab.support.files]$ cat message.enc
Salted__
x+,
L
m
!
[analyst@secOps lab.support.files]$
```

g. To make the file readable, run the OpenSSL command again, but this time add the -a option. The -a option tells OpenSSL to encode the encrypted message using a different encoding method of Base64 before storing the results in a file.

**[analyst@secOps lab.support.files]\$ openssl aes-256-cbc -a -in letter\_to\_grandma.txt -out message.enc**

```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -in letter_to_grandma.txt
-out message.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
[analyst@secOps lab.support.files]$
```

**h. cat message.enc**

```
[analyst@secOps lab.support.files]$ cat message.enc
U2FsdGVkX1+UKTA1/M7OnWohih3G17aCGLufs2Fn8R2IP2wHLSrU85rB2qH318/d
/T13jkC2wRzmXNRI2yt77nCw1vxZ/ntU14zNDLgfPQvPm80Tv9LTt7RrHEMI/15D
zsNbL9yPI0DuqdA1kvmxqEqX8WdV6gfAKTgqYnvNWrwB3PvIGXBc89nOU1jSC0ea
nTTChJMY7IdaPSqmZVo+qzdgSPMFJKcmbOI/Gs+DLs5TbJSIbLURvT8eH3R6AHu6
uxAilyukHH//32itz5R48UrY923CQ9bdFVaBc7y0ZPH1uLSLTh1FrCLuv9nCv66S
MCKdSui1qTxN+1neAarcbiz+SJPznyURKDTxAwt1G2U=
```

## Step2: Decrypting Messages with OpenSSL

With a similar OpenSSL command, it is possible to decrypt message.enc.

a. Use the command below to decrypt message.enc:

**[analyst@secOps lab.support.files]\$ openssl aes-256-cbc -a -d -in message.enc -out decrypted\_letter.txt**

```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -d -in message.enc -out d
ecrypted_letter.txt
enter aes-256-cbc decryption password:
```

b. OpenSSL will ask for the password used to encrypt the file. Enter the same password again.

c. When OpenSSL finishes decrypting the message.enc file, it saves the decrypted message in a text file called decrypted\_letter.txt. Use the cat display the contents of decrypted\_letter.txt:

**[analyst@secOps lab.support.files]\$ cat decrypted\_letter.txt**

```
[analyst@secOps lab.support.files]$ 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.
```

## Practical No.2

**Aim:** Demonstrate the use of Snort and Firewall Rules.

### **Part 1: Preparing the Virtual Environment**

- Launch Oracle VirtualBox and change the CyberOps Workstation for Bridged mode
- Launch the CyberOps Workstation VM, open a terminal and configure its network by executing the sh script.

[analyst@secOps ~]\$ sudo ./lab.support.files/scripts/configure\_as\_dhcp.sh

```
[analyst@secOps ~]$ sudo ./lab.support.files/scripts/configure_as_dhcp.sh
[sudo] password for analyst:
Configuring the NIC to request IP info via DHCP...
Requesting IP information...
IP Configuration successful.

[analyst@secOps ~]$
```

- Use the **ifconfig** command to verify CyberOps Workstation VM now has an IP address on your local network. You can also test connectivity to a public webserver by pinging [www.cisco.com](http://www.cisco.com). Use Ctrl+C to stop the pings.

```
[analyst@secOps ~]$ ping www.cisco.com
PING e2867.dsca.akamaiedge.net (96.6.39.125) 56(84) bytes of data:
64 bytes from a96-6-39-125.deploy.static.akamaitechnologies.com (96.6.39.125): icmp_seq=1 ttl=54 time=74.4 ms
64 bytes from a96-6-39-125.deploy.static.akamaitechnologies.com (96.6.39.125): icmp_seq=2 ttl=54 time=75.2 ms
64 bytes from a96-6-39-125.deploy.static.akamaitechnologies.com (96.6.39.125): icmp_seq=3 ttl=54 time=70.3 ms
64 bytes from a96-6-39-125.deploy.static.akamaitechnologies.com (96.6.39.125): icmp_seq=4 ttl=54 time=110 ms
64 bytes from a96-6-39-125.deploy.static.akamaitechnologies.com (96.6.39.125): icmp_seq=5 ttl=54 time=114 ms
```

### **Part 2: Firewall and IDS Logs**

Step 1: Real-Time IDS Log Monitoring

- From the CyberOps Workstation VM, run the script to start mininet.

```
[analyst@secOps ~]$ sudo ./lab.support.files/scripts/cyberops_extended_topo_no_fw.py
[analyst@secOps ~]$ sudo ./lab.support.files/scripts/cyberops_extended_topo_no_fw.py
[sudo] password for analyst:
Sorry, try again.
[sudo] password for analyst:
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
R1 R4 H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11
*** Starting controllers
*** Starting switches
*** Add routes
*** Post configure switches and hosts
*** Starting CLI:
mininet>
```

- From the mininet prompt, open a shell on R1 using the command below:  
**mininet> xterm R1**

```
mininet> xterm R1
mininet> [redacted]

"Node: R1"
[root@secOps analyst]# [redacted]
```

c. From R1's shell, start the Linux-based IDS, Snort.

```
[root@secOps analyst]# ./lab.support.files/scripts/start_snort.sh
ved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.8.1
Using PCRE version: 8.42 2018-03-20
Using ZLIB version: 1.2.11

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.0 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_PDP Version 1.0 <Build 1>
Commencing packet processing (pid=1448)
[redacted]
```

d. From the CyberOps Workstation VM mininet prompt, open shells for hosts H5 and H10.

```
mininet> xterm H5
[root@secOps analyst]# exit

mininet> xterm H10
```

e. H10 will simulate a server on the Internet that is hosting malware. On H10, run the mal\_server\_start.sh script to start the server.

```
[root@secOps analyst]# ./lab.support.files/scripts/mal_server_start.sh
[root@secOps analyst]# ./lab.support.files/scripts/mal_server_start.sh
[root@secOps analyst]# [redacted]
```

e. H10 will simulate a server on the Internet that is hosting malware. On H10, run the mal\_server\_start.sh script to start the server.


f. On H10, use netstat with the -tunpa options to verify that the web server is running. When used as shown below, netstat lists all ports currently assigned to services:

```
[root@secOps analyst]# netstat -tunpa
[root@secOps analyst]# netstat -tunpa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:80               0.0.0.0:*               LISTEN
1517/nginx: master
[root@secOps analyst]# [redacted]
```

g. In the R1 terminal window, an instance of Snort is running. To enter more commands on R1, open another R1 terminal by entering the xterm R1 again in the CyberOps Workstation VM terminal window. You may also want to arrange the terminal windows so that you can see and interact with each device.

h. In the new R1 terminal tab, run the tail command with the -f option to monitor the /var/log/snort/alert file in real-time. This file is where snort is configured to record alerts.

```
[root@secOps analyst]# tail -f /var/log/snort/alert
```



```

Node: R1
[root@secOps analyst]# tail -f /var/log/snort/alert

```

i. From H5, use the wget command to download a file named Nimda.Amm.exe. Designed to download content via HTTP, wget is a great tool for downloading files from web servers directly from the command line.

**[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe**

**Or**

**curl -O 209.165.202.133:6666/W32.Nimda.Amm.exe**

```

[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe
--2023-06-25 01:49:35-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... connected.
HTTP request sent, awaiting response... 200 OK
Length: 345088 (337K) [application/octet-stream]
Saving to: 'W32.Nimda.Amm.exe'

W32.Nimda.Amm.exe 100%[=====>] 337.00K --.-KB/s in 0.008s

2023-06-25 01:49:35 (40.0 MB/s) - 'W32.Nimda.Amm.exe' saved [345088/345088]

[root@secOps analyst]# curl -O 209.165.202.133:6666/W32.Nimda.Amm.exe
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 337k 100 337k    0     0  9.9M      0 --:--:-- --:--:-- --:--:-- 10.2M
[root@secOps analyst]#

```

j. As the malicious file was transiting R1, the IDS, Snort, was able to inspect its payload. The payload matched at least one of the signatures configured in Snort and triggered an alert on the second R1 terminal window (the tab where tail -f is running). The alert entry is show below. Your timestamp will be different:

```

0) {TCP} 209.165.200.235:43732 -> 209.165.202.133:6666
06/25-01:49:55.078139  [**] [1:1000003:0] Malicious Server Hit! [**] [Priority:
2) {TCP} 209.165.200.235:43734 -> 209.165.202.133:6666

```

On H5, use the tcpdump command to capture the event anddownload the malware file again so you can capture the transaction.type command.

**“tcpdump -i H5-eth0 -w nimda.download.pcap&”**

```

[root@secOps analyst]# tcpdump -i H5-eth0 -w nimda.download.pcap&
[1] 1595
[root@secOps analyst]# tcpdump: listening on H5-eth0, link-type EN10MB (Ethernet)
), capture size 262144 bytes

```

Press ENTER a few times to regain control of the shell while tcpdump runs in background.

Now that tcpdump is capturing packets, download the malware again. On H5, re-run the command.

**“curl -O 209.165.202.133:6666/W32.Nimda.Amm.exe”**

```

[root@secOps analyst]# curl -O 209.165.202.133:6666/W32.Nimda.Amm.exe
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 337k 100 337k    0     0  8219k      0 --:--:-- --:--:-- --:--:-- 8425k

```

m. Stop the capture by bringing tcpdump to foreground with the fg Because tcpdump was the only process sent to background, there is no need to specify the PID. Stop the tcpdump process with Ctrl+C. The tcpdump process stops and displays a summary of the capture. The number of packets may be different for your capture.

**[root@secOps analyst]# fg tcpdump -i h5-eth0 -w nimda.download.pcap**



```
[root@secOps analyst]# fg tcpdump -i H5-eth0 -w nimda.download.pcap
tcpdump -i H5-eth0 -w nimda.download.pcap
^C43 packets captured
43 packets received by filter
0 packets dropped by kernel
```

n. On H5, Use the ls command to verify the pcap file was in fact saved to disk and has size greater than zero:

```
[root@secOps analyst]# ls -l
```

```
total 1400
```

```
[root@secOps analyst]# ls -l
total 700
drwxr-xr-x 2 analyst analyst 4096 Mar 22 2018 Desktop
drwxr-xr-x 3 analyst analyst 4096 Mar 22 2018 Downloads
drwxr-xr-x 9 analyst analyst 4096 Jun 25 00:34 lab.support.files
-rw-r--r-- 1 root root 349393 Jun 25 02:01 nimda.download.pcap
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 root root 345088 Jun 25 01:58 W32.Nimda.Amm.exe
```

## Step 2: Tuning Firewall Rules Based on IDS Alerts

a. In the CyberOps Workstation VM, start a third R1 terminal window.

