INDEX

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

```
Hi Grandma,
I am writing this letter to thank you for the chocolate chip cookies you sent me. I
got them this morning and I have already eaten half of the box! They are absolutely
delicious!

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

Step 4: 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  -
out message.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
```

Step 5: cat message.enc

```
[analyst@secOps lab.support.files]$ cat message.enc
L[illegible]
[illegible]
[illegible]
[illegible]
[analyst@secOps lab.support.files]$
```

Step 6: openssl aes-256-cbc -a -in letter_to_grandma.txt -out message.enc

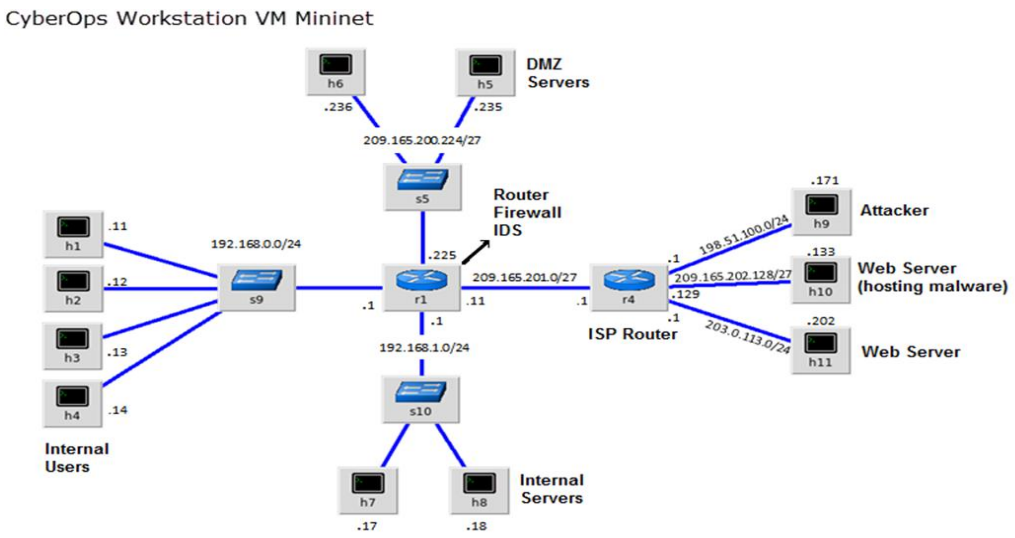
```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -in letter_to_grandma.txt
 -out message.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
```

Step 7: 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.
```

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**"

```
[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.
```

Step 2: Use the **ifconfig** command to verify that your Internet is working and type ping command "**ping www.google.com**"

```
[analyst@secOps ~]$ ping www.google.com
PING www.google.com (142.250.66.4) 56(84) bytes of data.
64 bytes from bom07s35-in-f4.1e100.net (142.250.66.4): icmp_seq=1 ttl=119 time=4
.99 ms
64 bytes from bom07s35-in-f4.1e100.net (142.250.66.4): icmp_seq=2 ttl=119 time=5
.19 ms
64 bytes from bom07s35-in-f4.1e100.net (142.250.66.4): icmp_seq=3 ttl=119 time=8
.87 ms
64 bytes from bom07s35-in-f4.1e100.net (142.250.66.4): icmp_seq=4 ttl=119 time=5
.00 ms
64 bytes from bom07s35-in-f4.1e100.net (142.250.66.4): icmp_seq=5 ttl=119 time=5
.11 ms
^Z
[1]+  Stopped                 ping 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"

```
[analyst@secOps ~]$ sudo ./lab.support.files/scripts/cyberops_extended_topo_no_fw.py
[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>
```

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

```
*** Post configure switches and hosts
*** Starting CLI:
mininet> xterm R1
mininet>
```

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

"**./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_POP  Version 1.0  <Build 1>
Commencing packet processing (pid=1044)
```

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

```
*** 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> xterm R1
mininet> xterm H5
mininet> xterm H10
mininet>
```

"Node: H10"

```
[root@secOps analyst]#
```

"Node: H5"

```
[root@secOps analyst]# exit
```

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**"

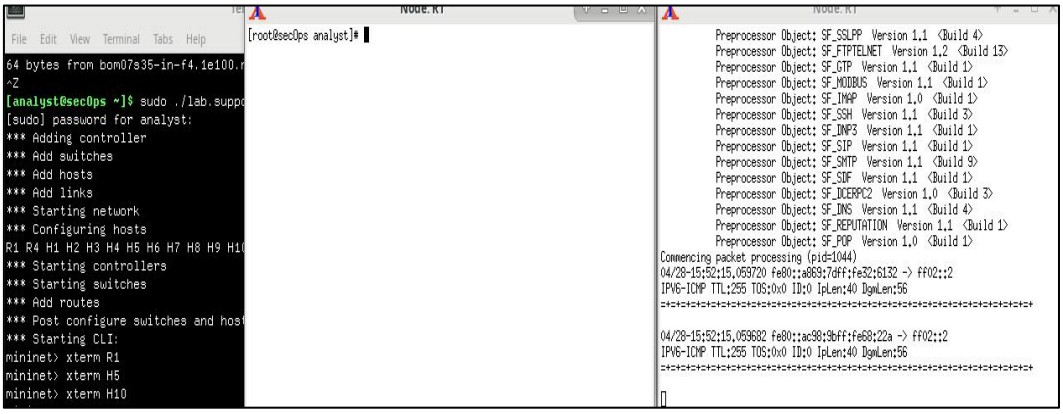
```
"Node: H10"
[root@secOps analyst]# ./lab.suport.files/scripts/mal_server_start.sh
bash: ./lab.suport.files/scripts/mal_server_start.sh: No such file or directory
[root@secOps analyst]# ./lab.support.files/scripts/mal_server_start.sh
[root@secOps analyst]#
```

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

"**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 0.0.0.0:6666            0.0.0.0:*               LISTEN
1100/nginx: master
[root@secOps analyst]#
```

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.

