**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Topic** | **Date** | **Sign** |
| **1** | **Encrypting and Decrypting Data Using Open SSL** |  |  |
| **2** | **Demonstrate the use of Snort and Firewall Rules** |  |  |
| **3** | **Demonstrate Extract an Executable from a PCAP** |  |  |
| **4** | **Demonstrate Analysis of DNS Traffic** |  |  |
| **5** | **Create your own syslog Server** |  |  |
| **6** | **Configure your Linux system to send syslog messages to a syslog server and**  **Read them** |  |  |
| **7** | **Install and Run Splunk on Linux** |  |  |
| **8** | **Install and Configure ELK on Linux** |  |  |
| **9** | **Install and Configure Gray Log on Linux** |  |  |
| **10** | **Demonstrate Conversion of Data into a Universal Format.** |  |  |

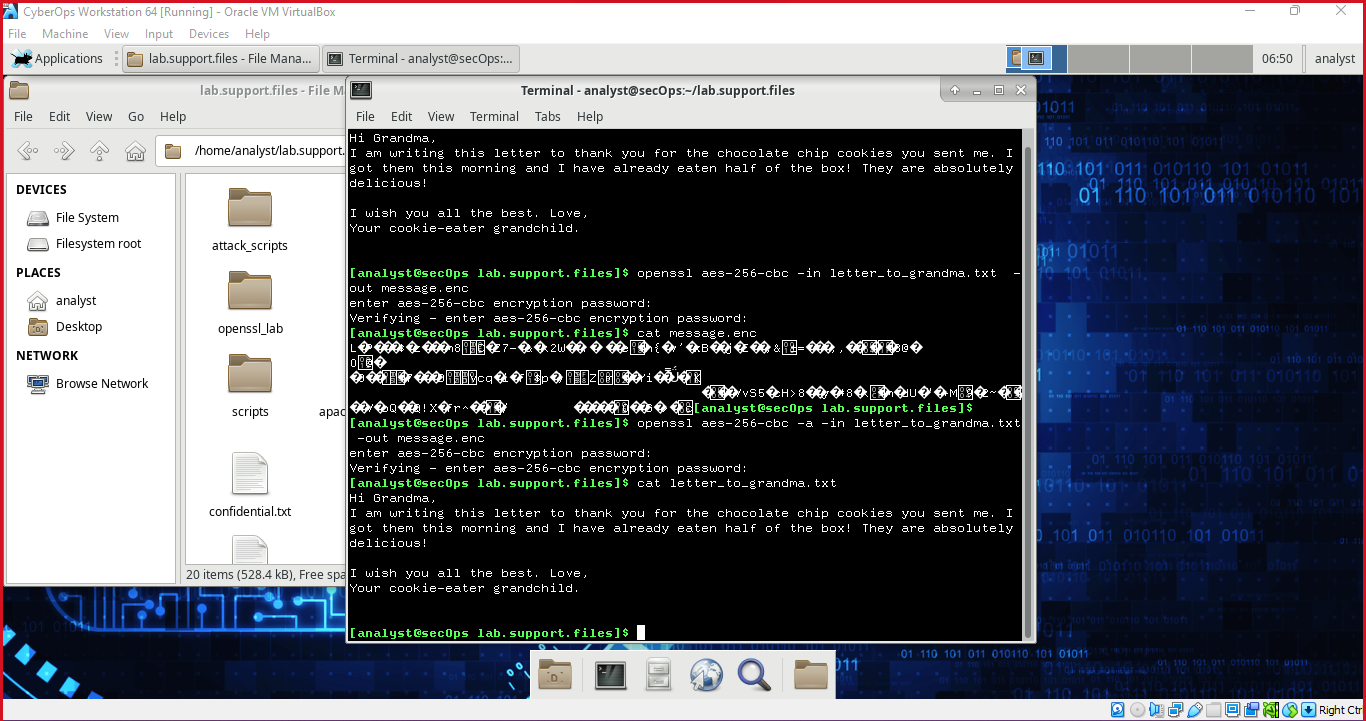
**Practical No 1**

**Solution:**

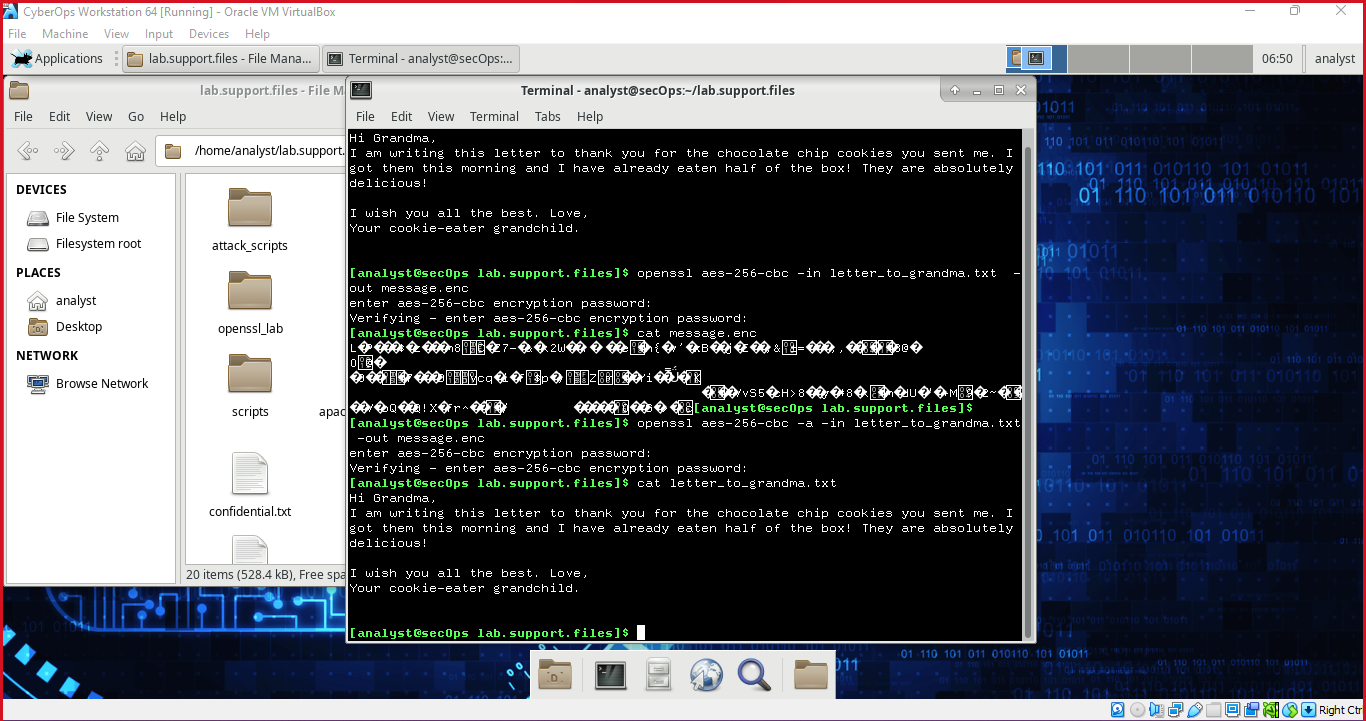
Step 1: cd /home/analyst/lab.support.files/

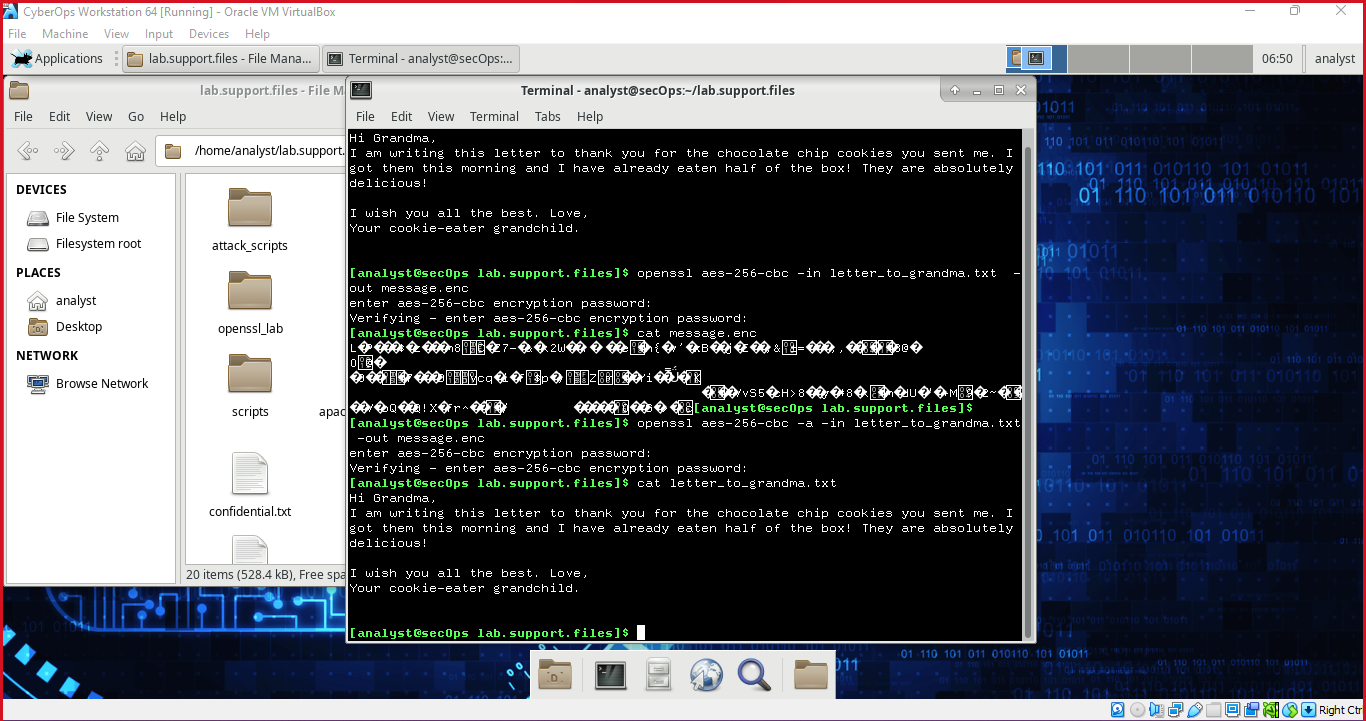
Step 2: open terminal from that directory.