```
mininet > xterm R1
```

b. In the new R1 terminal window, use the iptables command to list the chains and their rules currently in use:

```
[root@secOps ~]# iptables -L -v
```

```
[root@secOps analyst]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination

[root@secOps analyst]# █
```

c. Connections to the malicious server generate packets that must transverse the iptables firewall on R1. Packets traversing the firewall are handled by the FORWARD rule and therefore, that is the chain that will receive the blocking rule. To keep user computers from connecting to the malicious server identified in Step 1, add the following rule to the FORWARD chain on R1:

d. Use the iptables command again to ensure the rule was added to the FORWARD chain. The CyberOps Workstation VM may take a few seconds to generate the output:

```
[root@secOps ~]# iptables -I FORWARD -p tcp -d 209.165.202.133 --dport 6666 -j DROP
```

```
[root@secOps analyst]# iptables -I FORWARD -p tcp -d 209.165.202.133 --dport 66
66 -j DROP
[root@secOps analyst]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination
    0    0 DROP      tcp  --  any    any    anywhere                209.165.202.
133      tcp dpt:6666

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination
```

e. On H5, try to download the file again:

```

[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe
--2023-06-25 02:15:23-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... failed: Connection timed out.
Retrying.

--2023-06-25 02:17:33-- (try: 2) http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... ^C

```

### Part 3: Terminate and Clear Mininet Process

a. Navigate to the terminal used to start Mininet. Terminate the Mininet by entering quit in the main CyberOps VM terminal window.

```

mininet> quit
*** Stopping 0 controllers

*** Stopping 7 terms
*** Stopping 15 links
.....
*** Stopping 3 switches
S5 S9 S10
*** Stopping 13 hosts
R1 R4 H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11
*** Done

```

b. After quitting Mininet, clean up the processes started by Mininet. Enter the password cyberops when prompted.

[analyst@secOps scripts]\$ sudo mn -c

```

[analyst@secOps ~]$ sudo mn -c
[sudo] password for analyst:
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd
ovs-controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd
ovs-controller udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | grep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | grep -o '([-.[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn

```

## Practical No.3

**Aim:** Demonstrate Extract an Executable from a PCAP.

**Step 1:** open terminal and write this command “cd ./lab.support.files/pcaps/”

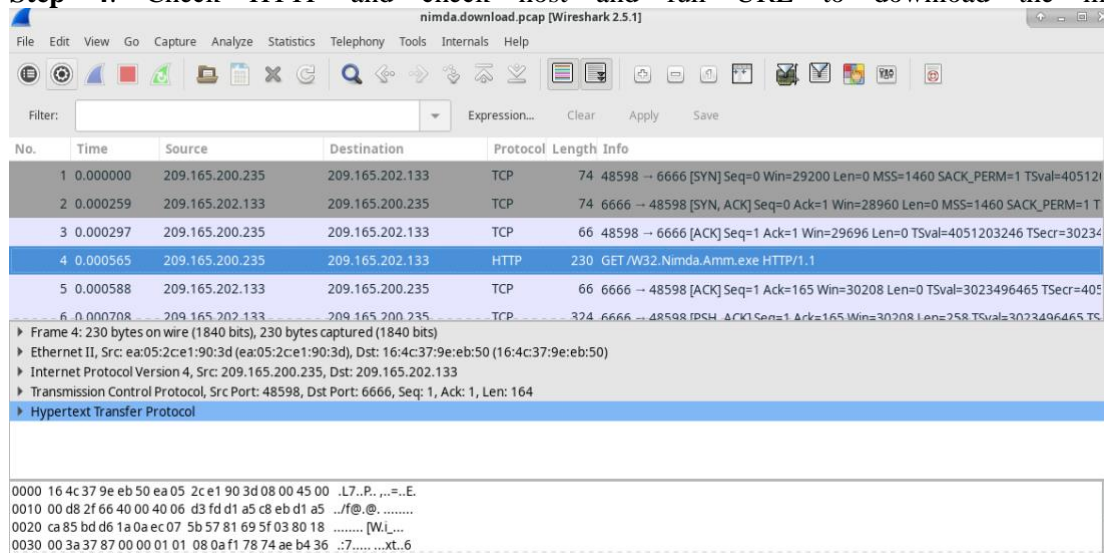
**Step 2:** write “ls -l” list command.

```
[analyst@secOps ~]$ cd lab.support.files/pcaps
[analyst@secOps pcaps]$ ls -l
total 4028
-rw-r--r-- 1 analyst analyst 371462 Mar 21 2018 nimda.download.pcap
-rw-r--r-- 1 analyst analyst 3750153 Mar 21 2018 wannacry_download.pcap
```

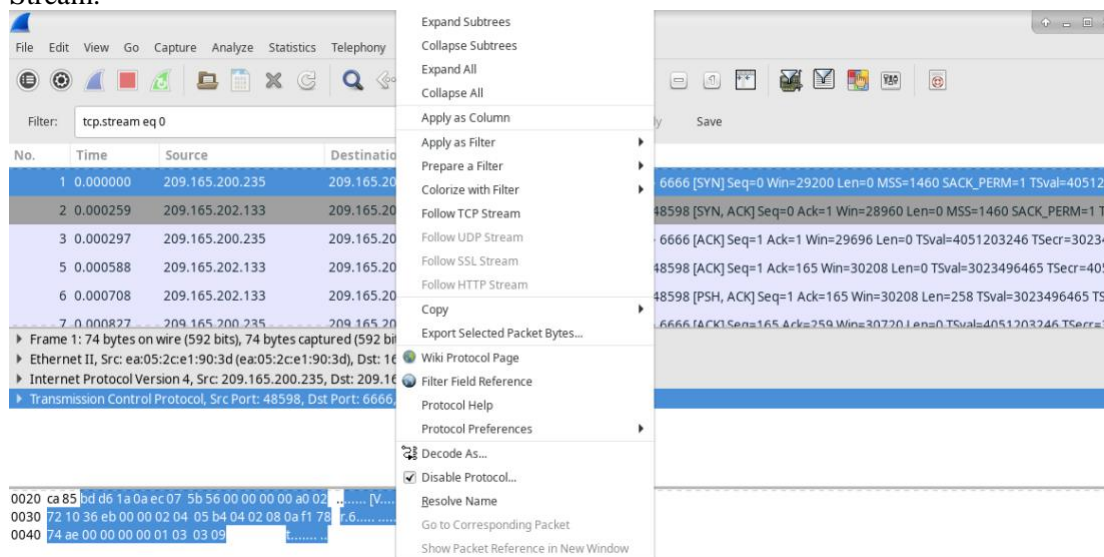
**Step 3:** On command prompt “wireshark-gtk nimda.download.pcap” (This will open the wireshark UI)