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

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

"**wget 209.165.202.133:6666/W32.Nimda.Amm.exe" Or use**

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

```
"Node: H5"
Warning: Binary output can mess up your terminal. Use "--output -" to tell
Warning: curl to output it to your terminal anyway, or consider "--output
Warning: <FILE>" to save to a file.
[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  10.9M      0 --:--:-- --:--:-- --:--:-- 11.3M
[root@secOps analyst]#
```

All alerts will be shown in R1 shell like this

```
"Node: R1"
[root@secOps analyst]# tail -f /var/log/snort/alert
04/28-16:06:03.825360  [**] [1:1000003:0] Malicious Server Hit! [**] [Priority:
0] {TCP} 209.165.200.235:42342 -> 209.165.202.133:6666
04/28-16:11:10.080319  [**] [1:1000003:0] Malicious Server Hit! [**] [Priority:
0] {TCP} 209.165.200.235:42344 -> 209.165.202.133:6666
```

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&**”

```
[root@secOps analyst]# tcpdump -i H5-eth0 -w nimda.download.pcap&
[1] 1160
[root@secOps analyst]# tcpdump: listening on H5-eth0, link-type EN10MB (Ethernet), capture size 26214
4 bytes
```

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”

```
[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  15.6M      0 --:--:-- --:--:-- --:--:-- 16.4M
[root@secOps analyst]#
```

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 packets may be different for your capture.

“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
^C53 packets captured
53 packets received by filter
0 packets dropped by kernel
[root@secOps analyst]#
```

Step 12: On **H5**, Use the **ls –l**

```
[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 Apr 28 14:37 lab.support.files
-rw-r--r-- 1 root    root    349745 Apr 28 16:25 nimda.download.pcap
drwxr-xr-x 2 analyst analyst   4096 Mar 21  2018 second_drive
-rw-r--r-- 1 root    root    345088 Apr 28 16:21 W32.Nimda.Amm.exe
[root@secOps analyst]#
```

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**

```
"Node: R1"
[root@secOps analyst]# iptable -L -v
bash: iptable: command not found
[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
```

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

```
"Node: R1"

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]# iptables -I FORWORD -p tcp -d 209.165.202.133 --dport 6666 -j DROP
iptables: No chain/target/match by that name.
[root@secOps analyst]# iptables -I FORWARD -p tcp -d 209.165.202.133 --dport 6666 -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
[root@secOps analyst]#
```

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

```
[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
  0     0    0     0    0     0      0      0 --:--:--  0:02:11 --:--:--     0curl: (7) Failed to con
nect to 209.165.202.133 port 6666: Connection timed out
[root@secOps analyst]#
```

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 5 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. "**sudo mn –c**"