Step 3: cat letter\_to\_grandma.txt

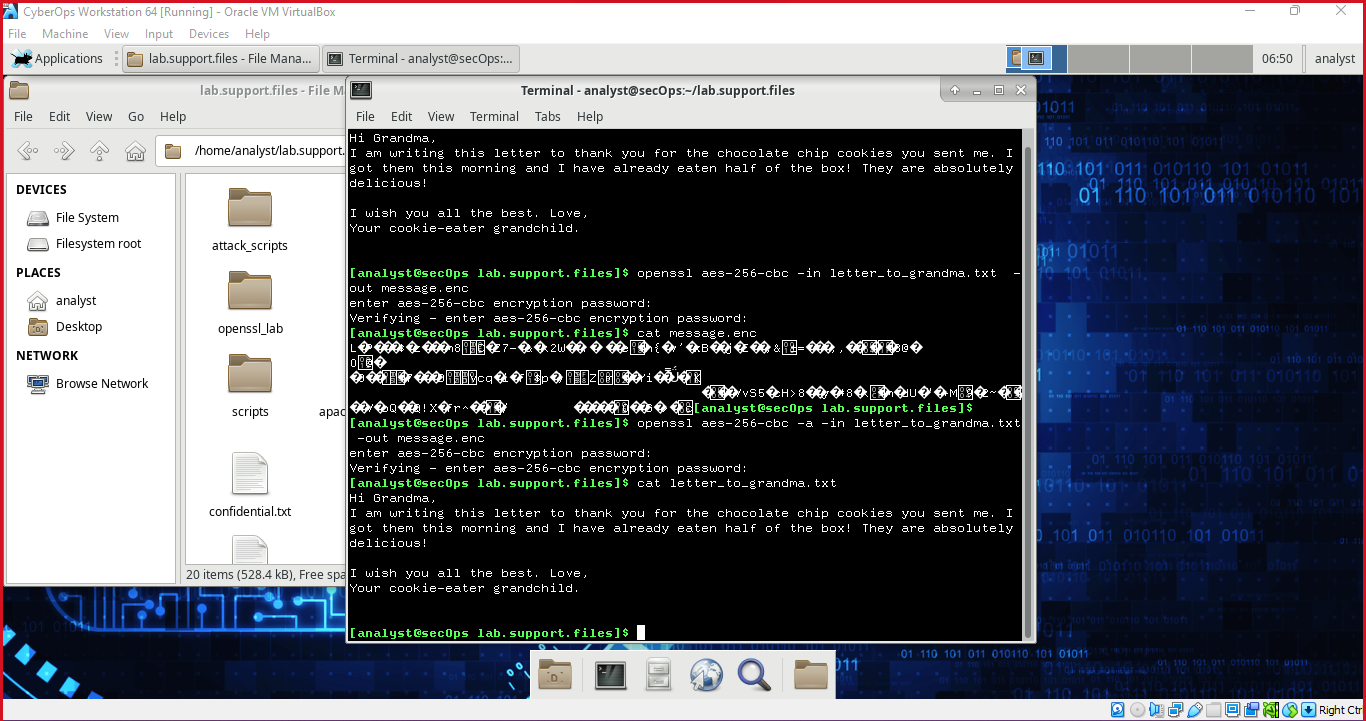


Step 4: openssl aes-256-cbc -in letter\_to\_grandma.txt -out message.enc

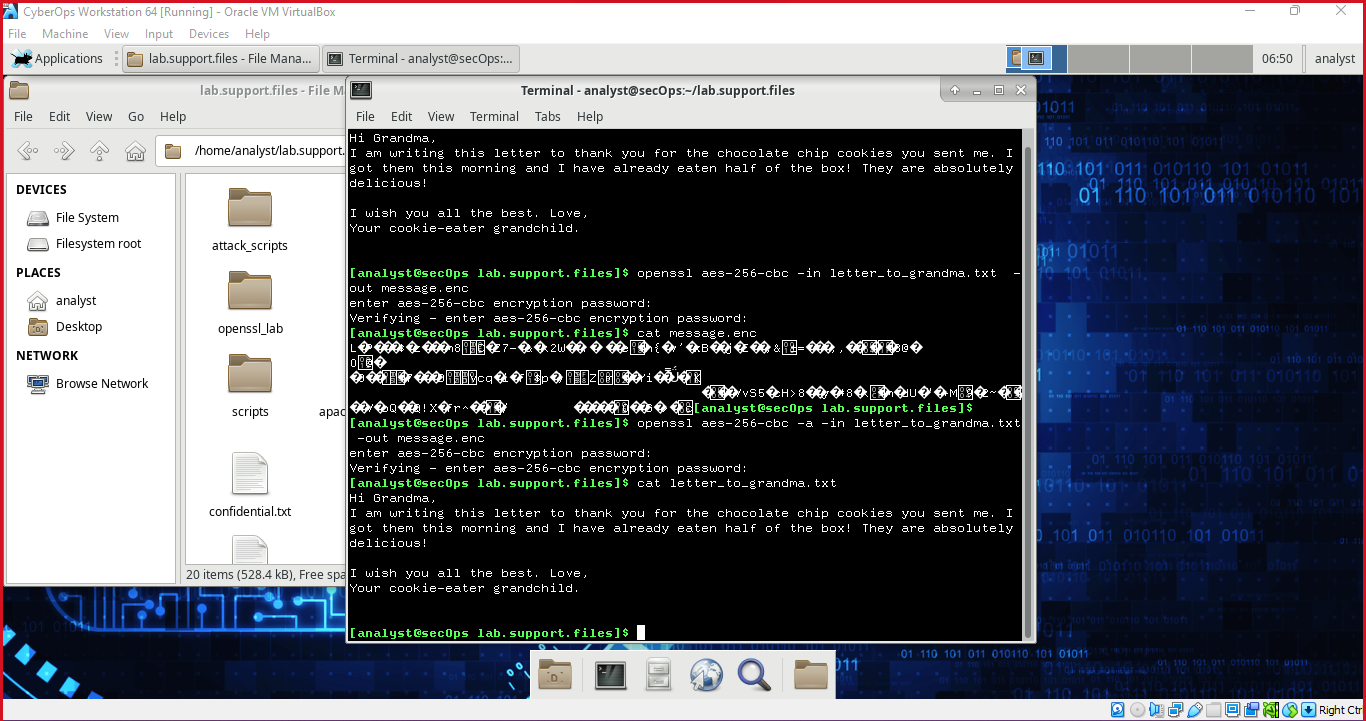


Step 5: cat message.enc

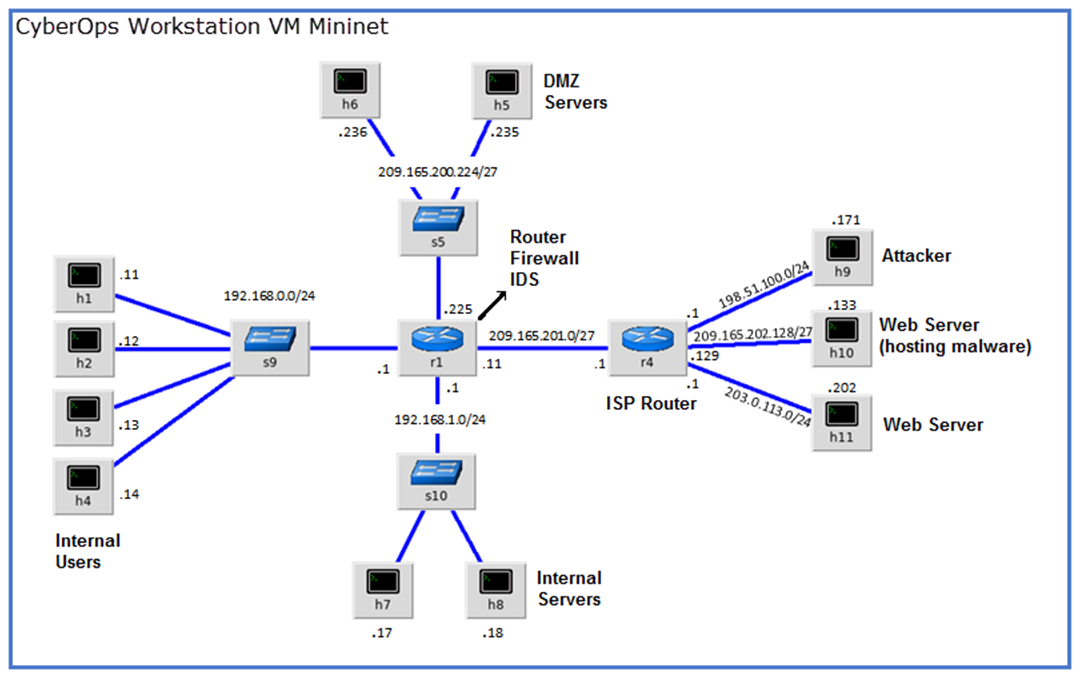
Step 6: openssl aes-256-cbc -a -in letter\_to\_grandma.txt -out message.enc



Step 7: cat letter\_to\_grandma.txt



**Practical No 2**



**Objectives**

**Part 1: Preparing the Virtual Environment**

**Part 2: Firewall and IDS Logs**

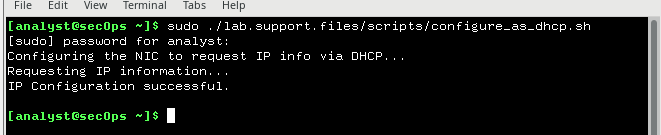
**Part 3: Terminate and Clear Mininet Process**

**Solution:**

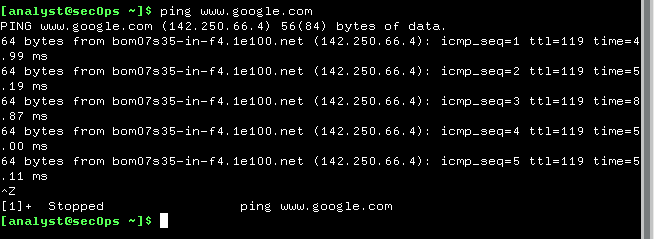
**Part 1: Preparing the Virtual Environment**

Step 1: Launch the **CyberOps Workstation VM**, open a terminal and type

“**sudo ./lab.support.files/scripts/configure\_as\_dhcp.sh**”



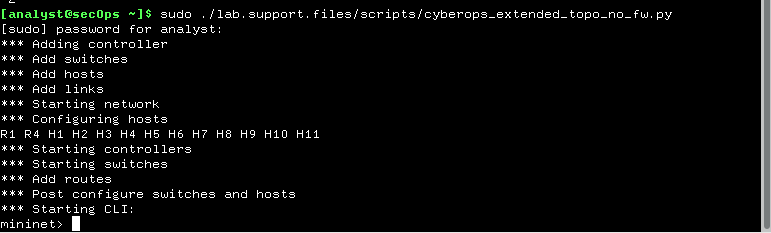
Step 2: Use the **ifconfig** command to verify that your Internet is working and type ping command “**ping** [**www.google.com**](http://www.google.com)**”**



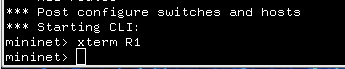
**Part 2: Firewall and IDS Logs.**

Step 1 :Real-Time IDS Log Monitoring by typing this command

**“sudo ./lab.support.files/scripts/cyberops\_extended\_topo\_no\_fw.py”**

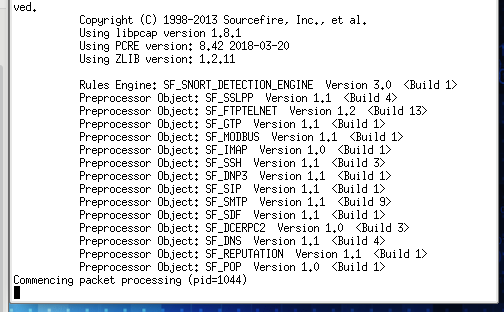


Step 2: From mininet we can open the new Shell typing xterm R1

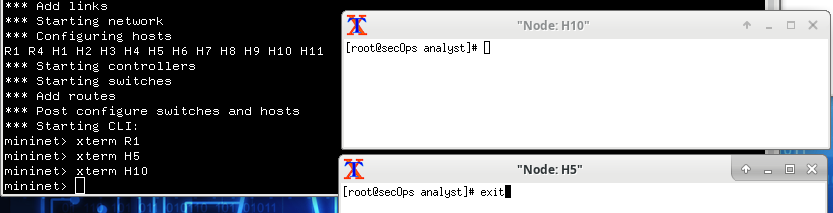


Step 3: From **R1**’s shell, start the Linux-based IDS, Snort.