```
[analyst@secOps pcaps]$ wireshark-gtk nimda.download.pcap
```

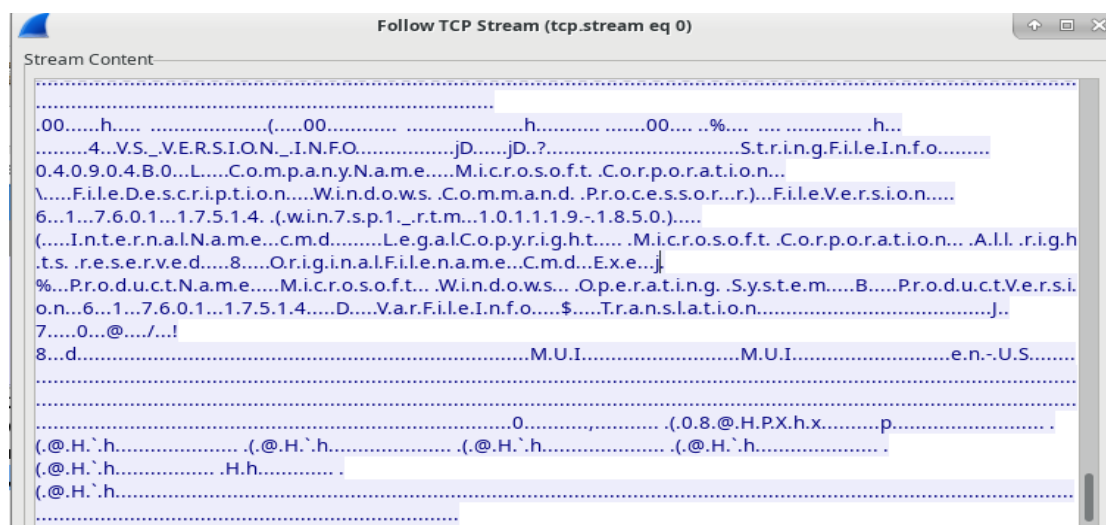
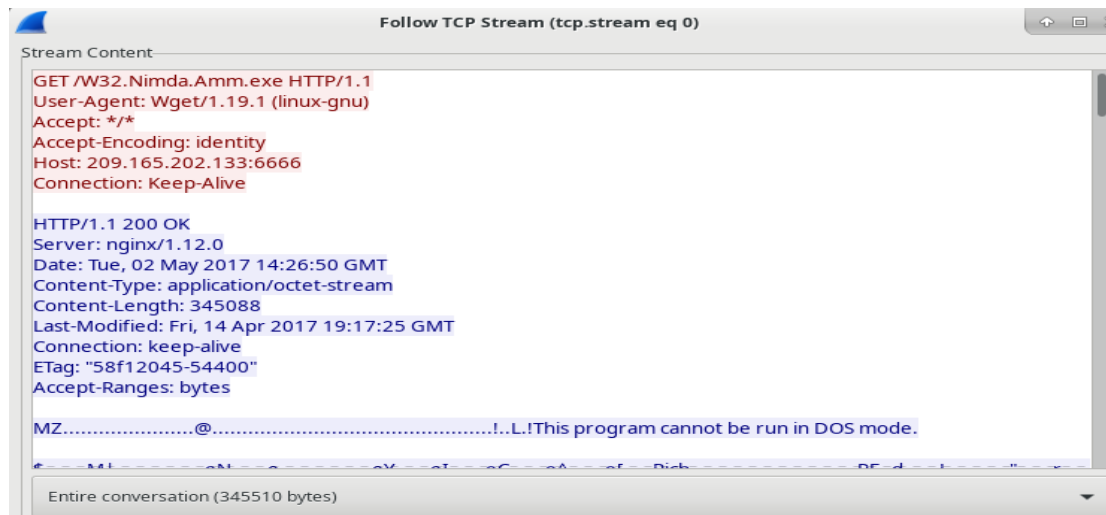
**Step 4:** Check HTTP and check host and full URL to download the malware file.



**Step 5:** right click on TCP which shows top on the list. Then click on Follow TCP Stream.

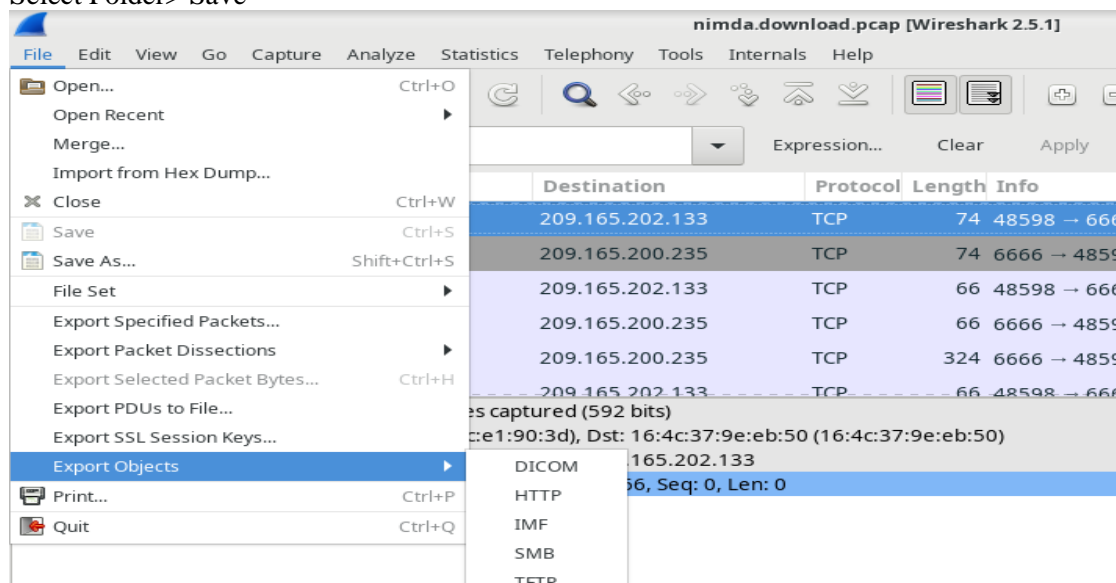


**Step 6:** Check the original file name in the Follow TCP Stream window.

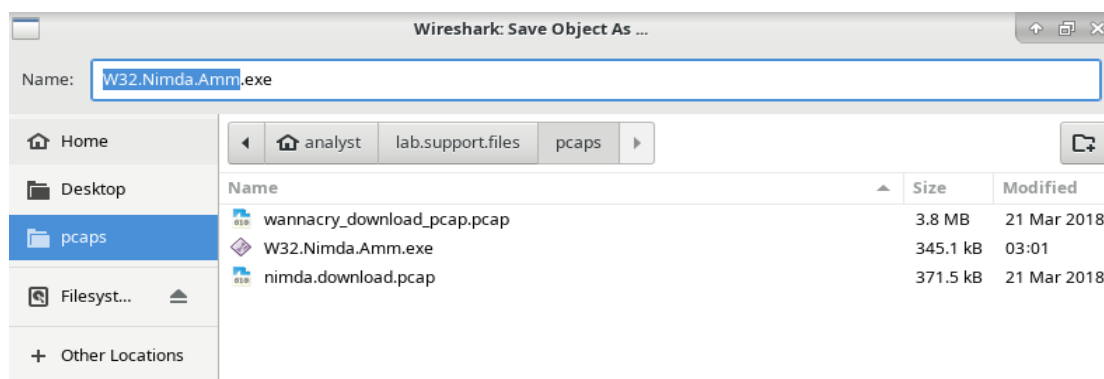
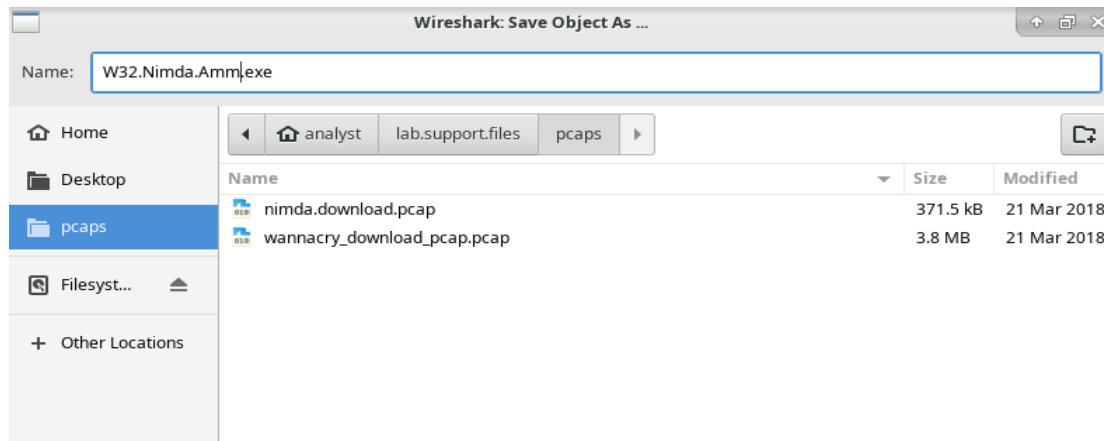
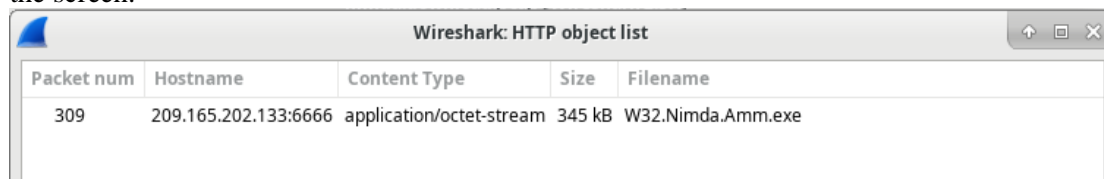


**Step 7:** Now we need to download and check that file by uploading to an online virustotal website.

Find exe file from HTTP>click file> select export obj > Select exe File >Save as > Select Folder> Save



In the HTTP object list window, select the W32.Nimda.Amm.exe file and click Save As at the bottom of the screen.



**Step 8:** In command prompt “ls -l” to check if the file is saved or not.

```
[analyst@sec0ps pcaps]$ ls -l
total 4368
-rw-r--r-- 1 analyst analyst 371462 Mar 21 2018 nimda.download.pcap
-rw-r--r-- 1 analyst analyst 345088 Jun 25 03:01 W32.Nimda.Amm.exe
-rw-r--r-- 1 analyst analyst 3750153 Mar 21 2018 wannacry_download_pcap.pcap
```

**Step 9:** to check the file information put this command “file W32.Nimda.Amm.exe”

```
[analyst@sec0ps pcaps]$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows
```

## Practical No.4

**Aim:** Demonstrate Analysis of DNS Traffic.

**Part 1: Capture DNS Traffic**

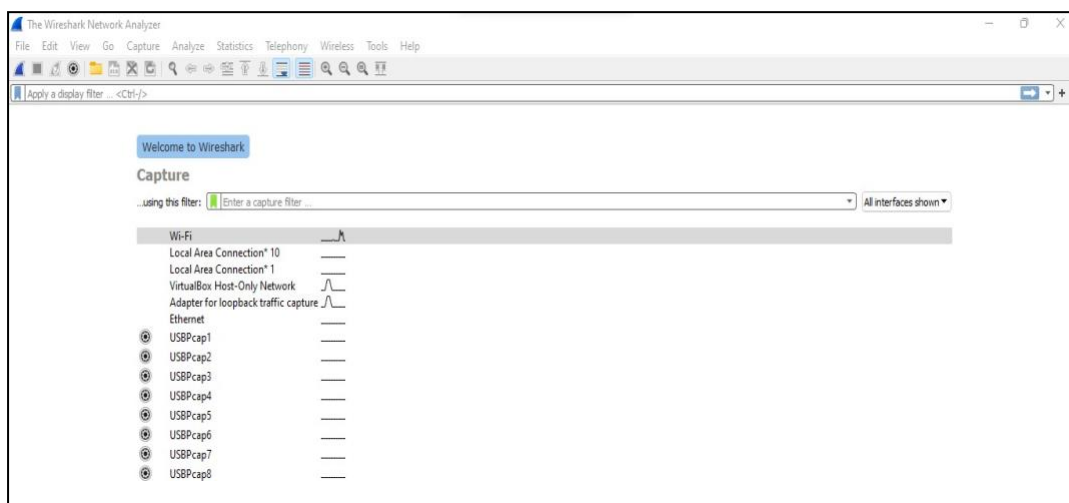
**Part 2: Explore DNS Query Traffic**

**Part 3: Explore DNS Response Traffic**

**Solution:**

**Part 1: Capture DNS Traffic**

Step 1: Open **Wireshark** and start a Wireshark capture by double clicking a network interface with traffic.



Step 2: At the Command Prompt, enter **ipconfig /flushdns** clear the DNS cache.