```
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -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
rm -f ~/.ssh/mn/*
*** Cleanup complete.
[1]+  Killed                  ping www.google.com
```

Practical No 3

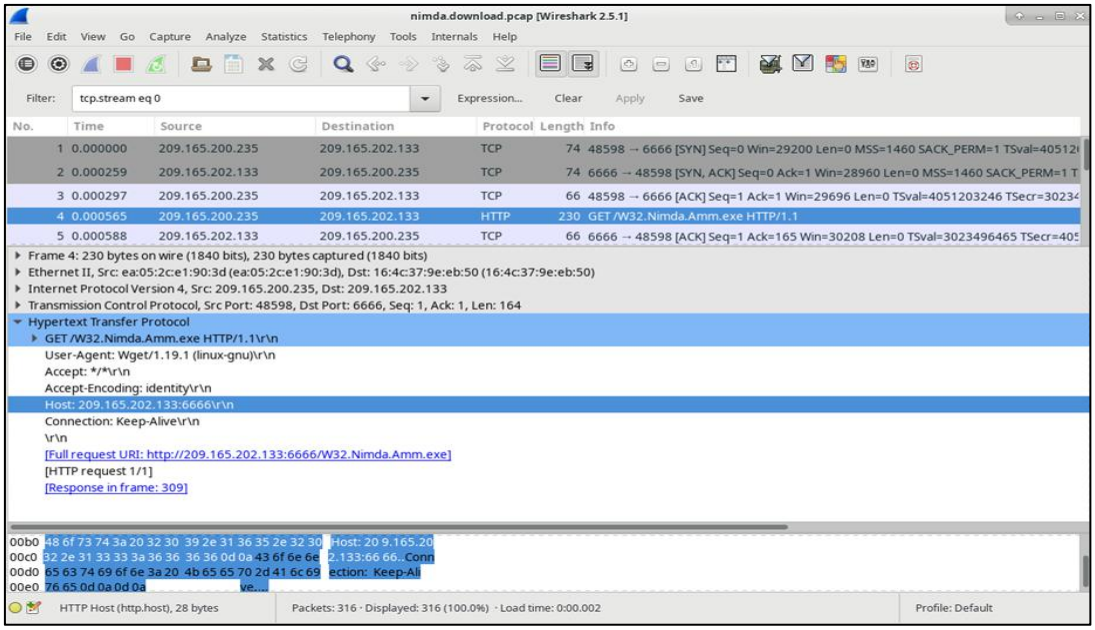
Solution:

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

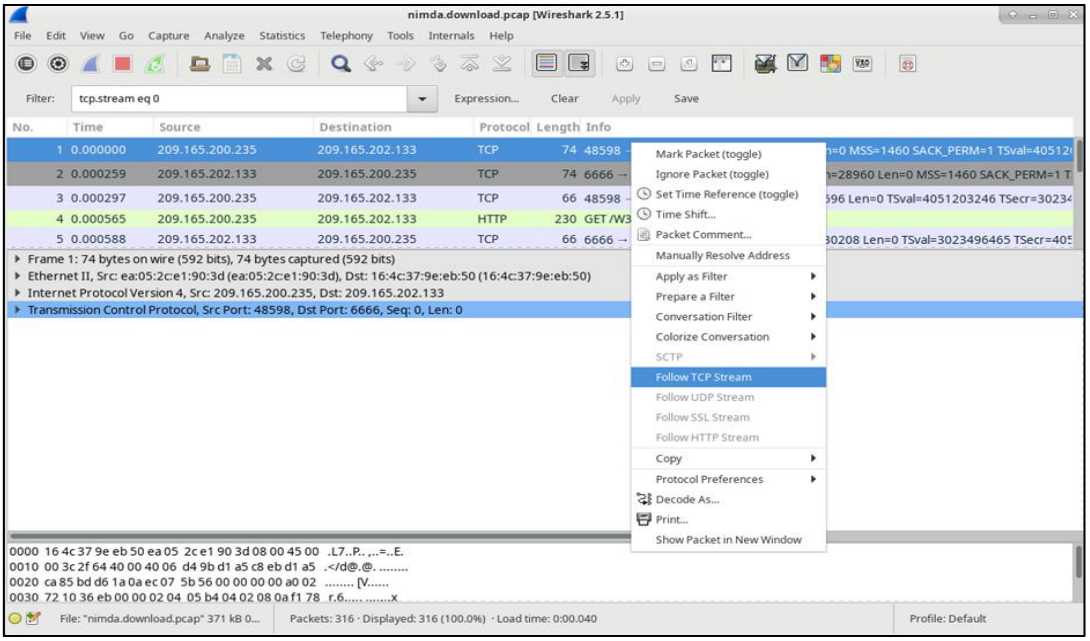
Step 2: write “ls -l” list command.

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

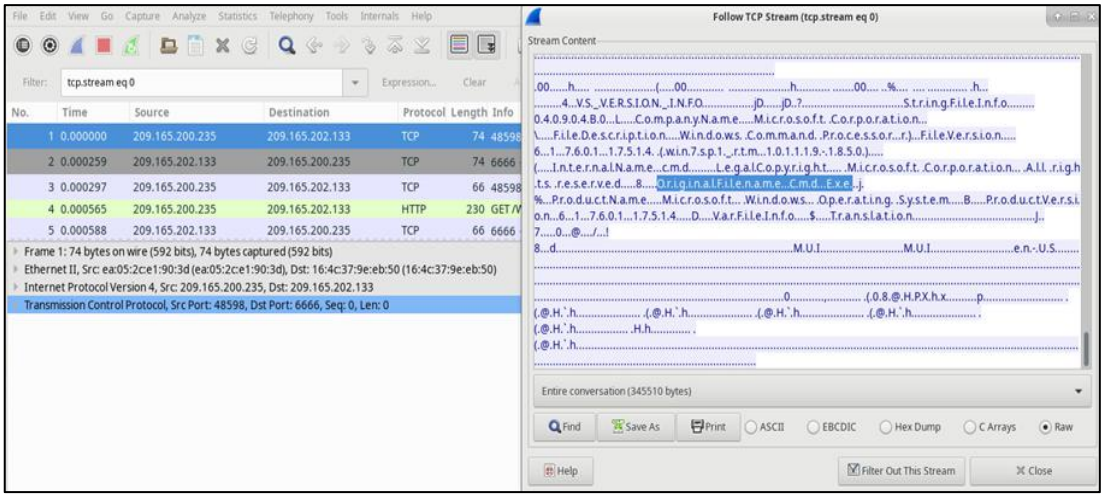
Step 4: Check HTTP and check host and full URL to download the malware file.



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

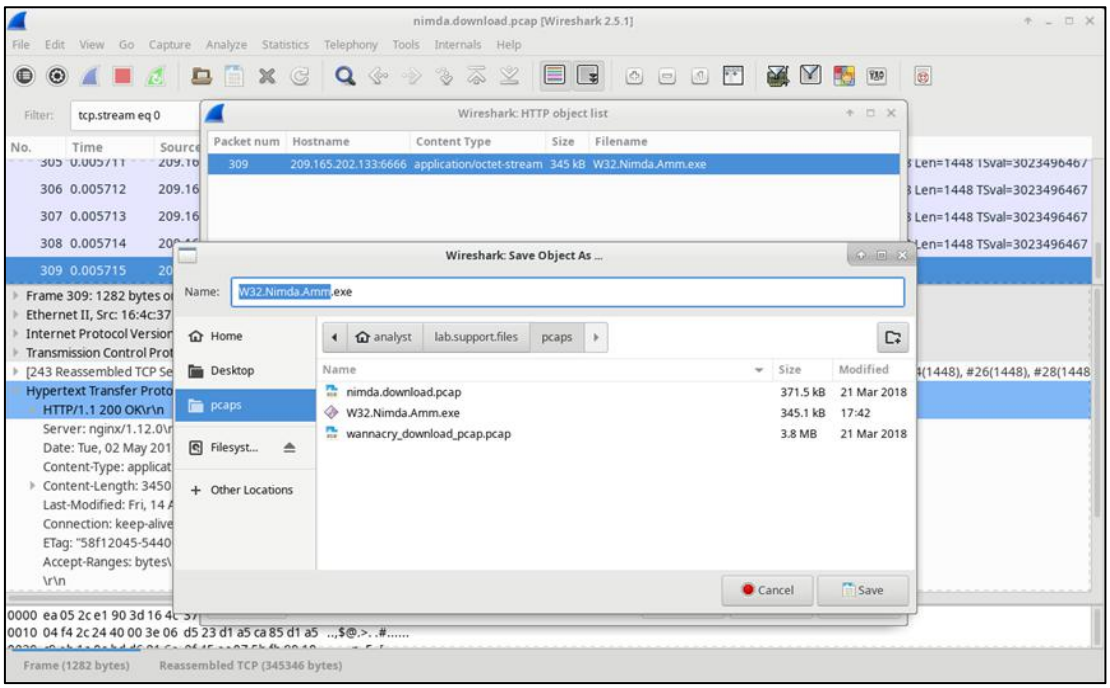


Step 6: Check the original file name in the Follow TCP Stream window.



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.



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