“.**/lab.support.files/scripts/start\_snort.sh**”



Step 4: From the **CyberOps Workstation VMmininet** prompt, open shells for hosts **H5** and **H10**.



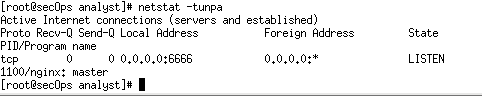
Step 5: **H10** will simulate a server and run malware on it.put command on Shell H10

“.**/lab.support.files/scripts/mal\_server\_start.sh**”

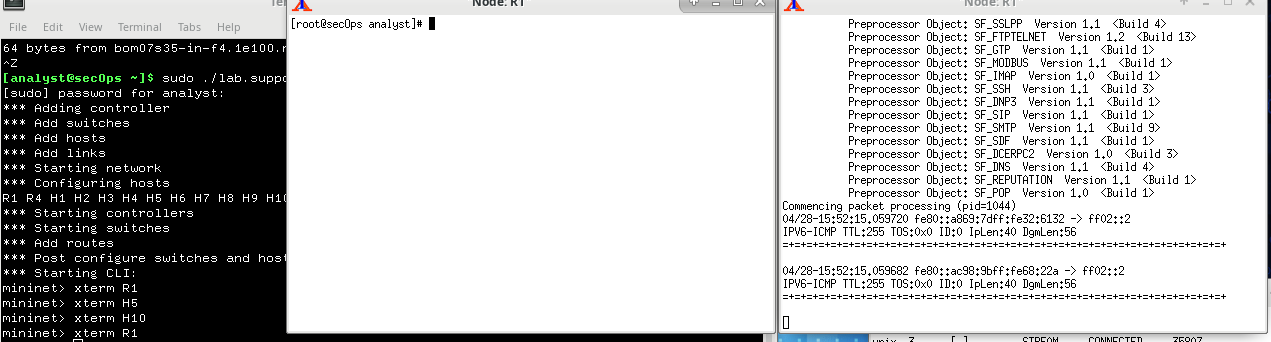


Step 6: On **H10**, use **netstat**with the **-tunpa** options to verify that the web server is running by this command

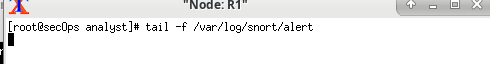
“**netstat -tunpa**”



Step 7: 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.



Step 8: In the new **R1** terminal tab, run the **tail** command “**tail -f /var/log/snort/alert**” we will get nothing because we didn't record the log.

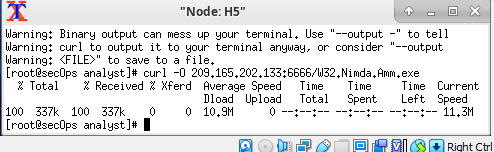


From H5, use the wget command to download a file named W32.Nimda.Amm.exe. Designed to download content via HTTP, wget is a great tool for downloading files from web servers directly from the command line.

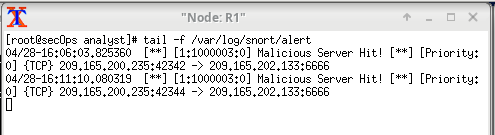
Put command

“**wget209.165.202.133:6666/W32.Nimda.Amm.exe”Or use**

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

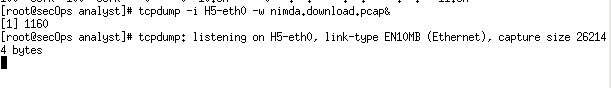


All alerts will be shown in R1 shell like this



Step 9: 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&**”



Step 10: Press **ENTER** a few times to regain control of the shell while **tcpdump** runs in background.

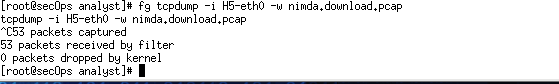
Now that **tcpdump** is capturing packets, download the malware again. On **H5**, re-run the command.

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

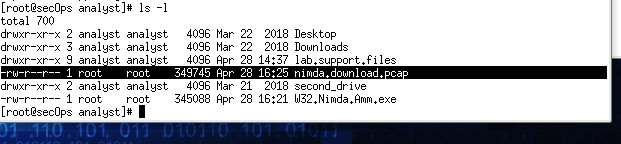


Step 11: Stop the capture by bringing**tcpdump** to foreground with the **fg** command. Because **tcpdump** was the only process sent to the background, there is no need to specify the PID. Stop the **tcpdump** process with **Ctrl+C**.The**tcpdump**process stops and displays a summary of the capture. The number of packetsmay be different for your capture.

**“fgtcpdump -i h5-eth0 -w nimda.download.pcap”**



Step 12: On **H5**, Use the **ls –l**



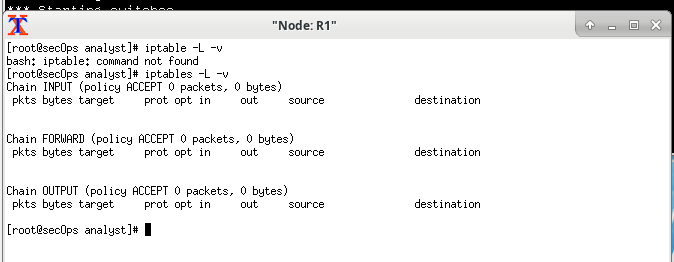
Step 13:

**Step 1: Tuning Firewall Rules Based on IDS Alerts**

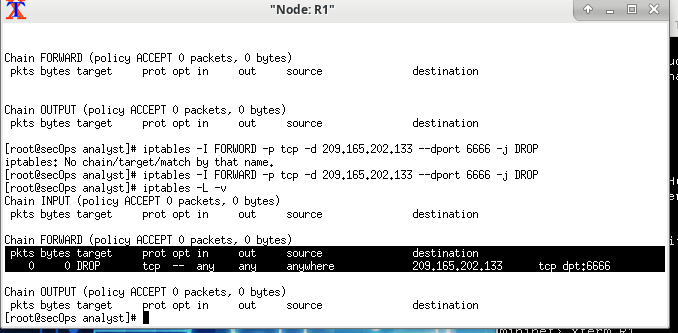
A) In the **CyberOps Workstation VM**, start a third R1 terminal window.

mininet>**xterm R1**

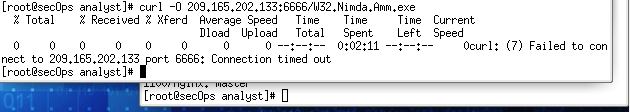
B) In the new **R1** terminal window, use the **iptables –L -v**



C) **iptables -I FORWARD -p tcp -d 209.165.202.133 --dport 6666 -j DROP**

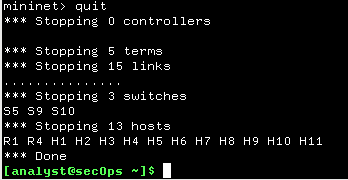


D) On **H5**, try to download the file again:

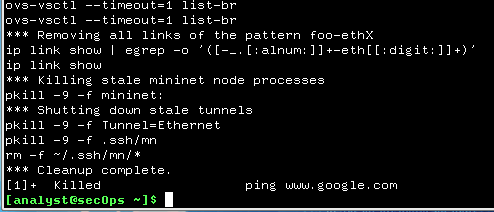


Part 3: Terminate and Clear Mininet Process.

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



B) After quitting Mininet, clean up the processes started by Mininet. Enter the password **cyberops** when prompted. “**sudomn –c”**