```
Command Prompt
Microsoft Windows [Version 10.0.22000.652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\singh>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\singh>
```

Step 3: Enter **nslookup** at the prompt to enter the nslookup interactive mode.

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

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\singh>nslookup
Default Server: UnKnown
Address: 192.168.0.1
```

Step 4: Enter the domain name of a website. The domain name www.cisco.com

```
> www.cisco.com
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
Name: e2867.dsca.akamaiedge.net
Addresses: 2600:1417:75:d9f::b33
           2600:1417:75:d8a::b33
           23.10.37.140
Aliases: www.cisco.com
         www.cisco.com.akadns.net
         wwwds.cisco.com.edgekey.net
         wwwds.cisco.com.edgekey.net.globalredir.akadns.net
```

Step 5: type **exit** in prompt it will exit the nslookup

```
2600:1417:75:d8a::b33
23.10.37.140
Aliases: www.cisco.com
         www.cisco.com.akadns.net
         wwwds.cisco.com.edgekey.net
         wwwds.cisco.com.edgekey.net.globalredir.akadns.net
> exit
```

## Part 2: Explore DNS Query Traffic.

Step 1: Observe the traffic captured in the Wireshark Packet List pane. Enter **udp.port == 53** in the filter box and click the arrow (or press enter) to display only DNS packets.

Select the DNS packet labeled **Standard query 0x0002 A www.cisco.com**. In the Packet Details pane, notice this packet has Ethernet II, Internet Protocol Version 4, User Datagram Protocol and Domain Name System (query).

No.	Time	Source	Destination	Protocol	Length	Info
1011	402.079863	192.168.0.1	192.168.0.109	DNS	133	Standard query response 0x0001 No such name PTR 1.0.168.192.in-addr.arpa SOA loca
1011	407.078114	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0002 A cisco.com
1011	407.083314	192.168.0.1	192.168.0.109	DNS	85	Standard query response 0x0002 A cisco.com A 72.163.4.185
1011	407.087650	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0003 AAAA cisco.com
1011	407.305034	192.168.0.1	192.168.0.109	DNS	97	Standard query response 0x0003 AAAA cisco.com AAAA 2001:420:1101:1::185
1094	446.745262	192.168.0.109	192.168.0.1	DNS	74	Standard query 0xd5a7 A www.google.com
1094	446.753348	192.168.0.1	192.168.0.109	DNS	90	Standard query response 0xd5a7 A www.google.com A 142.250.66.4
1098	544.429530	192.168.0.109	192.168.0.1	DNS	95	Standard query 0x2845 A optimizationguide-pa.googleapis.com
1098	544.440411	192.168.0.1	192.168.0.109	DNS	111	Standard query response 0x2845 A optimizationguide-pa.googleapis.com A 142.250.18
1098	544.510819	192.168.0.109	192.168.0.1	DNS	71	Standard query 0xe965 A i.ytimg.com

> Frame 101125: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF\_{B0FFBDC5-FBAB-485A-AC52-D96FFD6C9E89}, id 0  
 > Ethernet II, Src: Tp-LinkT\_d5:f0:be (d0:37:45:d5:f0:be), Dst: Tp-LinkT\_ab:04:dc (18:a6:f7:ab:04:dc)  
 > Internet Protocol Version 4, Src: 192.168.0.109, Dst: 192.168.0.1  
 > User Datagram Protocol, Src Port: 49750, Dst Port: 53  
 > Domain Name System (query)

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



No.	Time	Source	Destination	Protocol	Length	Info
1011..	402.079863	192.168.0.1	192.168.0.109	DNS	133	Standard query response 0x0001 No such name PTR 1.0.168.192.in-addr.arpa SOA
1011..	407.078114	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0002 A cisco.com
1011..	407.083314	192.168.0.1	192.168.0.109	DNS	85	Standard query response 0x0002 A cisco.com A 72.163.4.185
1011..	407.087650	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0003 AAAA cisco.com
1011..	407.305034	192.168.0.1	192.168.0.109	DNS	97	Standard query response 0x0003 AAAA cisco.com AAAA 2001:420:1101:1::185
1094..	446.745262	192.168.0.109	192.168.0.1	DNS	74	Standard query 0xd5a7 A www.google.com
1094..	446.753348	192.168.0.1	192.168.0.109	DNS	90	Standard query response 0xd5a7 A www.google.com A 142.250.66.4
1098..	544.429530	192.168.0.109	192.168.0.1	DNS	95	Standard query 0x2845 A optimizationguide-pa.googleapis.com
1098..	544.440411	192.168.0.1	192.168.0.109	DNS	111	Standard query response 0x2845 A optimizationguide-pa.googleapis.com A 142.2
1098..	544.510819	192.168.0.109	192.168.0.1	DNS	71	Standard query 0xe965 A i.ytimg.com

>	Frame 101125: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF_{B0FFBDC5-FBAB-485A-AC52-D96FFD6C9E89}, id 0
>	Ethernet II, Src: Tp-LinkT_d5:f0:be (d0:37:45:d5:f0:be), Dst: Tp-LinkT_ab:04:dc (18:a6:f7:ab:04:dc)
>	Destination: Tp-LinkT_ab:04:dc (18:a6:f7:ab:04:dc)
>	Address: Tp-LinkT_ab:04:dc (18:a6:f7:ab:04:dc)
>	.... 0. .... = LG bit: Globally unique address (factory default)
>	.... 0. .... = IG bit: Individual address (unicast)
>	Source: Tp-LinkT_d5:f0:be (d0:37:45:d5:f0:be)
>	Address: Tp-LinkT_d5:f0:be (d0:37:45:d5:f0:be)
>	.... 0. .... = LG bit: Globally unique address (factory default)
>	.... 0. .... = IG bit: Individual address (unicast)
>	Type: IPv4 (0x0800)
>	Internet Protocol Version 4, Src: 192.168.0.109, Dst: 192.168.0.1
>	User Datagram Protocol, Src Port: 49750, Dst Port: 53
>	Domain Name System (query)

What are the source and destination MAC addresses? Which network interfaces are these MAC addresses associated with?

In this example, the source MAC address is associated with the NIC on the PC and the destination MAC address is associated with the default gateway. If there is a local DNS server, the destination MAC address would be the MAC address of the local DNS server.

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

1011..	402.079863	192.168.0.1	192.168.0.109	DNS	133	Standard query response 0x0001 No such name PTR 1.0.168.192.1
1011..	407.078114	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0002 A cisco.com
1011..	407.083314	192.168.0.1	192.168.0.109	DNS	85	Standard query response 0x0002 A cisco.com A 72.163.4.185
1011..	407.087650	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0003 AAAA cisco.com
1011..	407.305034	192.168.0.1	192.168.0.109	DNS	97	Standard query response 0x0003 AAAA cisco.com AAAA 2001:420:1
1094..	446.745262	192.168.0.109	192.168.0.1	DNS	74	Standard query 0xd5a7 A www.google.com
1094..	446.753348	192.168.0.1	192.168.0.109	DNS	90	Standard query response 0xd5a7 A www.google.com A 142.250.66.
1098..	544.429530	192.168.0.109	192.168.0.1	DNS	95	Standard query 0x2845 A optimizationguide-pa.googleapis.com
1098..	544.440411	192.168.0.1	192.168.0.109	DNS	111	Standard query response 0x2845 A optimizationguide-pa.googlea
1098..	544.510819	192.168.0.109	192.168.0.1	DNS	71	Standard query 0xe965 A i.ytimg.com

>	Frame 101125: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF_{B0FFBDC5-FBAB-485A-AC52-D96FFD6C9E89},
>	Ethernet II, Src: Tp-LinkT_d5:f0:be (d0:37:45:d5:f0:be), Dst: Tp-LinkT_ab:04:dc (18:a6:f7:ab:04:dc)
>	Internet Protocol Version 4, Src: 192.168.0.109, Dst: 192.168.0.1
>	0100 .... = Version: 4
>	.... 0101 = Header Length: 20 bytes (5)
>	Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
>	Total Length: 55
>	Identification: 0x4ca3 (19619)
>	Flags: 0x00
>	...0 0000 0000 0000 = Fragment Offset: 0
>	Time to Live: 128
>	Protocol: UDP (17)
>	Header Checksum: 0x6c54 [validation disabled]
>	[Header checksum status: Unverified]
>	Source Address: 192.168.0.109
>	Destination Address: 192.168.0.1
>	User Datagram Protocol, Src Port: 49750, Dst Port: 53
>	Domain Name System (query)

What are the source and destination IP addresses? Which network interfaces are these IP addresses associated with?

In this example, the source IP address is associated with the NIC on the PC and the destination IP address is associated with the DNS server.



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

No.	Time	Source	Destination	Protocol	Length	Info
1011...	402.079863	192.168.0.1	192.168.0.109	DNS	133	Standard query response 0x0001 No such name P
1011...	407.078114	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0002 A cisco.com
1011...	407.083314	192.168.0.1	192.168.0.109	DNS	85	Standard query response 0x0002 A cisco.com A
1011...	407.087650	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0003 AAAA cisco.com
1011...	407.305034	192.168.0.1	192.168.0.109	DNS	97	Standard query response 0x0003 AAAA cisco.com
1094...	446.745262	192.168.0.109	192.168.0.1	DNS	74	Standard query 0xd5a7 A www.google.com
1094...	446.753348	192.168.0.1	192.168.0.109	DNS	90	Standard query response 0xd5a7 A www.google.c
1098...	544.429530	192.168.0.109	192.168.0.1	DNS	95	Standard query 0x2845 A optimizationguide-pa.
1098...	544.440411	192.168.0.1	192.168.0.109	DNS	111	Standard query response 0x2845 A optimization
1098...	544.510819	192.168.0.109	192.168.0.1	DNS	71	Standard query 0xe965 A i.ytimg.com

> Frame 101125: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF_{B0FFBDC5-FBAB-485A-AC52}
> Ethernet II, Src: Tp-Link_T_d5:f0:be (d0:37:45:d5:f0:be), Dst: Tp-Link_T_ab:04:dc (18:a6:f7:ab:04:dc)
> Internet Protocol Version 4, Src: 192.168.0.109, Dst: 192.168.0.1
> User Datagram Protocol, Src Port: 49750, Dst Port: 53
Source Port: 49750
Destination Port: 53
Length: 35
Checksum: 0x7344 [unverified]
[Checksum Status: Unverified]
[Stream index: 184]
> [Timestamps]
UDP payload (27 bytes)
> Domain Name System (query)

What are the source and destination ports? What is the default DNS port number?

The source port number is 58461 and the destination port is 53, which is the default DNS port number.