```
[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.pcap
```

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

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

Practical No 4

Objectives:

Part 1: Capture DNS Traffic

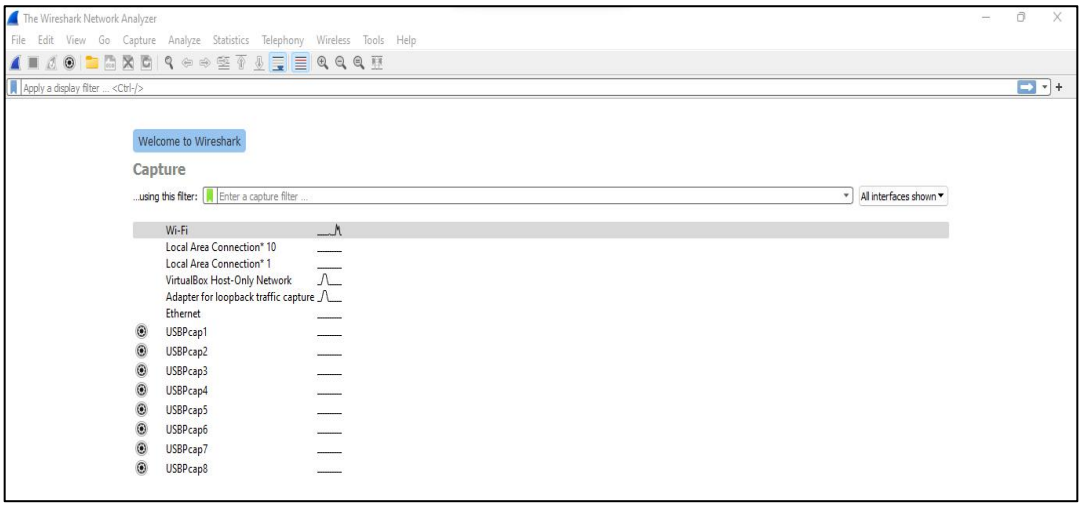
Part 2: Explore DNS Query Traffic

Part 3: Explore DNS Response Traffic

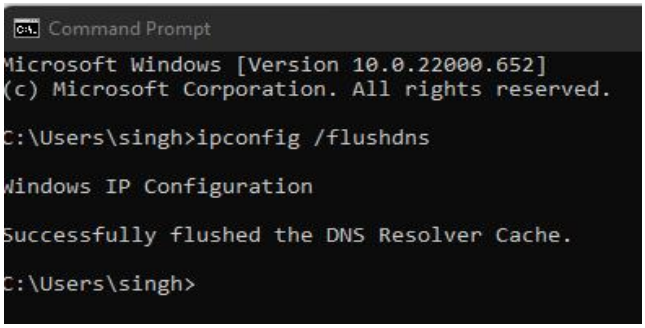
Solution:

Part 1: Capture DNS Traffic

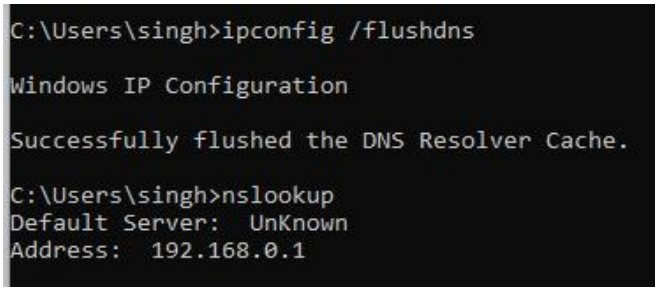
Step 1: Open **Wireshark** and start a Wireshark capture by double clicking a network interface with traffic.



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

```
> 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**

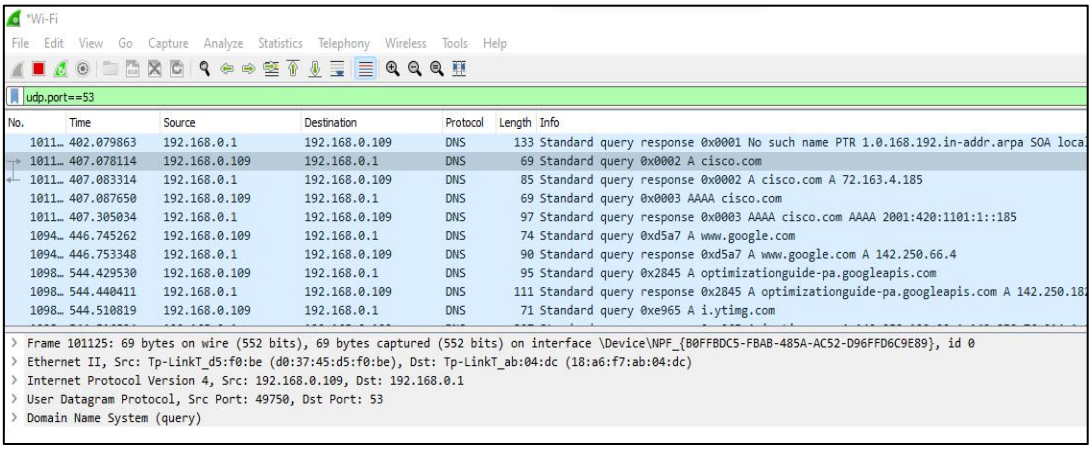
```
          2600:1417:75:d8a::b33
          23.10.37.140
Aliases:  www.cisco.com
          www.cisco.com.akadns.net
          wwwds.cisco.com.edgekey.net
          wwwds.cisco.com.edgekey.net.globalredir.akadns.net

> exit
```

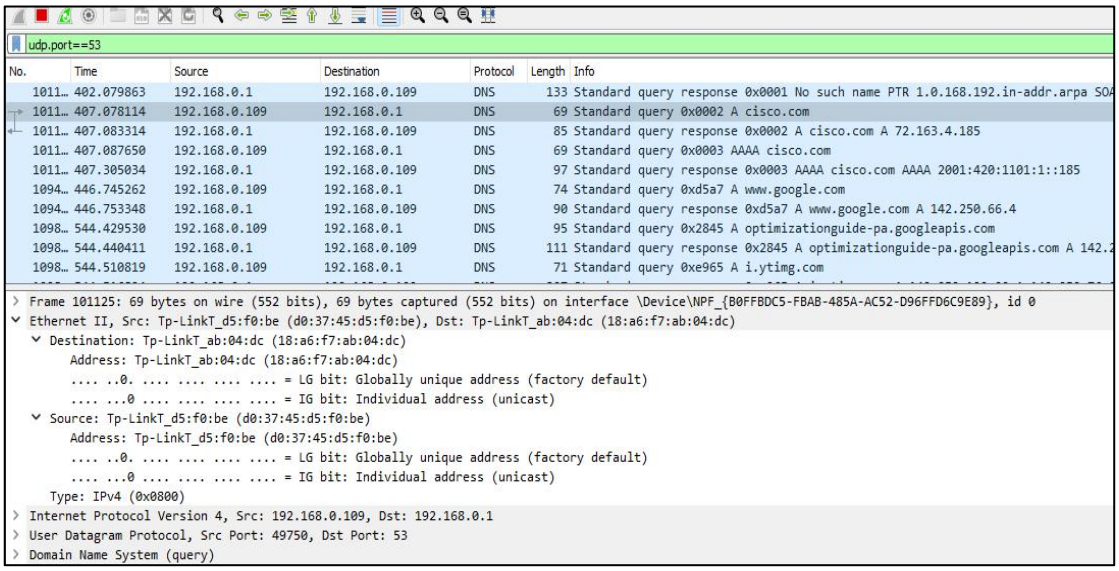
Part 2: Explore DNS Query Traffic.