**Practical No 3**

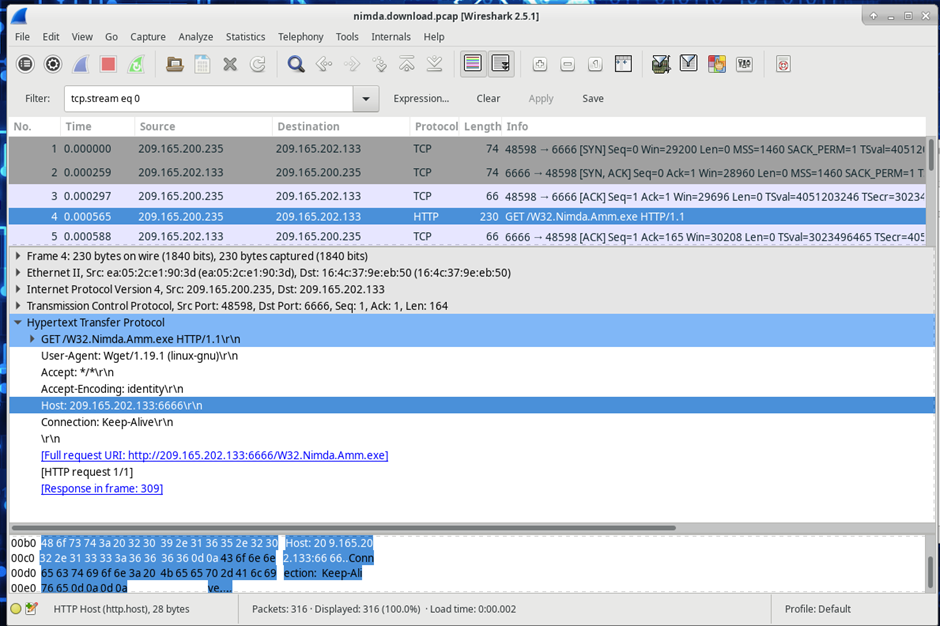
**Solution:**

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

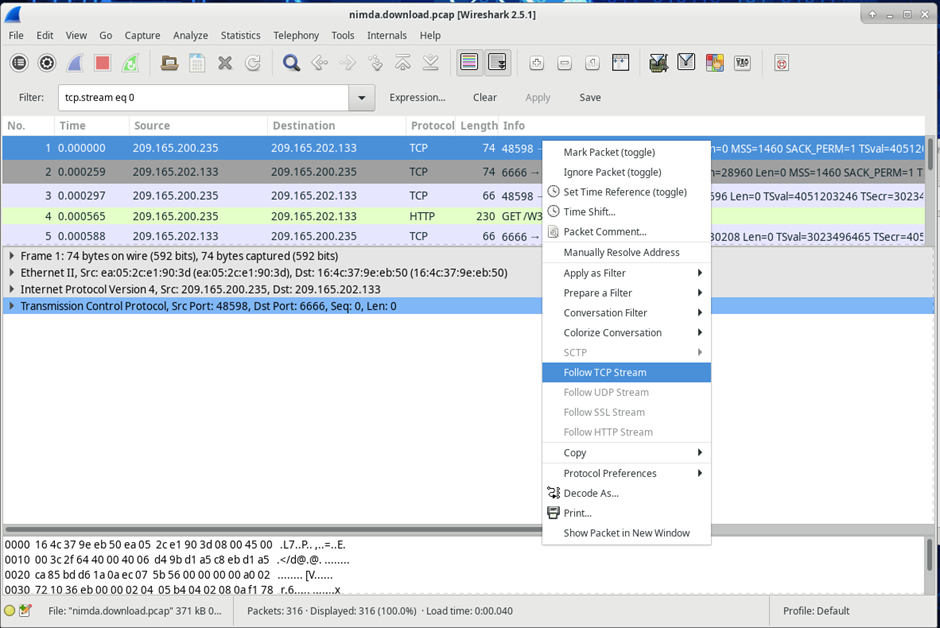
Step2: write “ls -l” list command.

Step3: On command promt“wireshark-gtknimda.download.pcap”(This will open the wireshark UI)

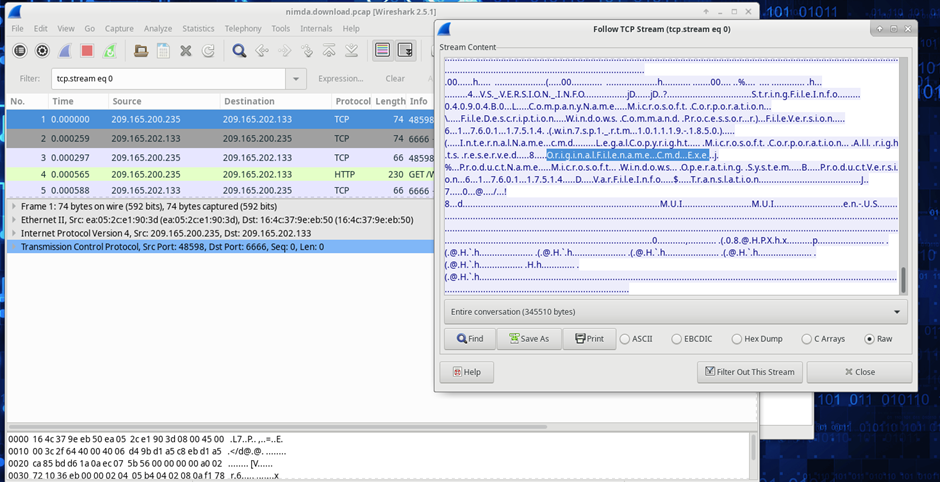
Step 4: Check HTTP and check host and full URL to download the malware file.



Step5: right click on TCP which shows top on the list. Then click on Follow TCP Stream.

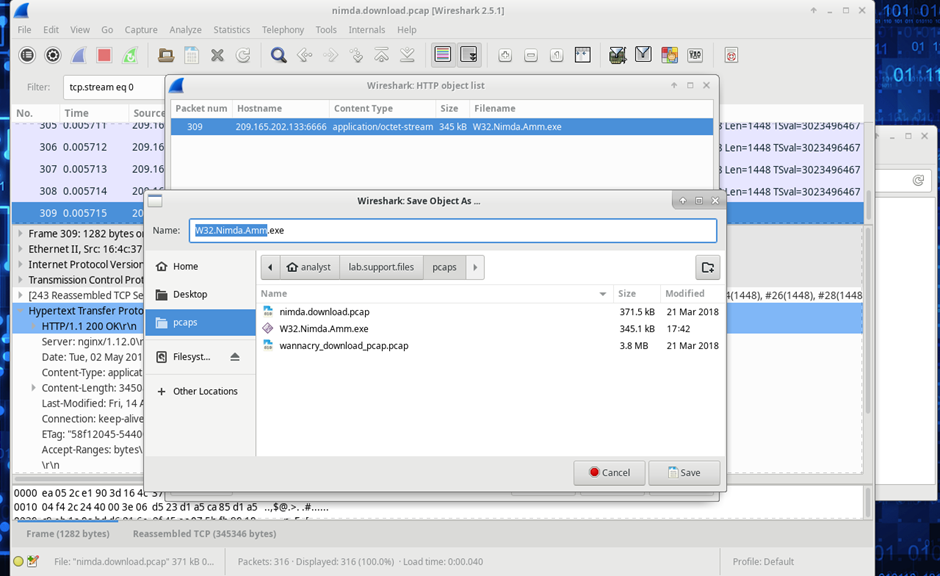


Step 6: Check the original file name in the Follow TCP Stream window.

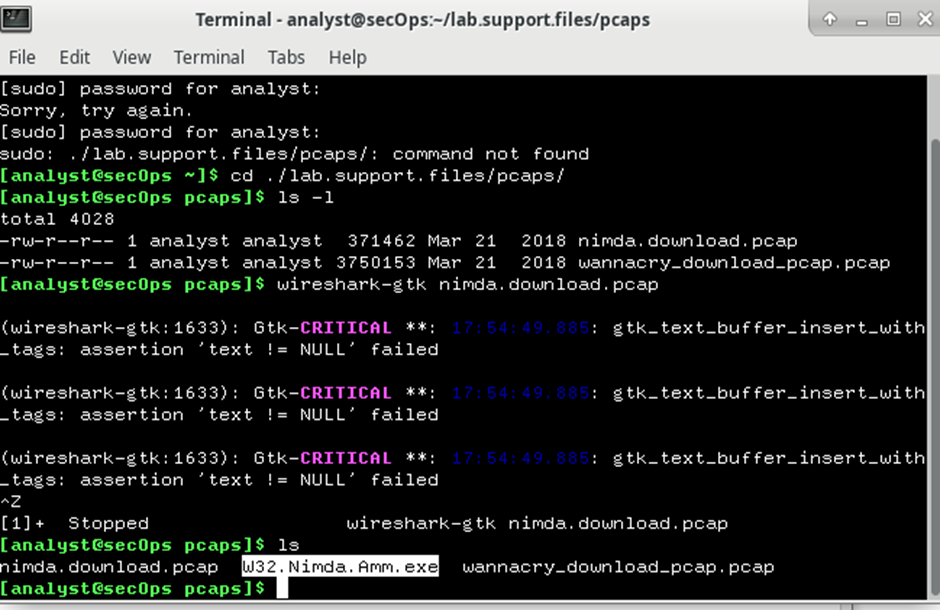


Step 7: Now we need to download and check that file by uploading to an online virustotal website.

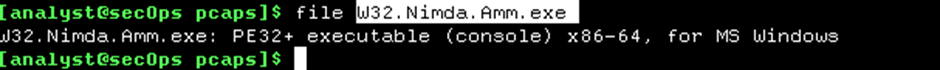
Find exe file from HTTP>click file> select export obj > Select exe File >Save as > Select Folder> Save.



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



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



**Practical No 4**

# **Objectives:**

**Part 1: Capture DNS Traffic**

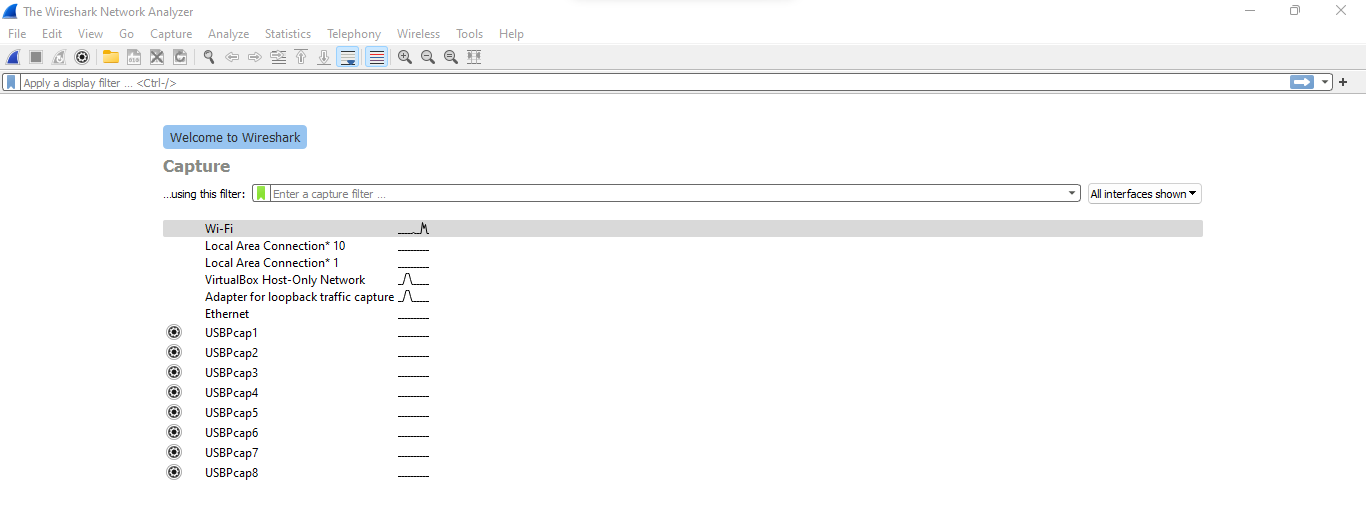
**Part 2: Explore DNS Query Traffic**

**Part 3: Explore DNS Response Traffic**

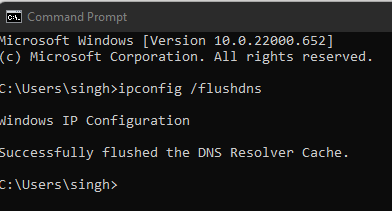
**Solution:**

**Part 1: Capture DNS Traffic**

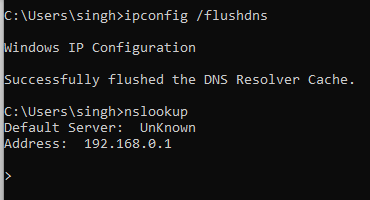
Step 1: Open **Wireshark** and start a Wireshark capture by double clicking a network interface with traffic.