Open a Command Prompt and enter **arp -a** and **ipconfig /all** to record the MAC and IP addresses of the PC

```
C:\Users\singh>arp -a

Interface: 192.168.56.1 --- 0xd
Internet Address      Physical Address      Type
192.168.56.255        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

Interface: 192.168.0.109 --- 0x11
Internet Address      Physical Address      Type
192.168.0.1           18-a6-f7-ab-04-dc    dynamic
192.168.0.255         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
```

```
Ethernet adapter VirtualBox Host-Only Network:

Connection-specific DNS Suffix . : 
Description . . . . . : VirtualBox Host-Only Ethernet Adapter
Physical Address. . . . . : 0A-00-27-00-00-0D
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::bc65:b322:40e8:7df5%13(Preferred)
IPv4 Address. . . . . : 192.168.56.1(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 
DHCPv6 IAID . . . . . : 671744039
DHCPv6 Client DUID. . . . . : 00-01-00-01-29-F4-0B-77-1C-6F-65-93-DA-5F
NetBIOS over Tcpip. . . . . : Enabled

Wireless LAN adapter Local Area Connection* 1:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : 
Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
Physical Address. . . . . : D2-37-45-D5-F0-BE
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
```

Compare the MAC and IP addresses in the Wireshark results to the results from the **ipconfig /all** results. What is your observation?

The IP and MAC addresses captured in the Wireshark results are the same as the addresses listed in **arp -a** and **ipconfig /all** command.

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

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

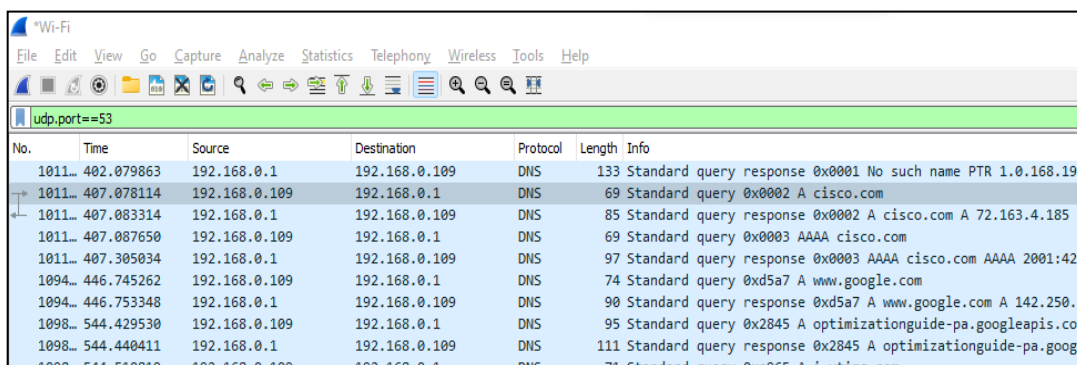
```

Ethernet II, Src: Tp-Link_05:10:0e (00:37:45:05:10:0e), Dst: Tp-Link_ab:04:dc
> Internet Protocol Version 4, Src: 192.168.0.109, Dst: 192.168.0.1
> User Datagram Protocol, Src Port: 49750, Dst Port: 53
✓ 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
    [Response In: 101126]

```

### Part 3: Explore DNS Response Traffic

Step 1: Select the corresponding response DNS packet labeled **Standard query response 0x0002 A www.cisco.com**.



No.	Time	Source	Destination	Protocol	Length	Info
1011	402.079863	192.168.0.1	192.168.0.109	DNS	133	Standard query response 0x0001 No such name PTR 1.0.168.192
1011	407.078114	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0002 A cisco.com
1011	407.083314	192.168.0.1	192.168.0.109	DNS	85	Standard query response 0x0002 A cisco.com A 72.163.4.185
1011	407.087650	192.168.0.109	192.168.0.1	DNS	69	Standard query 0x0003 AAAA cisco.com
1011	407.305034	192.168.0.1	192.168.0.109	DNS	97	Standard query response 0x0003 AAAA cisco.com AAAA 2001:420
1094	446.745262	192.168.0.109	192.168.0.1	DNS	74	Standard query 0xd5a7 A www.google.com
1094	446.753348	192.168.0.1	192.168.0.109	DNS	90	Standard query response 0xd5a7 A www.google.com A 142.250.6
1098	544.429530	192.168.0.109	192.168.0.1	DNS	95	Standard query 0x2845 A optimizationguide-pa.googleapis.com
1098	544.440411	192.168.0.1	192.168.0.109	DNS	111	Standard query response 0x2845 A optimizationguide-pa.google
1098	544.510819	192.168.0.109	192.168.0.1	DNS	71	Standard query 0xe965 A i.vtime.com

Step 2: Expand **Domain Name System (response)**. Then expand the **Flags**, **Queries**, and **Answers**. Observe the results.

1011...	407.078114	192.168.0.109	192.168.0.1	DNS	69 Standard query 0x0002 A cisco.com
1011...	407.083314	192.168.0.1	192.168.0.109	DNS	85 Standard query response 0x0002 A cisco.com
1011...	407.087650	192.168.0.109	192.168.0.1	DNS	69 Standard query 0x0003 AAAA cisco.com
1011...	407.305034	192.168.0.1	192.168.0.109	DNS	97 Standard query response 0x0003 AAAA cisco.com
1094...	446.745262	192.168.0.109	192.168.0.1	DNS	74 Standard query 0xd5a7 A www.google.com
1094...	446.753348	192.168.0.1	192.168.0.109	DNS	90 Standard query response 0xd5a7 A www.google.com
1098...	544.429530	192.168.0.109	192.168.0.1	DNS	95 Standard query 0x2845 A optimization.google.com
1098...	544.440411	192.168.0.1	192.168.0.109	DNS	111 Standard query response 0x2845 A optimization.google.com
1098...	544.510819	192.168.0.109	192.168.0.1	DNS	71 Standard query 0xe965 A i.ytimg.com

```

> Frame 101125: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF_{B0FFBDC5-FBAB-48
> Ethernet II, Src: Tp-LinkT_d5:f0:be (d0:37:45:d5:f0:be), Dst: Tp-LinkT_ab:04:dc (18:a6:f7:ab:04:dc)
> Internet Protocol Version 4, Src: 192.168.0.109, Dst: 192.168.0.1
> User Datagram Protocol, Src Port: 49750, Dst Port: 53
< 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
    [Response In: 101126]

```

## Practical No 5

### Aim: Create your own syslog Server

Step 1: To check whether rsyslog services already running or not use above command

“sudo systemctl status rsyslog”

```
ubuntu@ubuntu2004:~$ sudo systemctl status rsyslog
Unit rsyslog.service could not be found.
```

Step 2: In case not installed or running, install rsyslog using the following commands:

“sudo apt-get update”

“sudo apt-get install rsyslog”

```
ubuntu@ubuntu2004:~$ sudo apt-get install rsyslog
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-doc rsyslog-openssl
  | rsyslog-gnutls rsyslog-gssapi rsyslog-relp
The following NEW packages will be installed:
  rsyslog
0 upgraded, 1 newly installed, 0 to remove and 308 not upgraded.
Need to get 0 B/427 kB of archives.
After this operation, 1,695 kB of additional disk space will be used.
Selecting previously unselected package rsyslog.
(Reading database ... 148664 files and directories currently installed.)
Preparing to unpack .../rsyslog_8.2001.0-1ubuntu1.3_amd64.deb ...
Unpacking rsyslog (8.2001.0-1ubuntu1.3) ...
Setting up rsyslog (8.2001.0-1ubuntu1.3) ...
The user `syslog' is already a member of `adm'.
The user `syslog' is already a member of `tty'.
```

Step 3: Open rsyslog configuration file

“sudo nano /etc/rsyslog.conf”

```
ubuntu@ubuntu2004:~$ sudo nano /etc/rsyslog.conf
```

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

```
module(load="imuxsock") # provides support for local system logging
#module(load="imark") # 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")
```

Step 5: Add template right before GLOBAL DIRECTIVES section.

**\$template remote-incoming-**

**logs,"/var/log/%HOSTNAME%/%PROGRAMNAME%.log"**

**\*.\* ?remote-incoming-logs**

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

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

#####
#### GLOBAL DIRECTIVES ####
#####
```

Step 6: Save and restart rsyslog service:

**“sudo systemctl restart rsyslog”**

```
ubuntu@ubuntu2004:~$ sudo systemctl restart rsyslog
```

Step 7: Confirme that rsyslog service is listening on configured ports

**“ss -tunelp | grep 514”**

```
ubuntu@ubuntu2004:~$ ss -tunelp | grep 514
udp      UNCONN    0      0      0.0.0.0:514      0.0.0.0:*
   ino:86633 sk:4 <->
udp      UNCONN    0      0      [::]:514      [::]:*
   ino:86634 sk:8 v6only:1 <->
tcp      LISTEN     0      25      0.0.0.0:514      0.0.0.0:*
   ino:86637 sk:9 <->
tcp      LISTEN     0      25      [::]:514      [::]:*
   ino:86638 sk:d v6only:1 <->
```

Step 8: Allow rsyslog firewall port rules

**“sudo ufw allow 514/tcp” “sudo**

**ufw allow 514/udp”**

```
ubuntu@ubuntu2004:~$ sudo ufw allow 514/tcp
Rules updated
Rules updated (v6)
ubuntu@ubuntu2004:~$ sudo ufw allow 514/udp
Skipping adding existing rule
Skipping adding existing rule (v6)
```

Step 9: To verify configuration, run the following command:

**“sudo rsyslogd -N1 -f /etc/rsyslog.conf**

```
ubuntu@ubuntu2004:~$ sudo rsyslogd -N1 -f /etc/rsyslog.conf
rsyslogd: version 8.2001.0, config validation run (level 1), master config /etc/
rsyslog.conf
rsyslogd: End of config validation run. Bye.
```



## Practical No.6

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

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

Step 2: Open kali linux and install rsyslog using the following commands

**“sudo apt-get update”**

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

**“sudo apt-get install rsyslog”**