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).



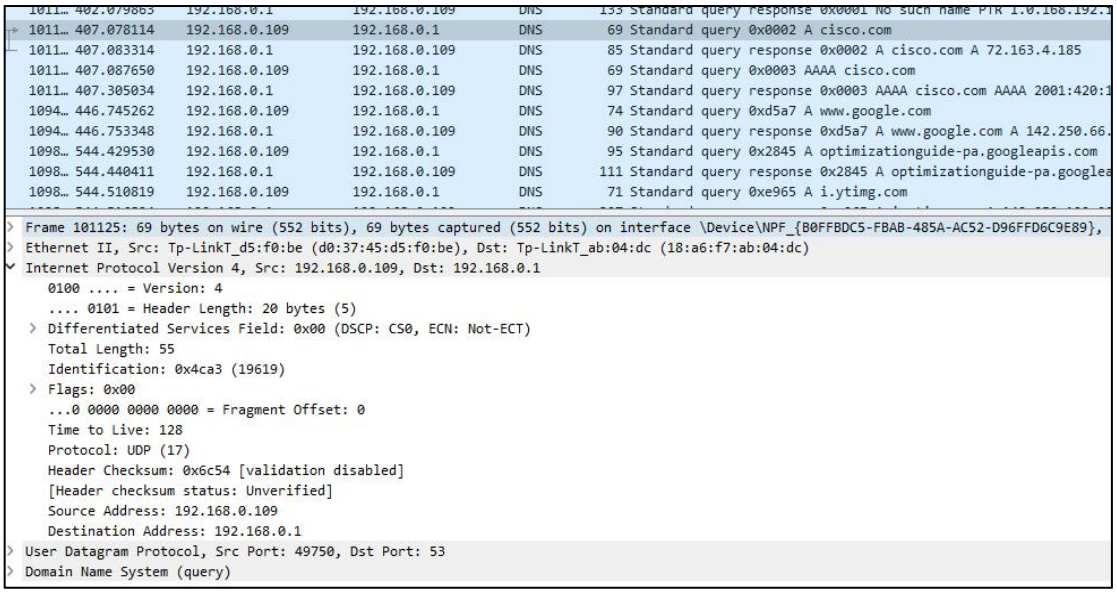
3. Expand **Ethernet II** to view the details. Observe the source and destination fields.



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.

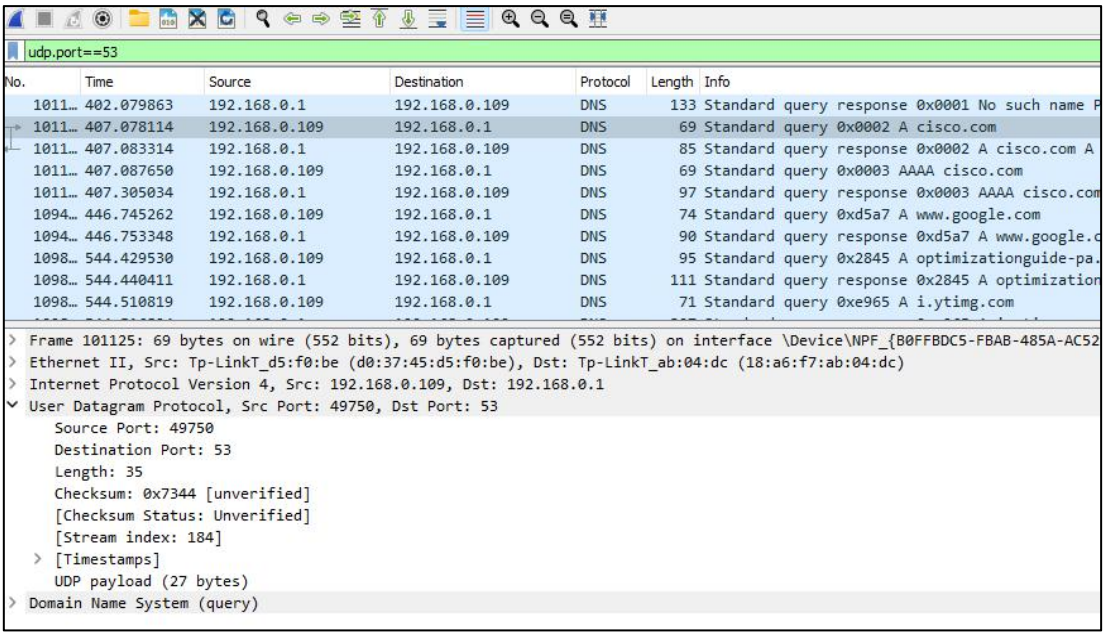
- Expand **Internet Protocol Version 4**. Observe the source and destination IPv4 addresses.