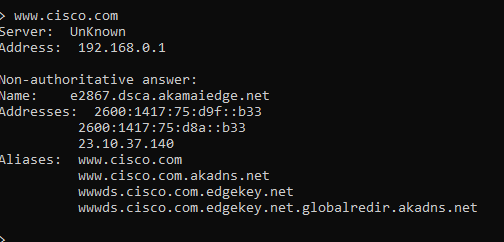
Step 2: At the Command Prompt, enter **ipconfig /flushdns** clear the DNS cache.



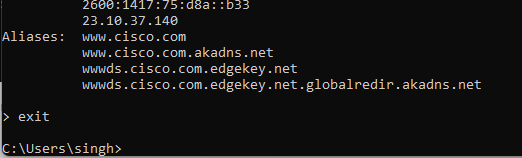
Step 3: Enter **nslookup** at the prompt to enter the nslookup interactive mode.



Step 4: Enter the domain name of a website. The domain name [www.cisco.com](https://www.cisco.com/)

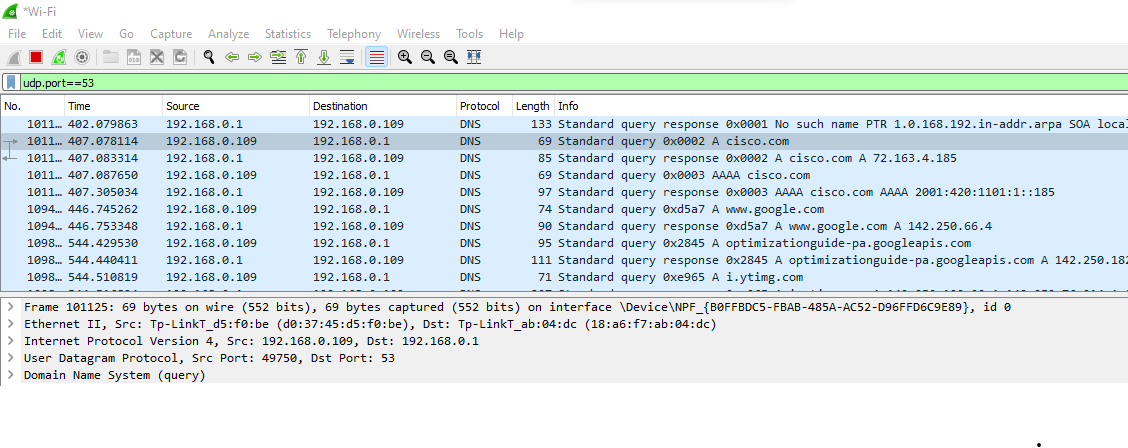


Step 5: type **exit in prompt it will exit the nslookup**

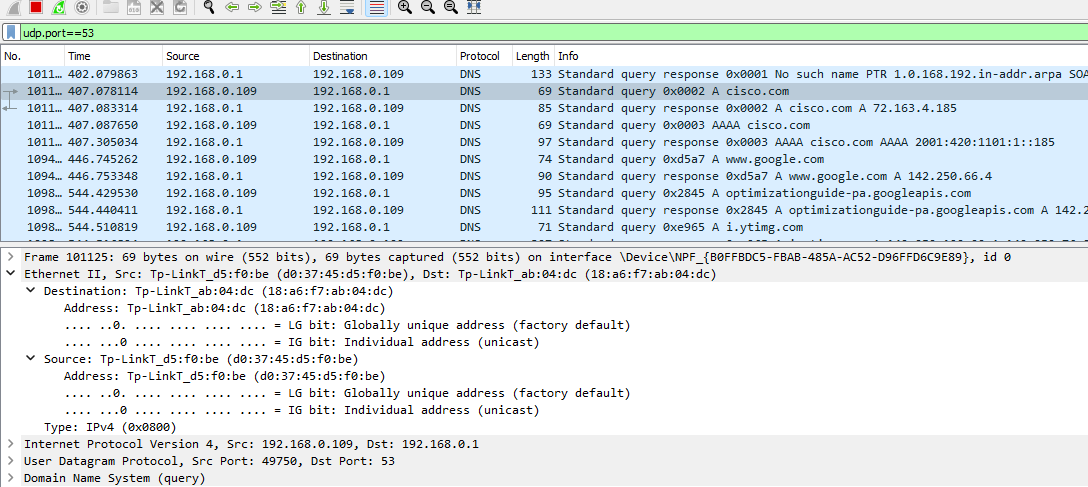


**Part 2: Explore DNS Query Traffic**.

1. 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.
2. 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).

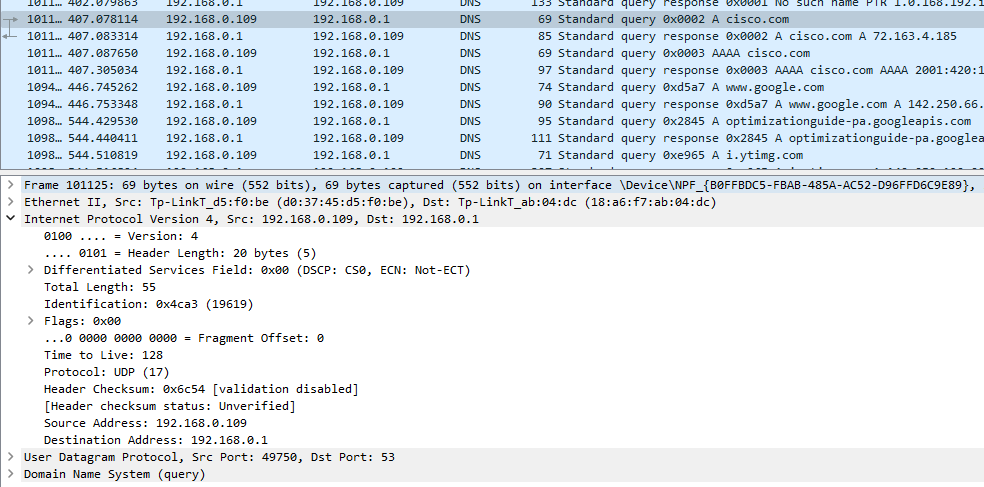


1. Expand **Ethernet II** to view the details. Observe the source and destination fields.



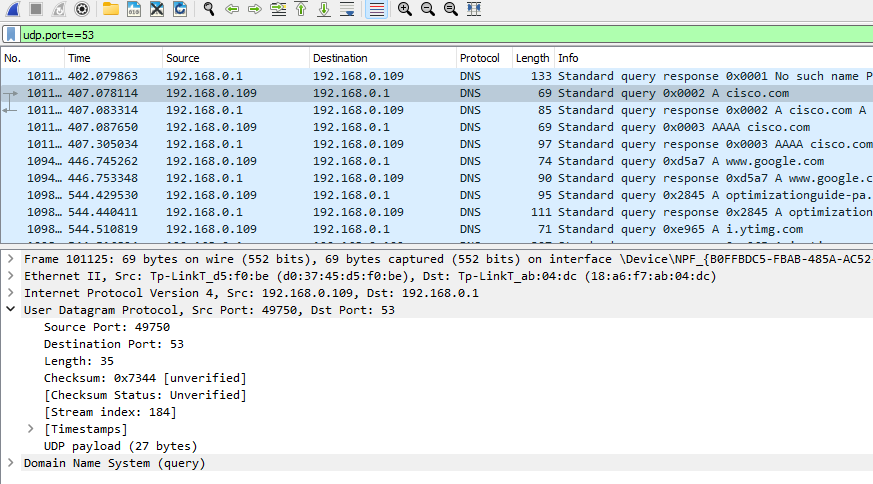
What are the source and destination MAC addresses? Which network interfaces are these MAC addresses associated with?

* 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.



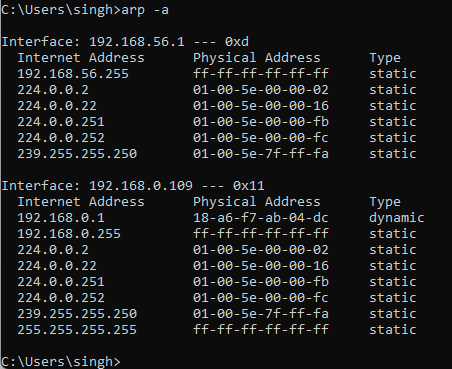
What are the source and destination IP addresses? Which network interfaces are these IP addresses associated with?

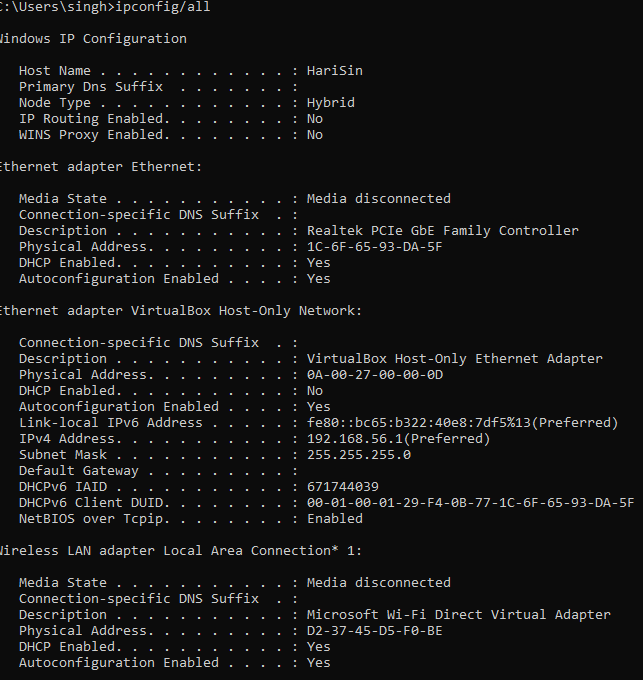
* In this example, the source IP address is associated with the NIC on the PC and the destination IP address is associated with the DNS server.
* Expand the **User Datagram Protocol**. Observe the source and destination ports.