```
(kali@kali)-[~]
$ sudo apt-get install rsyslog
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libestr0 libfastjson4 liblognorm5
Suggested packages:
  rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-doc rsyslog-openssl | rsyslog-gnutls rsyslog-gssapi rs
The following NEW packages will be installed:
  libestr0 libfastjson4 liblognorm5 rsyslog
0 upgraded, 4 newly installed, 0 to remove and 781 not upgraded.
Need to get 829 kB of archives.
After this operation, 2,280 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kali.download/kali kali-rolling/main amd64 libestr0 amd64 0.1.11-1 [9,204 B]
Get:2 http://kali.download/kali kali-rolling/main amd64 libfastjson4 amd64 1.2304.0-1 [28.9 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 liblognorm5 amd64 2.0.6-4 [67.2 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 rsyslog amd64 8.2302.0-1 [723 kB]
Fetched 829 kB in 23s (36.1 kB/s)
Selecting previously unselected package libestr0:amd64.
(Reading database ... 302802 files and directories currently installed.)
```

Step 3: Open rsyslog configuration file

**“sudo nano /etc/rsyslog.conf”**

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

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

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

# provides TCP syslog reception
```

Step 4: Add above lines at the end of the file

**@192.168.137.50:514**

**\*.\* @192.168.137.50:514**

```

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:*
@192.168.137.50:514
*.*@192.168.137.50:514

```

Note: You can enable to send logs over UDP. For TCP use @@ , instead of one

Step 5: For the end add these following variables in case when the rsyslog server goes down.

```

$ActionQueueFileName queue
$ActionQueueMaxDiskSpace 1g
$ActionQueueSaveOnShutdown on
$ActionQueueType LinkedList
$ActionResumeRetryCount -1

```

```

*.emerg                                :omusrmsg:*
@192.168.137.50:514
*.*@192.168.137.50:514

@ActionQueueFileName queue
@ActionQueueMaxDiskSpace 1g
@ActionQueueSaveOnShutdown on
@ActionQueueType LinkedList
@ActionResumeRetryCount -1

```

Step 6: Then Save and exit the file

Step 7: restart the rsyslog service  
**“sudo systemctl restart rsyslog”**

```

(kali@kali)-[~]
$ sudo systemctl restart rsyslog

```

### Verify the logs

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

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

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

Step 9: To check logs use the following command: Let's for example inspect rsyslogd.log.

**“sudo tail -f /var/log/kali/rsyslogd.log”**

```

ubuntu@ubuntu2004:~$ sudo tail -f /var/log/kali/rsyslogd.log
2022-05-18T05:47:20-04:00 kali rsyslogd: [origin software="rsyslogd" swVersion="
8.2204.0" x-pid="8621" x-info="https://www.rsyslog.com"] start
2022-05-18T05:47:20-04:00 kali rsyslogd: [origin software="rsyslogd" swVersion="
8.2204.0" x-pid="4842" x-info="https://www.rsyslog.com"] exiting on signal 15.
2022-05-18T05:47:20-04:00 kali rsyslogd: imuxsock: Acquired UNIX socket '/run/sy
stemd/journal/syslog' (fd 3) from systemd. [v8.2204.0]
2022-05-18T05:47:20-04:00 kali rsyslogd: [origin software="rsyslogd" swVersion="
8.2204.0" x-pid="8621" x-info="https://www.rsyslog.com"] start

```

## Practical No.7

### Aim: Install and Run Splunk on Linux.

Step1: Download Splunk Installer

“cd /tmp && wget

<https://download.splunk.com/products/splunk/releases/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb>”

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

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

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

Step 2: Install Splunk

“sudo dpkg -i splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb”

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

Step 3: Enable the Splunk to start at boot

Press enter key till you reach to the end of the agreement, then you have to accept the license agreement by typing “y”.

Then you have to enter the initial admin password and use this password to access the web portal

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

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



```

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

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

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

```

Step 4: Start the Splunk service

“sudo service splunk start”

```

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

```

Step 5: Check splunk service Status

“sudo service splunk status”

```

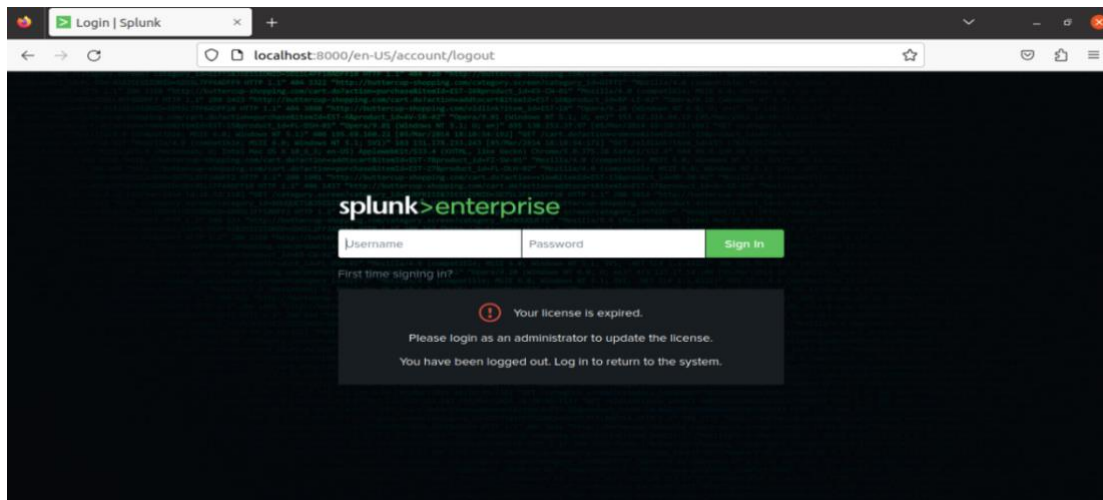
● splunk.service - LSB: Start splunk
   Loaded: loaded (/etc/init.d/splunk; generated)
   Active: active (running) since Sun 2023-06-25 08:46:44 PDT; 46s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 4025 ExecStart=/etc/init.d/splunk start (code=exited, status=0>
    Tasks: 166 (limit: 2214)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

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

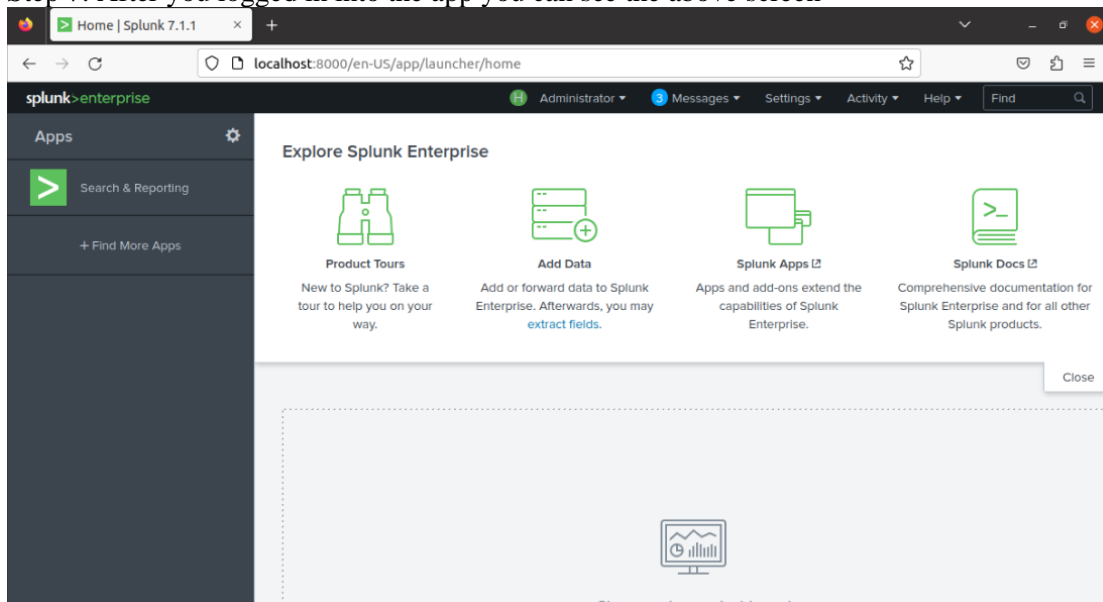
```

Step 6: Splunk will be started at port 8000. You can access the application via URL

“http://localhost:8000/“. To logged in into the app enter username as “admin” then enter your password. In my case the password is “admin!123”



Step 7: After you logged in into the app you can see the above screen



## Practical No.8

**Aim:** Install and Configure ELK on Linux.

### **Part 1: Installing java**

Step 1: write the below command and update and install the jdk

**“sudo apt update”**

```
ubuntu@ubuntu:~/Desktop$ sudo apt update
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [111 kB]
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable/main i386 Packages [81.3 kB]
Hit:5 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:7 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 206 kB in 3s (63.5 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
144 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

**Install java**

**“sudo apt install default-jre”**

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

Step 2: check the java version by this command “java -version”

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

### **Part 2: Install and Configure the Elasticsearch**

#### **Elastic Search**

Elasticsearch store logs coming from external sources and offers real-time distributed search and analytics with the RESTful web interface.

Step 1: Download and install the GPG signing key.

**“curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -”**

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

Step 2: Set up the Elasticsearch repository on your system by running the below command.

**“echo “deb https://artifacts.elastic.co/packages/7.x/apt stable main” | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list”**

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

Step 3: Update the repository cache and then install the Elasticsearch package.

**“sudo apt update”**

```
ubuntu@ubuntu:~/Desktop$ sudo apt update
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [111 kB]
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable/main i386 Packages [81.3 kB]
Hit:5 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:7 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 206 kB in 3s (63.5 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
144 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

**“sudo apt install elasticsearch”**

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

Step 4: Edit the Elasticsearch configuration file to set the cluster name for Graylog set up.

**“sudo nano /etc/elasticsearch/elasticsearch.yml”**

**Uncomment**

**network.host:localhost**

**http.port:9200**

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

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

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

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

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

Step 7: You will then test whether your Elasticsearch service is running. Do it by sending an HTTP request:

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

```
ubuntu@ubuntu:~/Desktop$ curl -X GET "localhost:9200"  
{  
  "name" : "ubuntu",  
  "cluster_name" : "elasticsearch",  
  "cluster_uuid" : "dnVrXuFqQlUC0sVRLKxZ3w",  
  "version" : {  
    "number" : "7.17.10",  
    "build_flavor" : "default",  
    "build_type" : "deb",  
    "build_hash" : "fec68e3150eda0c307ab9a9d7557f5d5fd71349",  
    "build_date" : "2023-04-23T05:33:18.138275597Z",  
    "build_snapshot" : false,  
    "lucene_version" : "8.11.1",  
    "minimum_wire_compatibility_version" : "6.8.0",  
    "minimum_index_compatibility_version" : "6.0.0-beta1"  
  },  
  "tagline" : "You Know, for Search"  
}
```



## Practical No: 9

### Aim: Install and Configure GrayLog on Linux

#### Part 1: Install Java and Els

Step 1: Install Java and Els (Practical 8)

Step 2: Edit the Elasticsearch configuration file to set the cluster name for Graylog set up.