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

```
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
```

```
thernet 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

ireless 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?

  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.

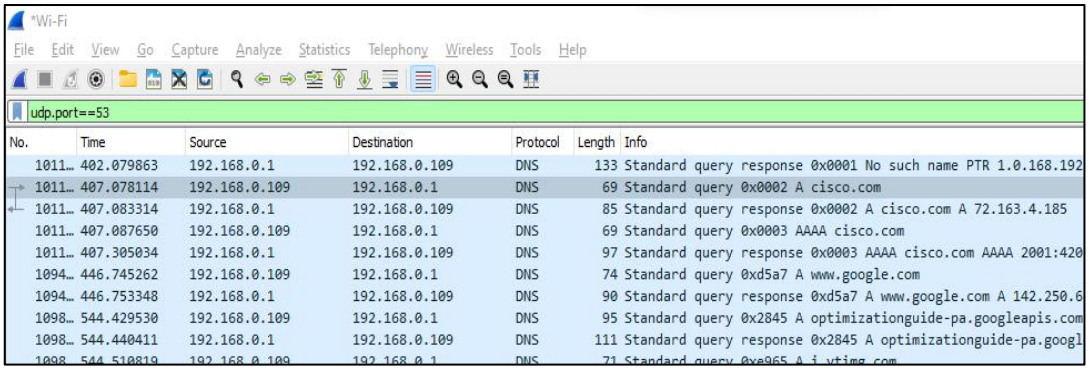
- Expand **Domain Name System (query)** in the Packet Details pane. Then expand the **Flags** and **Queries**.

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

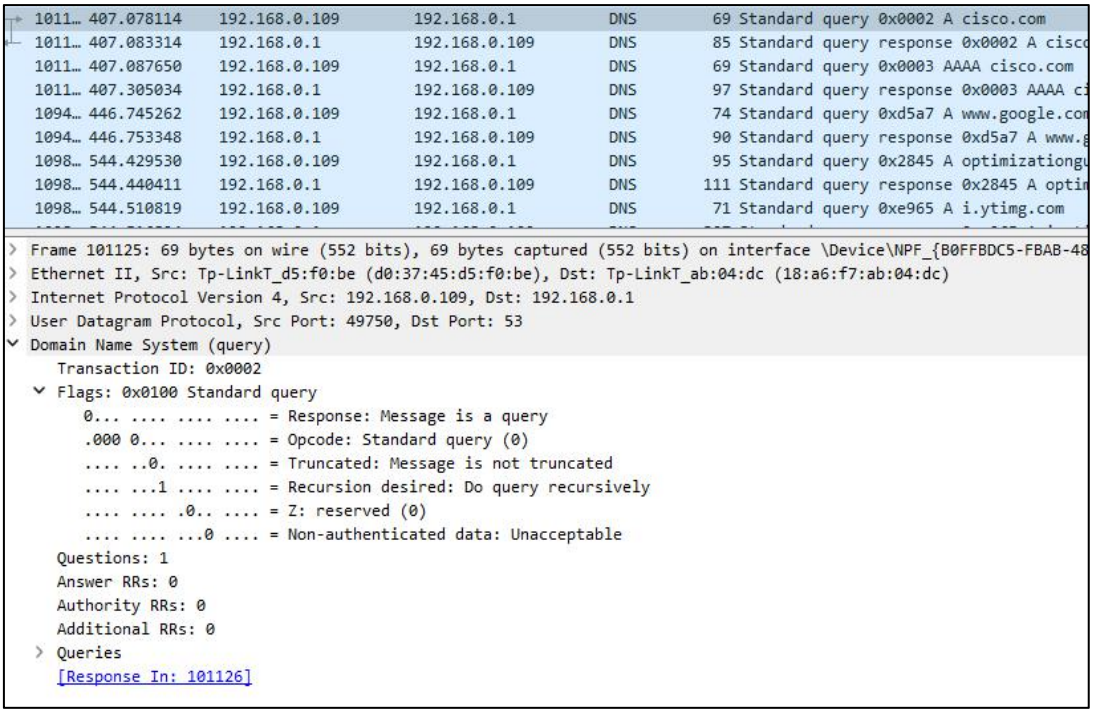
```
> Ethernet II, Src: Tp-Link1_d5:f0:be (d0:37:45:d5:f0:be), Dst: Tp-Link1_ab:04:dc
> Internet Protocol Version 4, Src: 192.168.0.109, Dst: 192.168.0.1
> User Datagram Protocol, Src Port: 49750, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x0002
  v Flags: 0x0100 Standard query
      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**.



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

"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="immark")  # provides --MARK-- message capability

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

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

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

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"

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
[sudo] password for kali:
Ign:1 http://ftp.harukasan.org/kali kali-rolling InRelease
Ign:1 http://ftp.harukasan.org/kali kali-rolling InRelease
Ign:1 http://ftp.harukasan.org/kali kali-rolling InRelease
```

"sudo apt-get install rsyslog"

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install rsyslog
Reading package lists ... Done
Building dependency tree ... Done
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 1019 not upgraded.
Need to get 0 B/727 kB of archives.
After this operation, 1,981 kB of additional disk space will be used.
Selecting previously unselected package rsyslog.
```

Step 3: Open rsyslog configuration file

"sudo nano /etc/rsyslog.conf"

```
#
# Set the default permissions for all log files.
#
$FileOwner root
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PreserveFQDN on
#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
```

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

```
        cron,daemon.none;\
        mail.none                -/var/log/messages

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

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

^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify
```

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

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

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

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

^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify
```

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/"

```
ubuntu@ubuntu2004:~$ ls /var/log/
alternatives.log  dmesg           gdm3             private
apt               dmesg.0         gpu-manager.log  remotelogs
auth.log          dmesg.1.gz      hp               speech-dispatcher
boot.log          dmesg.2.gz      installer        syslog
boot.log.1        dmesg.3.gz      journal          syslog.1
bootstrap.log     dmesg.4.gz      kali             ubuntu2004
btmp              dpkg.log        kern.log         ubuntu-advantage.log
cups              faillog         lastlog          unattended-upgrades
dist-upgrade      fontconfig.log  openvpn          wtmp
```

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:

```
ubuntu@ubuntu2004:~$ sudo ls /var/log/kali
CRON.log  rsyslogd.log
```

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
2022-05-18T05:52:21-04:00 kali rsyslogd: [origin software="rsyslogd" swVersion="
8.2204.0" x-pid="8621" x-info="https://www.rsyslog.com"] exiting on signal 15.
2022-05-18T05:52:21-04:00 kali rsyslogd: imuxsock: Acquired UNIX socket '/run/sy
stemd/journal/syslog' (fd 3) from systemd.  [v8.2204.0]
2022-05-18T05:52:21-04:00 kali rsyslogd: [origin software="rsyslogd" swVersion="
8.2204.0" x-pid="9917" x-info="https://www.rsyslog.com"] start
2022-05-18T05:52:21-04:00 kali rsyslogd: [origin software="rsyslogd" swVersion="
8.2204.0" x-pid="8621" x-info="https://www.rsyslog.com"] exiting on signal 15.
2022-05-18T05:52:21-04:00 kali rsyslogd: imuxsock: Acquired UNIX socket '/run/sy
stemd/journal/syslog' (fd 3) from systemd.  [v8.2204.0]
2022-05-18T05:52:21-04:00 kali rsyslogd: [origin software="rsyslogd" swVersion="
8.2204.0" x-pid="9917" x-info="https://www.rsyslog.com"] start
```

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"

```
ubuntu@ubuntu2004:/tmp$ cd /tmp && wget https://download.splunk.com/products/sp
lunk/releases/7.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb
--2022-05-17 03:57:43--  https://download.splunk.com/products/splunk/releases/7
.1.1/linux/splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb
Resolving download.splunk.com (download.splunk.com)... 18.66.78.115, 18.66.78.1
7, 18.66.78.30, ...
Connecting to download.splunk.com (download.splunk.com)|18.66.78.115|:443... co
nnected.
HTTP request sent, awaiting response... 200 OK
Length: 263297630 (251M) [binary/octet-stream]
Saving to: 'splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.deb'

splunk-7.1.1-8f0ead 100%[===================>] 251.10M  10.6MB/s    in 24s
```

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

```
ubuntu@ubuntu2004:/tmp$ sudo dpkg -i splunk-7.1.1-8f0ead9ec3db-linux-2.6-amd64.
deb
Selecting previously unselected package splunk.
(Reading database ... 145742 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@ubuntu2004:/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 SPLUN
K
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 OTHE
R
ENTITY FOR WHICH YOU ARE ACTING, ON BEHALF OF YOURSELF AS AN INDIVIDUAL; AND (B
```

```
Splunk Software License Agreement 04.24.2018

Do you agree with this license? [y/n]: y

This appears to be your first time running this version of Splunk.

An Admin password must be set before installation proceeds.
Password must contain at least:
   * 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/sh
are/splunk/search_mrsparkle/modules'.
Init script installed at /etc/init.d/splunk.
Init script is configured to run at boot.
ubuntu@ubuntu2004:/tmp$
```

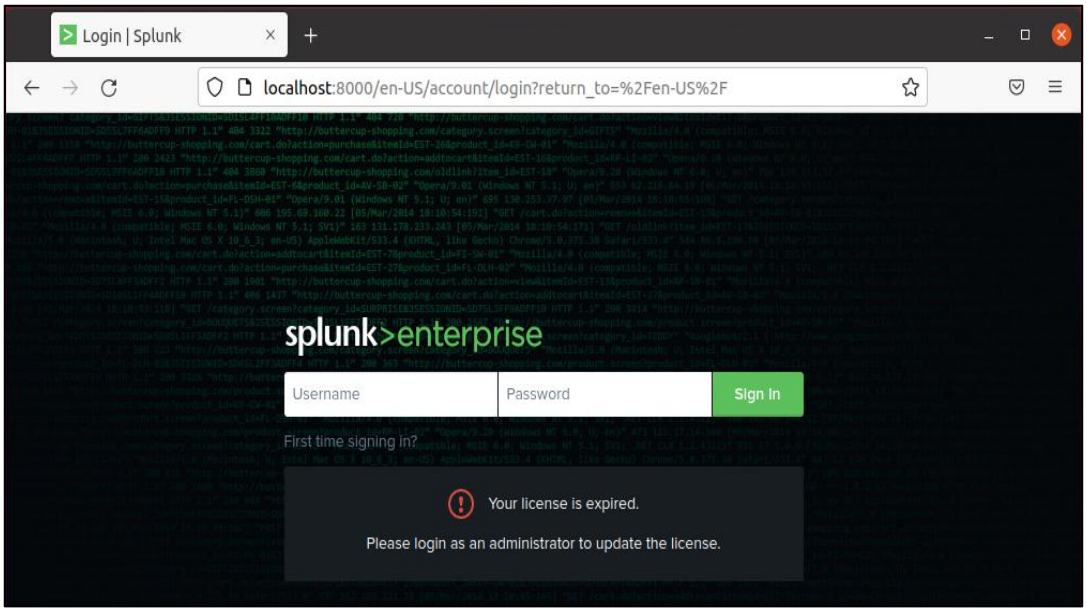
Step 4: Start the Splunk service
"sudo service splunk start"

```
ubuntu@ubuntu2004:/tmp$ sudo service splunk start
```