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

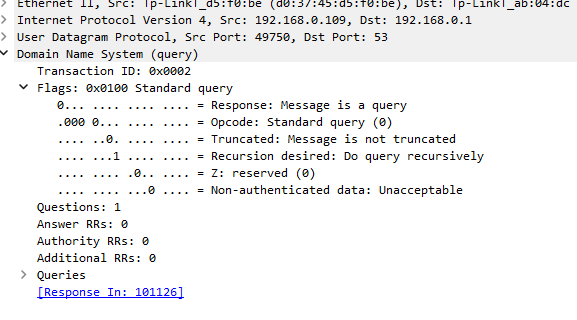
**

**

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

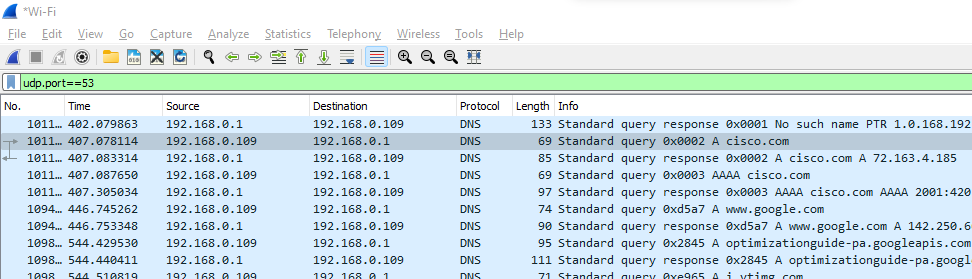
***Type your answers here.***

* 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.

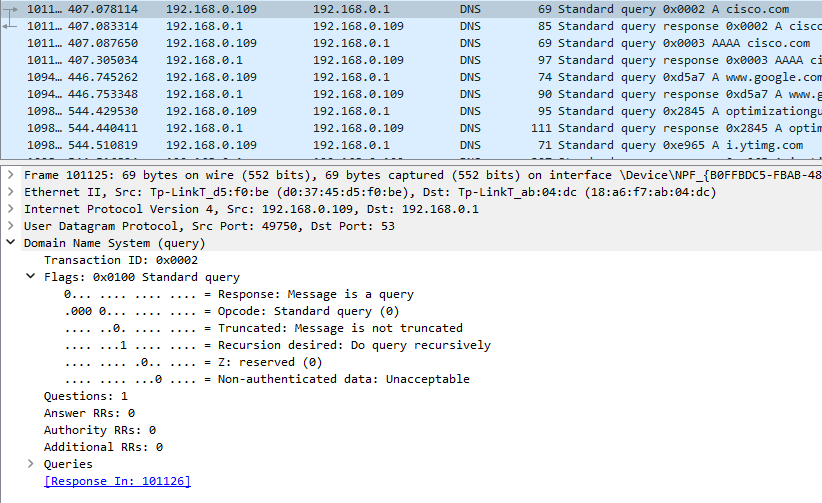
****

## **Part 3:Explore DNS Response Traffic**

Step 1: Select the corresponding response DNS packet labeled **Standard query response 0x0002 A** [**www.cisco.com**](http://www.cisco.com).



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



**Practical No 5**

**Solution:**

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

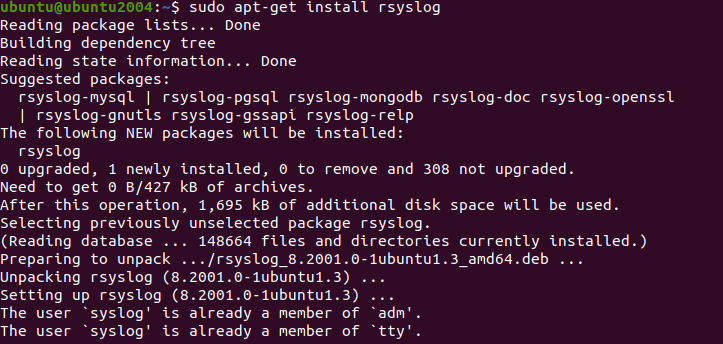
**“sudosystemctl status rsyslog”**



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

**“sudo apt-get update”**

**“sudo apt-get install rsyslog”**

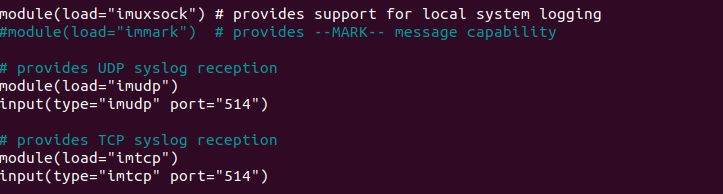


Step 3: Open rsyslog configuration file

**“sudo nano /etc/rsyslog.conf”**



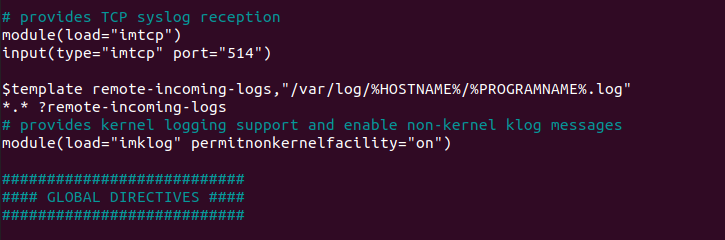
Step 4: Uncomment above four lines that enable udp and tcp port binding:



Step 5: Add template right before GLOBAL DIRECTIVES section.

**$template remote-incoming-logs,"/var/log/%HOSTNAME%/%PROGRAMNAME%.log"**

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



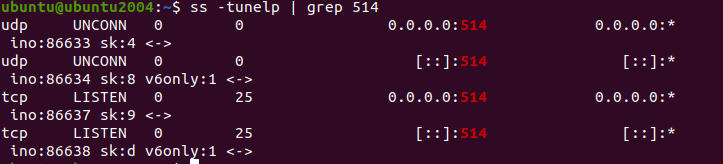
Step 6: Save and restart rsyslog service:

**“sudosystemctl restart rsyslog”**



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

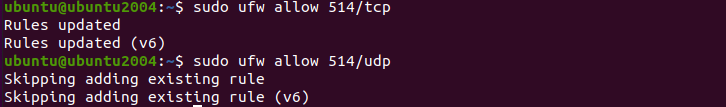
**“ss -tunelp | grep 514”**



Step 8: Allow rsyslog firewall port rules

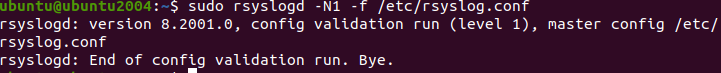
**“sudoufw allow 514/tcp”**

**“sudoufw allow 514/udp”**



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

**“sudorsyslogd -N1 -f /etc/rsyslog.conf”**



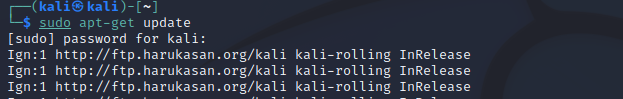
**Practical No 6**

**Solution:**

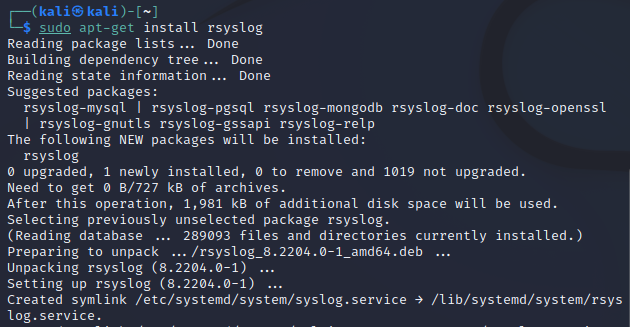
Step 1: Install and configure rsyslog server first for that please refer practical no 5.

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

**“sudo apt-get update”**

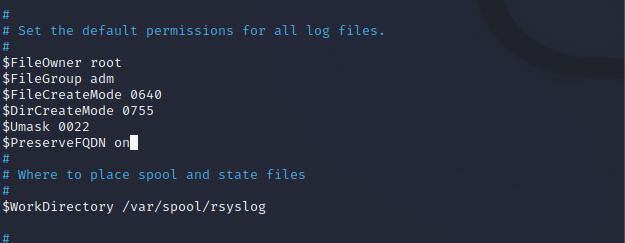


**“sudo apt-get install rsyslog”**



Step 3:Open rsyslog configuration file

**“sudo nano /etc/rsyslog.conf”**

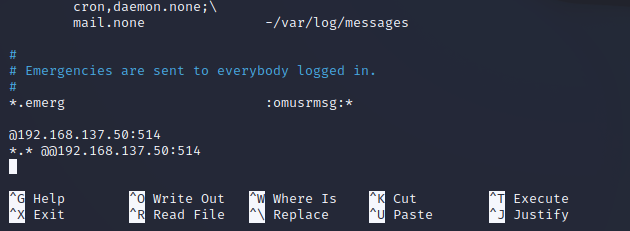


Step 4:Add above lines at the end of the file

**@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.

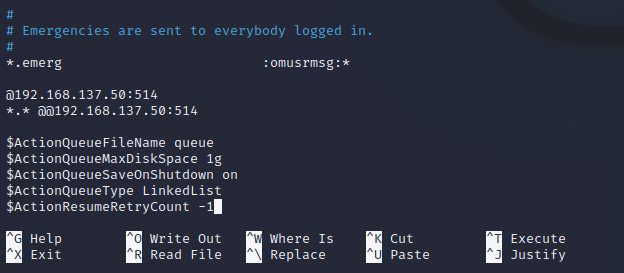
**$ActionQueueFileName queue**

**$ActionQueueMaxDiskSpace 1g**

**$ActionQueueSaveOnShutdown on**

**$ActionQueueType LinkedList**

**$ActionResumeRetryCount -1**



Step 6: Then Save and exit the file

Step 7: restart the rsyslog service

**“sudosystemctl restart rsyslog”**

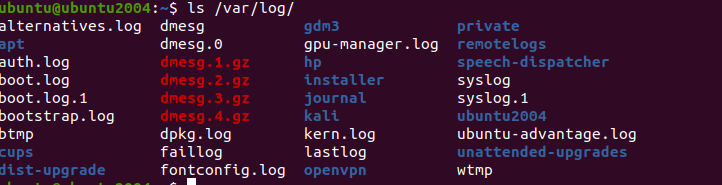


**Verify the logs**

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

Step 8:Go to your Rsyslog server to verify the logs from your client machine

**“ls /var/log/”**

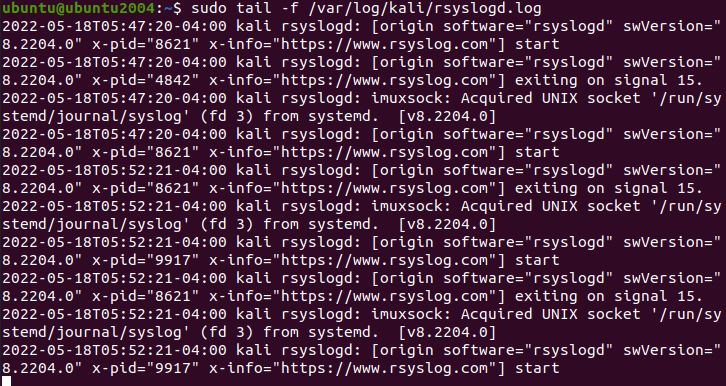


In my case, the directory named **kali** is the name of my client machine which I am currently using. We will enter this directory and see something like this:



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

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



**Practical No 7**

**Solution:**

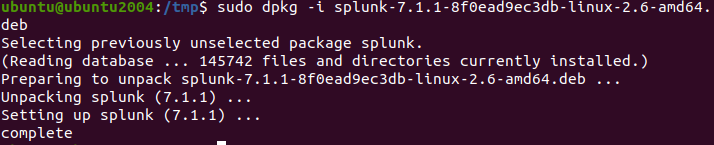
Step1: Download Splunk Installer

**“cd /tmp&&wget**[**https://download.splunk.com/products/splunk/releases/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb**](https://download.splunk.com/products/splunk/releases/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb)**”**



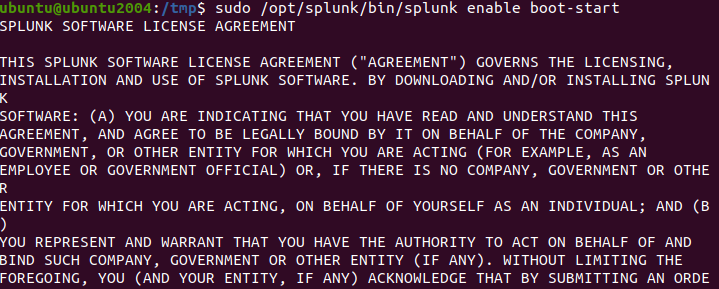
Step 2: Install Splunk

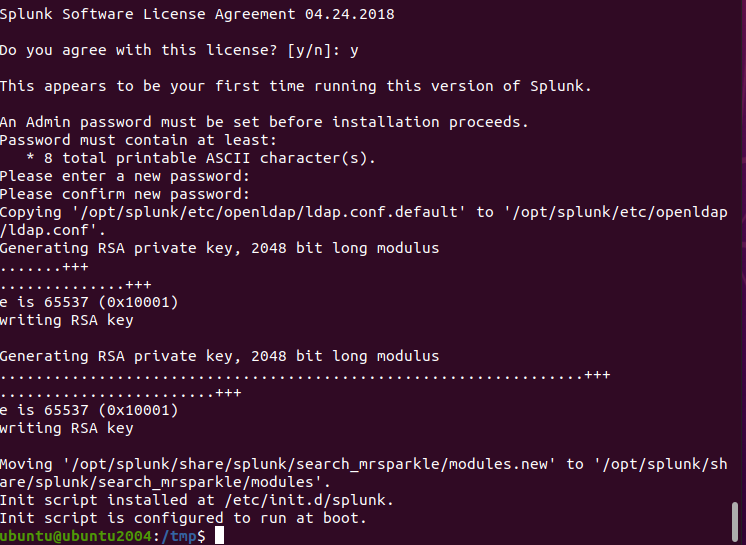
**“sudodpkg -i splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb”**



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.





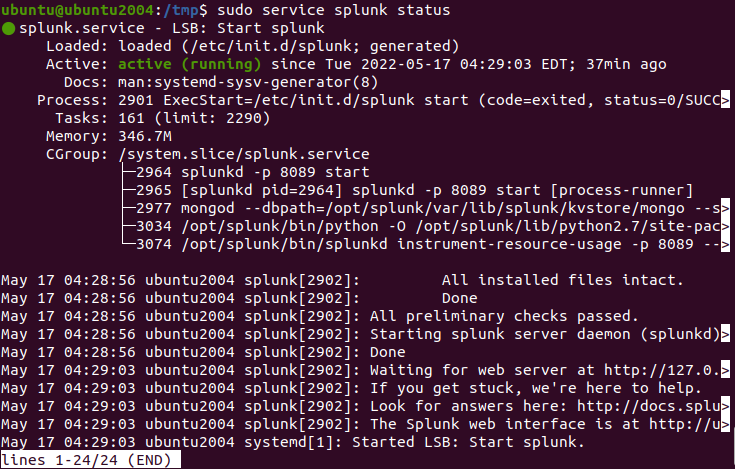
Step 4: Start the Splunk service

**“sudo service splunk start”**

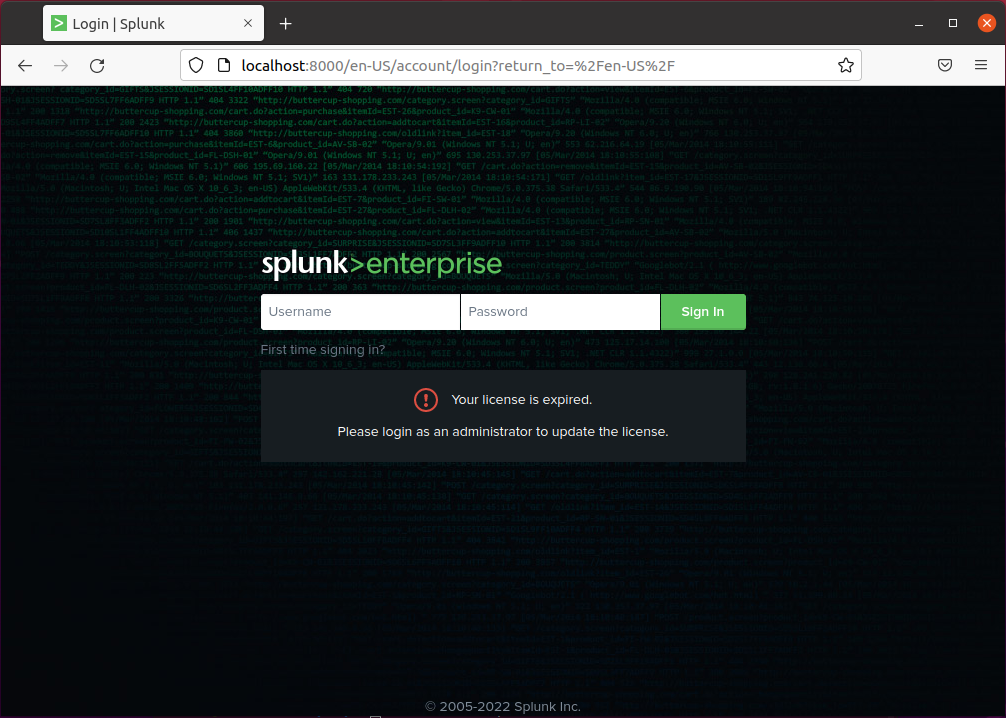


Step 5: Check splunk service Status

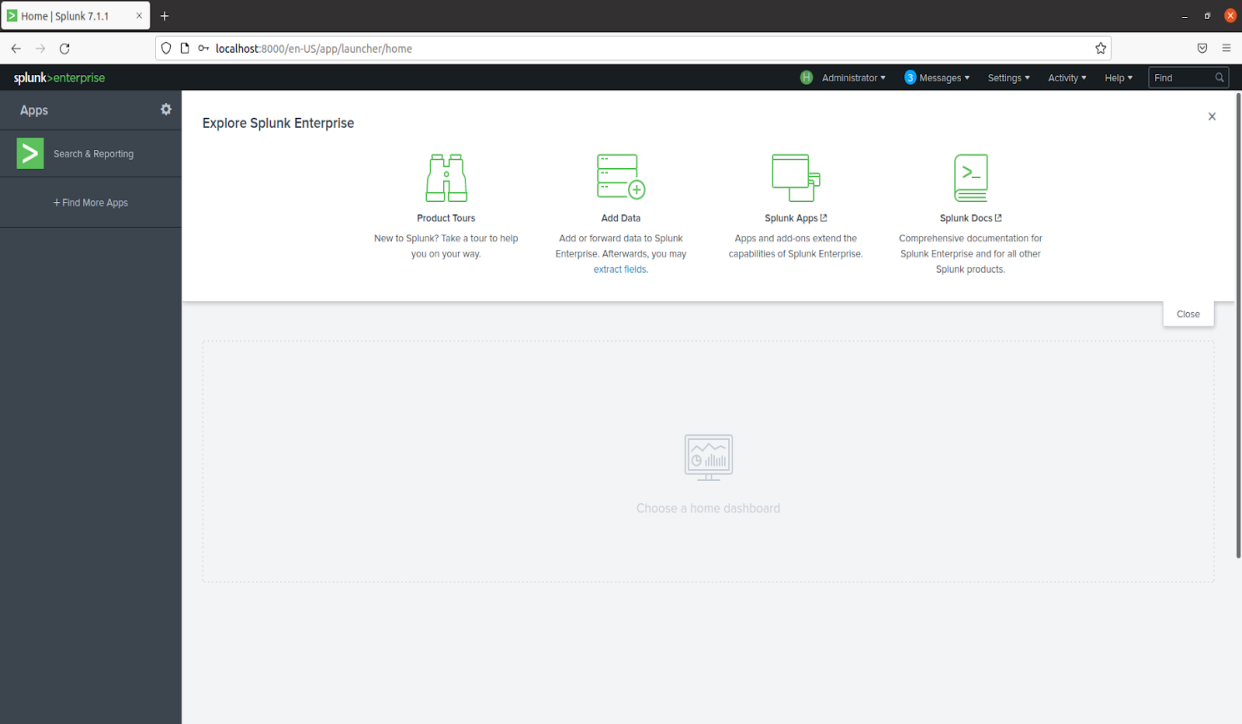
**“sudo service splunk status”**



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



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



**Practical No 8**

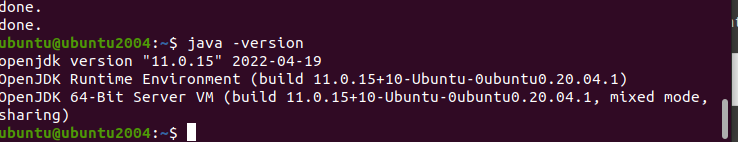
**Solution:**

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

**“sudo apt update”**

**“sudo apt install -y apt-transport-https openjdk-11-jre-headless uuid-runtime pwgen curl dirmngr”**

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



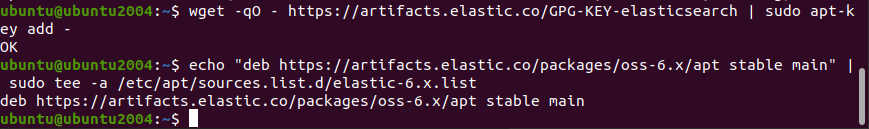
Part 2: Install Elastic Search – Elasticsearch store logs coming from external sources and offers real-time distributed search and analytics with the RESTful web interface.

Step 1: Download and install the GPG signing key.