“sudo nano /etc/elasticsearch/elasticsearch.yml”

```
#
# Use a descriptive name for your cluster:
#
cluster.name: graylog
#
# ----- Node -----
```

Step 3: Set the cluster name as graylog, as shown below. Then, uncomment the line and below add this line “**action.auto\_create\_index: false**” then save.

```
# ----- Various -----
#
# Require explicit names when deleting indices:
#
#action.destructive_requires_name: true
action.auto_create_index: false
# ----- Security -----
```

Step 4: Start the Elasticsearch service to read the new configurations.

“sudo systemctl daemon-reload”

“sudo systemctl start elasticsearch”

“sudo systemctl enable elasticsearch”

```
ubuntu@ubuntu:~/Desktop$ sudo nano /etc/elasticsearch/elasticsearch.yml
ubuntu@ubuntu:~/Desktop$ sudo systemctl daemon-reload
ubuntu@ubuntu:~/Desktop$ sudo systemctl start elasticsearch
ubuntu@ubuntu:~/Desktop$ sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/
systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
ubuntu@ubuntu:~/Desktop$
```

Step 5: Elastic search should be now listening on port 9200. Use the curl command to check the Elasticsearch’s response

“curl -X GET http://localhost:9200 ”

```
ubuntu@ubuntu:~/Desktop$ curl -X GET http://localhost:9200
{
  "name" : "ubuntu",
  "cluster_name" : "graylog",
  "cluster_uuid" : "dnVrXuFqQluC0sVRLKxZ3w",
  "version" : {
    "number" : "7.17.10",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "fec68e3150eda0c307ab9a9d7557f5d5fd71349",
    "build_date" : "2023-04-23T05:33:18.138275597Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

## Part 2: Install MongoDB

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

### Step 1: "sudo apt update"

```
ubuntu@ubuntu:~/Desktop$ sudo apt update
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Fetched 222 kB in 4s (51.6 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
144 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

### "sudo apt install -y mongodb-server"

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

Step 2: Start the MongoDB and enable it on the system start-up.

### "sudo systemctl start mongodb"

### "sudo systemctl enable mongodb"

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

## Part 4: Install GrayLog Server

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

Step 1: Download and install the Graylog 3.3 repository configuration package.

### "wget https://packages.graylog2.org/repo/packages/graylog-4.2-repository\_latest.deb"

```
ubuntu@ubuntu:~/Desktop$ wget https://packages.graylog2.org/repo/packages/graylog-4.2-repository_latest.deb
--2023-06-25 23:16:39-- https://packages.graylog2.org/repo/packages/graylog-4.2-repository_latest.deb
Resolving packages.graylog2.org (packages.graylog2.org)... 104.21.88.209, 172.67.153.95, 2606:4700:3035::ac43:995f, ...
Connecting to packages.graylog2.org (packages.graylog2.org)|104.21.88.209|:443..
. connected.
HTTP request sent, awaiting response... 302 Found
Location: https://graylog-package-repository.s3.eu-west-1.amazonaws.com/packages/graylog-4.2-repository_latest.deb?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20230626T061641Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credential=AKIAIJSI6MCSXPXFVDPIA%2F20230626%2Feu-west-1%2Fs3%2Faws4_request&X-Amz-Signature=6c219605c118e6563deae52860456cd63adfb6295aa4d156e78a968ac1d02ed0 [following]
--2023-06-25 23:16:41-- https://graylog-package-repository.s3.eu-west-1.amazona
```

**“sudo dpkg -i graylog-3.3-repository\_latest.deb”**

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

**Step 2: Update the repository cache. “sudo apt update”**

```
ubuntu@ubuntu:~/Desktop$ sudo apt update
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [59.9 kB]
Get:3 https://packages.graylog2.org/repo/debian stable InRelease [58.1 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [95.5 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [940 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [275 kB]
```

**Step 3: Install the Graylog server using the following command.****“sudo apt install -y graylog-server”**

```
ubuntu@ubuntu:~/Desktop$ sudo apt install -y graylog-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  graylog-server
0 upgraded, 1 newly installed, 0 to remove and 144 not upgraded.
Need to get 197 MB of archives.
After this operation, 218 MB of additional disk space will be used.
Ign:1 https://packages.graylog2.org/repo/debian stable/4.2 amd64 graylog-server
all 4.2.13-1
Get:1 https://packages.graylog2.org/repo/debian stable/4.2 amd64 graylog-server
all 4.2.13-1 [197 MB]
Fetched 15.6 MB in 42s (373 kB/s)
Selecting previously unselected package graylog-server.
(Reading database ... 191189 files and directories currently installed.)
Preparing to unpack .../graylog-server_4.2.13-1_all.deb ...
Unpacking graylog-server (4.2.13-1) ...
Setting up graylog-server (4.2.13-1) ...
#####
```

**Step 4: You must set a secret to secure the user passwords. Use the pwgen command to generate the secret.****“pwgen -N 1 -s 96”**

```
ubuntu@ubuntu:~/Desktop$ sudo apt install pwgen
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  pwgen
0 upgraded, 1 newly installed, 0 to remove and 144 not upgraded.
Need to get 18.1 kB of archives.
After this operation, 52.2 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 pwgen amd64 2.08-2 [18.1 kB]
Fetched 18.1 kB in 1s (16.9 kB/s)
Selecting previously unselected package pwgen.
(Reading database ... 191212 files and directories currently installed.)
```



```
ubuntu@ubuntu:~/Desktop$ pwgen -N 1 -s 96
dHhrek7amsHYKJ4l0IKuJC6w0PbVZ0nCY7Ea4fPBTzQT5xWmrSpnvHY6Q1ePeBVF58R2mNEH18RDRqDi
W4DxLvL4xb38D0e8
```

Step 5: **sudo gedit /etc/graylog/server/server.conf** edit the conf file and put

Then, place the secret like below.

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

```
# You MUST set a secret to secure/pepper the stored user passwords here. Use at>
# Generate one by using for example: pwgen -N 1 -s 96
# ATTENTION: This value must be the same on all Graylog nodes in the cluster.
# Changing this value after installation will render all user sessions and encr>
password_secret = dHhrek7amsHYKJ4l0IKuJC6w0PbVZ0nCY7Ea4fPBTzQT5xWmrSpnvHY6Q1ePe>
```

Step 6: Now, generate a hash (sha256) password for the root user (not to be confused with the system user, the root user of graylog is admin).

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

Replace password with the choice of your password. Put this command in terminal

**"echo -n password | sha256sum"**

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

Step 7: Edit the server.conf file again.in terminal

**"sudo nano /etc/graylog/server/server.conf"**

```
# You MUST set a secret to secure/pepper the stored user passwords here. Use at>
# Generate one by using for example: pwgen -N 1 -s 96
# ATTENTION: This value must be the same on all Graylog nodes in the cluster.
# Changing this value after installation will render all user sessions and encr>
password_secret = dHhrek7amsHYKJ4l0IKuJC6w0PbVZ0nCY7Ea4fPBTzQT5xWmrSpnvHY6Q1ePe>
```

## Part 5: Setup Graylog web interface

From version Graylog 2.x, the web interface is being served directly by the Graylog server. Step 1: Enable the Graylog web interface by editing the server.conf file.

**"sudo gedit /etc/graylog/server/server.conf"**

Put **http\_bind\_address = 192.168.0.10:9000**

**http\_external\_uri = http://public\_ip:9000/**

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

#### HTTP publish URI
#
# The HTTP URI of this Graylog node which is used to communicate with the other
# nodes using the Graylog web interface
```

Step 2: Start and enable the Graylog service.

Place the below command

**"sudo systemctl daemon-reload"**

**"sudo systemctl start graylog-server"**

**"sudo systemctl enable graylog-server"**

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

**Step 3:** Keep looking Graylog server startup logs. This log will be useful for you to troubleshoot Graylog in case of any issues.

**“sudo tail -f /var/log/graylog-server/server.log”**

**Step 4:** On the successful start of the Graylog server, you should get the following message in the log file.

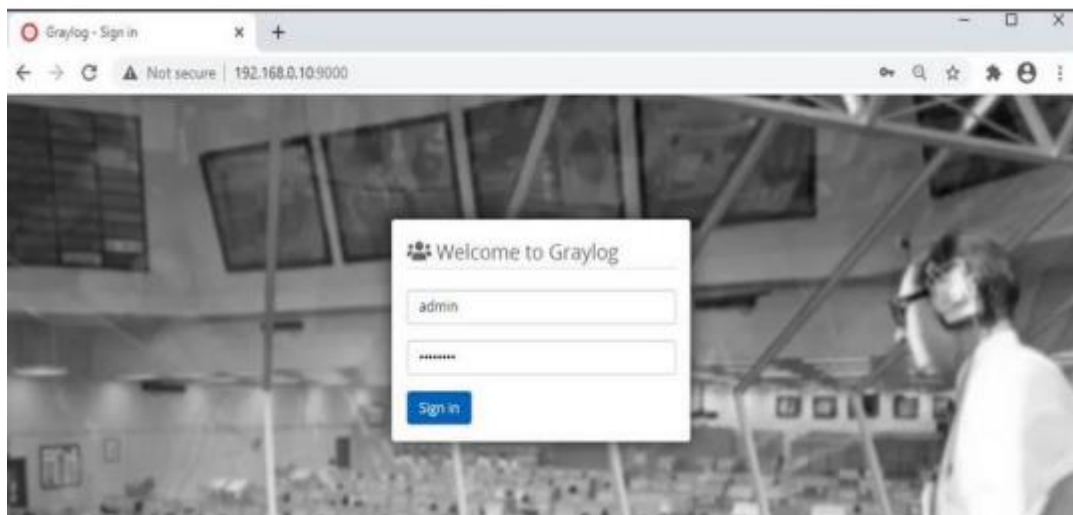
You will be able to see the log file.

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

### Access Graylog

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

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



## Practical No.10

**Aim: Demonstrate Conversion of Data into a Universal Format.**

### **Part 1: Normalize Timestamps in a Log Files.**

Step 1: Launch the CyberOps Workstation VM.

Step 2: open terminal and type “cd /home/analyst/lab.support.files/”

Then type “ls -l”