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

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

**“echo "deb https://artifacts.elastic.co/packages/oss-6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list”**



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

**“sudo apt update”**

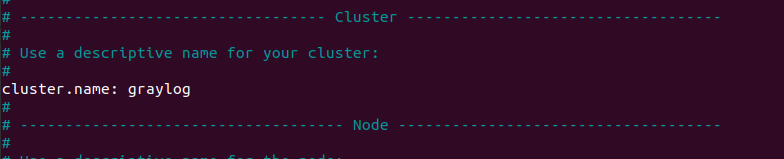
**“sudo apt install -y elasticsearch-oss”**

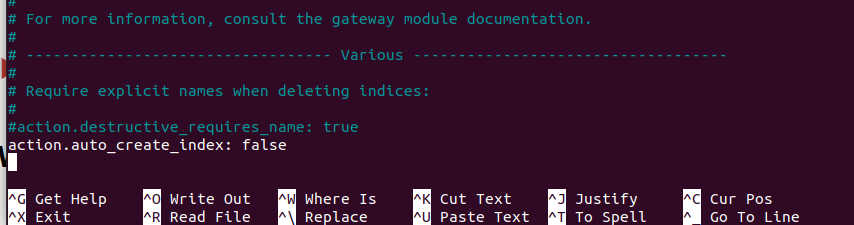
Step 4: Edit the Elasticsearch configuration file to set the cluster name for Graylog set up.

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

Step 5: Set the cluster name as graylog, as shown below. Then, uncomment the line and below add this line.

“**action.auto\_create\_index: false**” then save.



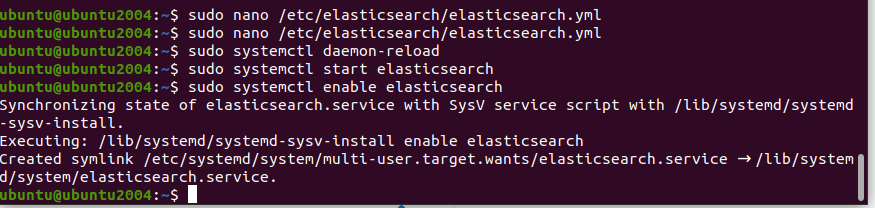


Step 6: Start the Elasticsearch service to read the new configurations.

**“sudosystemctl daemon-reload”**

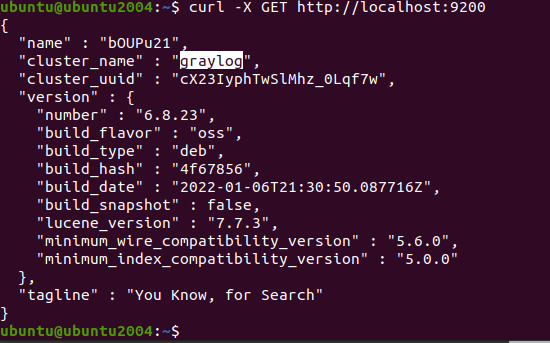
**“sudosystemctl start elasticsearch”**

**“sudosystemctl enable elasticsearch”**



Step 7: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 ”**



Part 3: 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”**

**“sudo apt install -y mongodb-server”**

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

**“sudosystemctl start mongodb”**

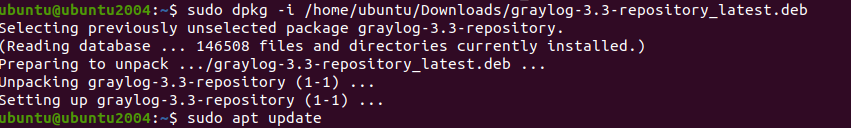
**“sudosystemctl enable mongodb”**

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-3.3-repository\_latest.deb”**

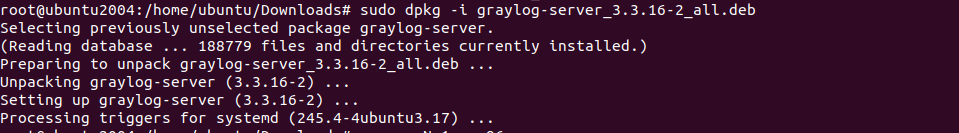
**“sudodpkg -i graylog-3.3-repository\_latest.deb”**



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

Step 3: Install the Graylog server using the following command.

**“sudodpkg -igraylog-server”**



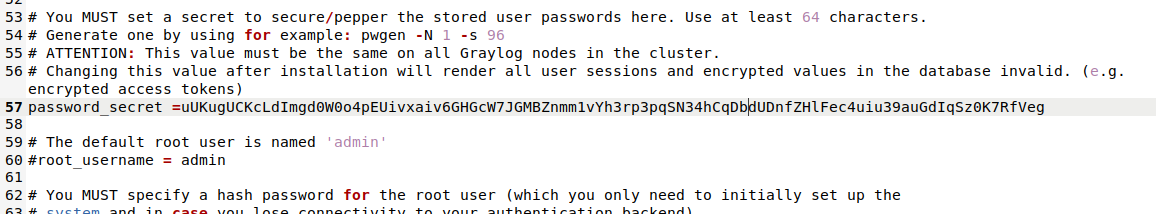
Step 4: You must set a secret to secure the user passwords. Use the pwgen command to generate the secret.

**“pwgen -N 1 -s 96”**



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

Then, place the secret like below.

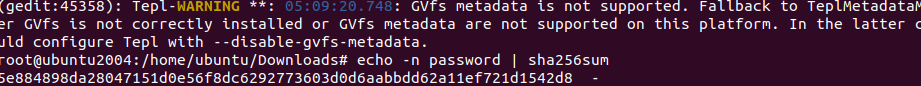


Step 6: Now, generate a hash (sha256) password for the root user (not to be confused with the system user, the root user of graylog is admin).

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”**



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

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



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.

**“sudogedit /etc/graylog/server/server.conf”**

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

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

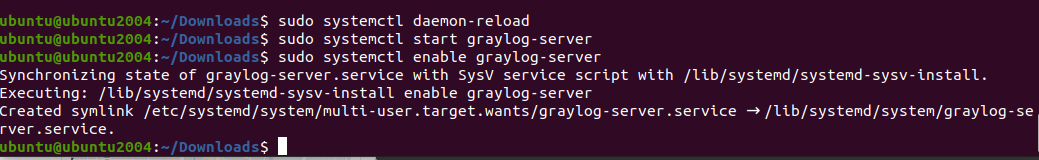
Step 2: Start and enable the Graylog service.

Place the below command

**“sudosystemctl daemon-reload”**

**“sudosystemctl start graylog-server”**

**“sudosystemctl enable graylog-server”**



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.

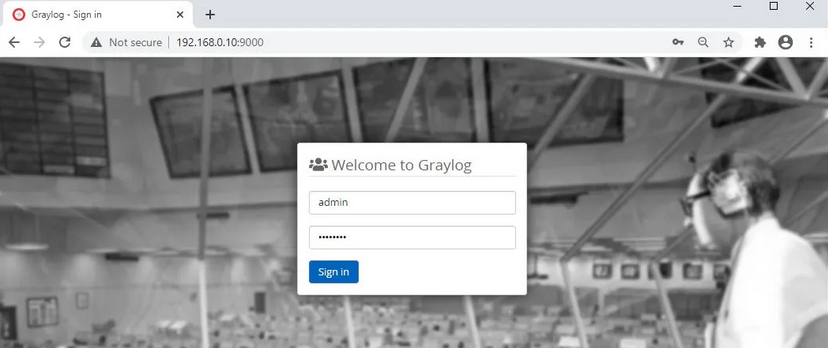
You will able to see the log file.

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](https://www.itzgeek.com/post/how-to-install-google-chrome-on-ubuntu-20-04/) and point it to.

**“**[**http://ip.add.re.ss:9000**](http://ip.add.re.ss:9000)**”**type in browser.



**Practical No 9**

### **Objectives:**

* **Part 1: Normalize Timestamps in a Log File**
* **Part 2: Normalize Timestamps in an Apache Log File**

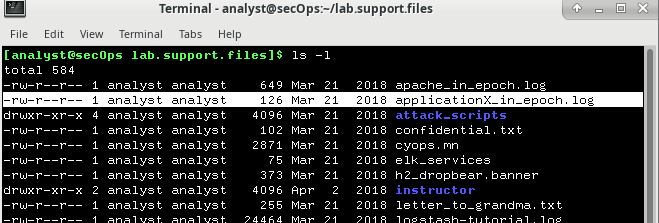
**Solution:**

**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/”**

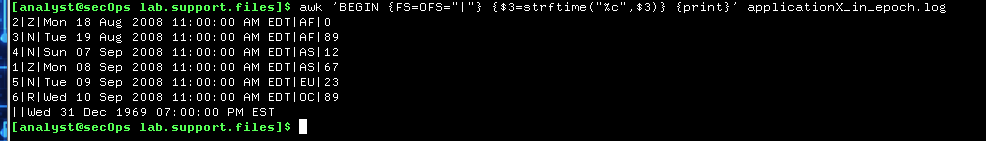
Then type **“ls –l”**



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

Write the command

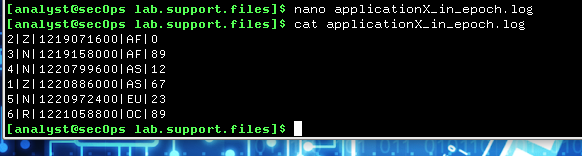
**“awk 'BEGIN {FS=OFS="|"} {$3=strftime("%c",$3)} {print}' applicationX\_in\_epoch.log”**



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.**

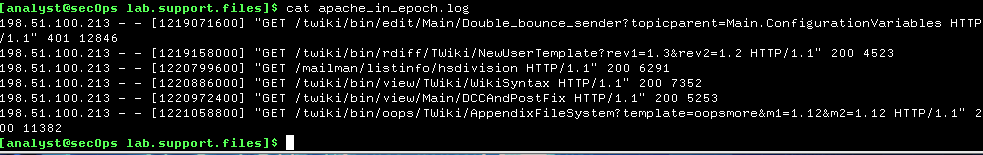
Step 4: Use **nano** (or your favorite text editor) to remove the extra empty line at the end of the file



#### **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.

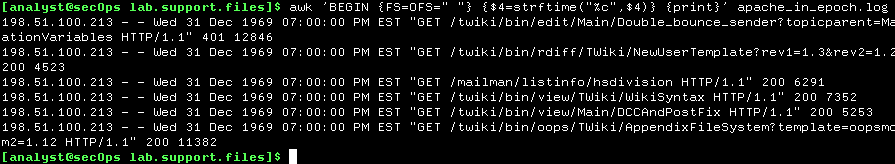
Step 1: Open the terminal and type **cat apache\_in\_epoch.log.**



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.

**“awk 'BEGIN {FS=OFS=" "} {$4=strftime("%c",$4)} {print}' apache\_in\_epoch.log”**



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,

**“awk 'BEGIN {FS=OFS=" "} {gsub(/[|]/,"",$4)}{print}{$4=strftime("%c",$4)}{print}' apache\_in\_epoch.log”**