```
[analyst@secOps lab.support.files]$ ls -l
total 588
-rw-r--r-- 1 analyst analyst 649 Mar 21 2018 apache_in_epoch.log
-rw-r--r-- 1 analyst analyst 126 Mar 21 2018 applicationX_in_epoch.log
drwxr-xr-x 4 analyst analyst 4096 Mar 21 2018 attack_scripts
-rw-r--r-- 1 analyst analyst 102 Mar 21 2018 confidential.txt
-rw-r--r-- 1 analyst analyst 2871 Mar 21 2018 cyops.mn
-rw-r--r-- 1 analyst analyst 255 Jun 25 00:34 decrypted_letter.txt
-rw-r--r-- 1 analyst analyst 75 Mar 21 2018 elk_services
-rw-r--r-- 1 analyst analyst 373 Mar 21 2018 h2_dropbear.banner
```

Step 3: Issue the following AWK command to convert and print the result on the terminal:

Write the command

**“awk 'BEGIN {FS=OFS="|"} {\$3=strftime("%c",\$3)} {print}' applicationX\_in\_epoch.log”**

```
[analyst@secOps lab.support.files]$ awk 'BEGIN {FS=OFS="|"} {$3=strftime("%c",$3)} {print}' applicationX_in_epoch.log
2|Z|Mon 18 Aug 2008 11:00:00 AM EDT|AF|0
3|N|Tue 19 Aug 2008 11:00:00 AM EDT|AF|89
4|N|Sun 07 Sep 2008 11:00:00 AM EDT|AS|12
1|Z|Mon 08 Sep 2008 11:00:00 AM EDT|AS|67
5|N|Tue 09 Sep 2008 11:00:00 AM EDT|EU|23
6|R|Wed 10 Sep 2008 11:00:00 AM EDT|OC|89
|Wed 31 Dec 1969 07:00:00 PM EST
```

The command above is an AWK script. It may seem complicated. The main structure of the AWK script above is as follows:

- **awk** – This invokes the AWK interpreter.
- **„BEGIN** – This defines the beginning of the script.
- **{ }** – This defines actions to be taken in each line of the input text file. An AWK script can have several actions.
- **FS = OFS = “|”** – This defines the field separator (i.e., delimiter) as the bar (|) symbol. Different text files may use different delimiting characters to separate fields. This operator allows the user to define what character is used as the field separator in the current text file.
- **\$3** – This refers to the value in the third column of the current line. In the applicationX\_in\_epoch.log, the third column contains the timestamp in epoch to be converted.
- **strftime** – This is an AWK internal function designed to work with time. The %c and \$3 in between parenthesis are the parameters passed to strftime.
- **applicationX\_in\_epoch.log** – This is the input text file to be loaded and used.

Because you are already in the lab.support.files directory, you do not need to add path information, /home/analyst/lab.support.files/applicationX\_in\_epoch.log.

Step 4: Use nano (or your favorite text editor) to remove the extra empty line at the end of the file  
[analyst@secOps lab.support.files]\$ nano applicationX\_in\_epoch.log

```
[analyst@secOps lab.support.files]$ nano applicationX_in_epoch.log
[analyst@secOps lab.support.files]$ cat applicationX_in_epoch.log
2|Z|1219071600|AF|0
3|N|1219158000|AF|89
4|N|1220799600|AS|12
1|Z|1220886000|AS|67
5|N|1220972400|EU|23
6|R|1221058800|OC|89
```

Step 5: While printing the result on the screen is useful for troubleshooting the script, analysts will likely need to save the output in a text file. Redirect the output of the script above to a file named applicationX\_in\_human.log to save it to a file:

```
[analyst@secOps lab.support.files]$ awk 'BEGIN {FS=OFS="|"} {$3=strftime("%c",$3)} {print}' applicationX_in_epoch.log > applicationX_in_human.log
```

Use cat to view the **applicationX\_in\_human.log**. Notice that the extra line is now removed and the timestamps for the log entries have been converted to human readable format.

```
[analyst@secOps lab.support.files]$ cat applicationX_in_human.log
```

```
[analyst@secOps lab.support.files]$ awk 'BEGIN {FS=OFS="|"} {$3=strftime("%c",$3)} {print}' applicationX_in_epoch.log > applicationX_in_human.log
[analyst@secOps lab.support.files]$ cat applicationX_in_human.log
2|Z|Mon 18 Aug 2008 11:00:00 AM EDT|AF|0
3|N|Tue 19 Aug 2008 11:00:00 AM EDT|AF|89
4|N|Sun 07 Sep 2008 11:00:00 AM EDT|AS|12
1|Z|Mon 08 Sep 2008 11:00:00 AM EDT|AS|67
5|N|Tue 09 Sep 2008 11:00:00 AM EDT|EU|23
6|R|Wed 10 Sep 2008 11:00:00 AM EDT|OC|89
```

## Part 2: Normalize Timestamps in an Apache Log File

Similar to what was done with the applicationX\_in\_epoch.log file, Apache web server log files can also be normalized.

Step 1: Open the terminal and type cat apache\_in\_epoch.log.

```
[analyst@secOps lab.support.files]$ cat apache_in_epoch.log
```

```
[analyst@secOps ~]$ cd /home/analyst/lab.support.files/
[analyst@secOps lab.support.files]$ ls -l
total 592
-rw-r--r-- 1 analyst analyst 649 Mar 21 2018 apache_in_epoch.log
-rw-r--r-- 1 analyst analyst 125 Jun 25 03:32 applicationX_in_epoch.log
-rw-r--r-- 1 analyst analyst 251 Jun 25 03:46 applicationX_in_human.log
drwxr-xr-x 4 analyst analyst 4096 Mar 21 2018 attack_scripts
-rw-r--r-- 1 analyst analyst 102 Mar 21 2018 confidential.txt
-rw-r--r-- 1 analyst analyst 2871 Mar 21 2018 cyops.mn
-rw-r--r-- 1 analyst analyst 255 Jun 25 00:34 decrypted_letter.txt
-rw-r--r-- 1 analyst analyst 75 Mar 21 2018 elk_services
-rw-r--r-- 1 analyst analyst 373 Mar 21 2018 h2_dropbear.banner
drwxr-xr-x 2 analyst analyst 4096 Apr 2 2018 instructor
[analyst@secOps lab.support.files]$ cat apache_in_epoch.log
198.51.100.213 - - [1219071600] "GET /twiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP/1.1" 401 12846
198.51.100.213 - - [1219158000] "GET /twiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1" 200 4523
198.51.100.213 - - [1220799600] "GET /mailman/listinfo/hsdivision HTTP/1.1" 200 6291
198.51.100.213 - - [1220886000] "GET /twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
198.51.100.213 - - [1220972400] "GET /twiki/bin/view/Main/DCCAndPostFix HTTP/1.1" 200 5253
198.51.100.213 - - [1221058800] "GET /twiki/bin/oops/TWiki/AppendixFileSystem?template=oopsmore&m1=1.12&m2=1.12 HTTP/1.1" 200 11382
```

Step 2: In the CyberOps Workstation VM terminal, a copy of the Apache log file, apache\_in\_epoch.log, is stored in the /home/analyst/lab.support.files.

Step 3: type this command in the terminal to see the log in human readable.

```
[analyst@secOps lab.support.files]$ awk 'BEGIN {FS=OFS=" "} {$4=strftime("%c",$4)} {print}' apache_in_epoch.log
```

```
[analyst@secOps lab.support.files]$ awk 'BEGIN {FS=OFS=" "} {$4=strftime("%c",$4)} {print}' apache_in_epoch.log
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP/1.1" 401 12846
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1" 200 4523
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /mailman/listinfo/hsdivision HTTP/1.1" 200 6291
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/view/Main/DCCAndPostFix HTTP/1.1" 200 5253
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/oops/TWiki/AppendixFileSystem?template=oopsmore&m1=1.12&m2=1.12 HTTP/1.1" 200 11382
```

Step 4: Before moving forward, think about the output of the script.

Can you guess what caused the incorrect output? Is the script incorrect? What are the relevant differences between the **applicationX\_in\_epoch.log** and **apache\_in\_epoch.log**?

The problem is the square brackets in the course file. The script expects the timestamp to be in the Unix Epoch format which does not include the square brackets. Because the script does not know what number represents the "[" character, it assumes zero and returns the Unix beginning of time in UTC - 5.

Step 5: To fix the problem, the square brackets must be removed from the timestamp field before the conversion takes place. Adjust the script by adding two actions before the conversion. As shown,

```
[analyst@secOps lab.support.files]$ awk 'BEGIN {FS=OFS=" "} {gsub(/[[]/, "", $4)} {print} {$4=strftime("%c",$4)} {print}' apache_in_epoch.log
```

```
[analyst@secOps lab.support.files]$ awk 'BEGIN {FS=OFS=" "} {gsub(/[[]/, "", $4)} {print} {$4=strftime("%c",$4)} {print}' apache_in_epoch.log
198.51.100.213 - - [1219071600] "GET /twiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP/1.1" 401 12846
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP/1.1" 401 12846
198.51.100.213 - - [1219158000] "GET /twiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1" 200 4523
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1" 200 4523
198.51.100.213 - - [1220799600] "GET /mailman/listinfo/hsdivision HTTP/1.1" 200 6291
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /mailman/listinfo/hsdivision HTTP/1.1" 200 6291
198.51.100.213 - - [1220886000] "GET /twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
198.51.100.213 - - [1220972400] "GET /twiki/bin/view/Main/DCCAndPostFix HTTP/1.1" 200 5253
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/view/Main/DCCAndPostFix HTTP/1.1" 200 5253
198.51.100.213 - - [1221058800] "GET /twiki/bin/oops/TWiki/AppendixFileSystem?template=oopsmore&m1=1.12&m2=1.12 HTTP/1.1" 200 11382
```