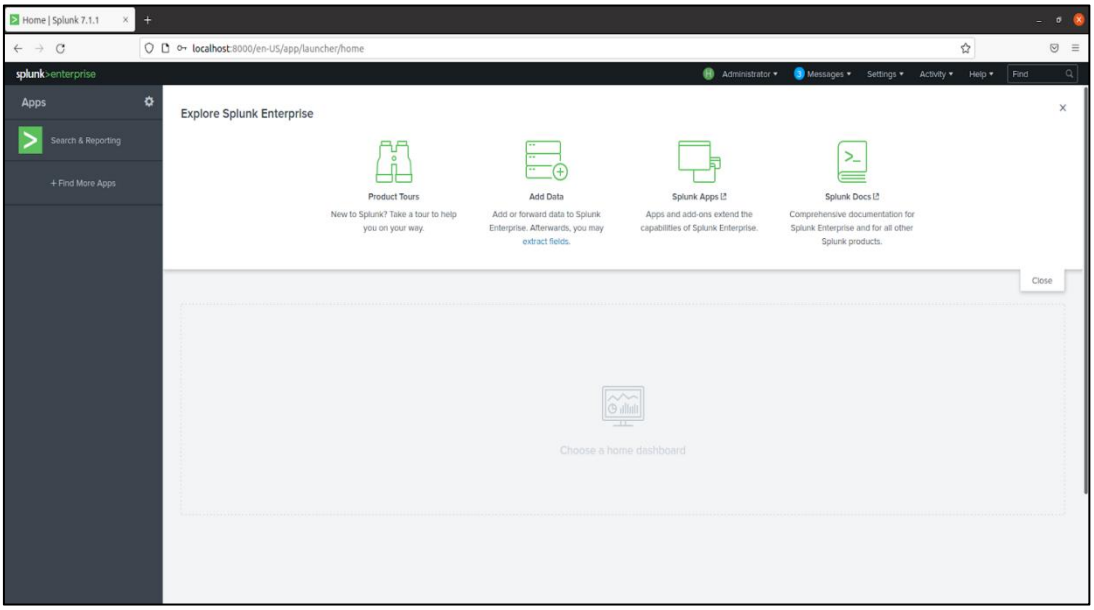
Step 5: Check splunk service Status
"sudo service splunk status"

```
ubuntu@ubuntu2004:/tmp$ sudo service splunk status
● splunk.service - LSB: Start splunk
     Loaded: loaded (/etc/init.d/splunk; generated)
     Active: active (running) since Tue 2022-05-17 04:29:03 EDT; 37min ago
       Docs: man:systemd-sysv-generator(8)
    Process: 2901 ExecStart=/etc/init.d/splunk start (code=exited, status=0/SUCC>
      Tasks: 161 (limit: 2290)
     Memory: 346.7M
     CGroup: /system.slice/splunk.service
             ├─2964 splunkd -p 8089 start
             ├─2965 [splunkd pid=2964] splunkd -p 8089 start [process-runner]
             ├─2977 mongod --dbpath=/opt/splunk/var/lib/splunk/kvstore/mongo --s>
             ├─3034 /opt/splunk/bin/python -O /opt/splunk/lib/python2.7/site-pac>
             └─3074 /opt/splunk/bin/splunkd instrument-resource-usage -p 8089 -->
```

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 "**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"

```
done.
done.
ubuntu@ubuntu2004:~$ java -version
openjdk version "11.0.15" 2022-04-19
OpenJDK Runtime Environment (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1)
OpenJDK 64-Bit Server VM (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1, mixed mode,
sharing)
ubuntu@ubuntu2004:~$
```

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"

```
ubuntu@ubuntu2004:~$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-k
ey add -
OK
ubuntu@ubuntu2004:~$ 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
deb https://artifacts.elastic.co/packages/oss-6.x/apt stable main
ubuntu@ubuntu2004:~$
```

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.

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

```
#
# For more information, consult the gateway module documentation.
#
# --------------------------------- Various ----------------------------------
#
# Require explicit names when deleting indices:
#
#action.destructive_requires_name: true
action.auto_create_index: false

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line
```

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

"sudo systemctl daemon-reload"

"sudo systemctl start elasticsearch"

"sudo systemctl enable elasticsearch"

```
ubuntu@ubuntu2004:~$ sudo nano /etc/elasticsearch/elasticsearch.yml
ubuntu@ubuntu2004:~$ sudo nano /etc/elasticsearch/elasticsearch.yml
ubuntu@ubuntu2004:~$ sudo systemctl daemon-reload
ubuntu@ubuntu2004:~$ sudo systemctl start elasticsearch
ubuntu@ubuntu2004:~$ 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.service → /lib/system
d/system/elasticsearch.service.
ubuntu@ubuntu2004:~$
```

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 "

```
ubuntu@ubuntu2004:~$ curl -X GET http://localhost:9200
{
  "name" : "bOUPu21",
  "cluster_name" : "graylog",
  "cluster_uuid" : "cX23IyphTwSlMhz_0Lqf7w",
  "version" : {
    "number" : "6.8.23",
    "build_flavor" : "oss",
    "build_type" : "deb",
    "build_hash" : "4f67856",
    "build_date" : "2022-01-06T21:30:50.087716Z",
    "build_snapshot" : false,
    "lucene_version" : "7.7.3",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
ubuntu@ubuntu2004:~$
```

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.
"sudo systemctl start mongodb"
"sudo systemctl 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"

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

```
ubuntu@ubuntu2004:~$ sudo dpkg -i /home/ubuntu/Downloads/graylog-3.3-repository_latest.deb
Selecting previously unselected package graylog-3.3-repository.
(Reading database ... 146508 files and directories currently installed.)
Preparing to unpack .../graylog-3.3-repository_latest.deb ...
Unpacking graylog-3.3-repository (1-1) ...
Setting up graylog-3.3-repository (1-1) ...
ubuntu@ubuntu2004:~$ sudo apt update
```

Step 2: Update the repository cache. "sudo apt update"

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

"sudo dpkg -i graylog-server"

```
root@ubuntu2004:/home/ubuntu/Downloads# sudo dpkg -i graylog-server_3.3.16-2_all.deb
Selecting previously unselected package graylog-server.
(Reading database ... 188779 files and directories currently installed.)
Preparing to unpack graylog-server_3.3.16-2_all.deb ...
Unpacking graylog-server (3.3.16-2) ...
Setting up graylog-server (3.3.16-2) ...
Processing triggers for systemd (245.4-4ubuntu3.17) ...
```

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"

```
root@ubuntu2004:/home/ubuntu/Downloads# pwgen -N 1 -s 96
uUKugUCKcLdImgd0W0o4pEUivxaiv6GHGcW7JGMBZnmm1vYh3rp3pqSN34hCqDbdUDnfZHlFec4uiu39auGdIqSz0K7RfVeg
root@ubuntu2004:/home/ubuntu/Downloads#
```

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

Then, place the secret like below.

```
# You MUST set a secret to secure/pepper the stored user passwords here. Use at least 64 characters.
# 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 encrypted values in the database invalid. (e.g.
  encrypted access tokens)
password_secret =uUKugUCKcLdImgd0W0o4pEUivxaiv6GHGcW7JGMBZnmm1vYh3rp3pqSN34hCqDbdUDnfZHlFec4uiu39auGdIqSz0K7RfVeg

# The default root user is named 'admin'
#root_username = admin

# You MUST specify a hash password for the root user (which you only need to initially set up the
```

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"

```
(gedit:45358): Tepl-WARNING **: 05:09:20.748: GVfs metadata is not supported. Fallback to TeplMetadataM
er GVfs is not correctly installed or GVfs metadata are not supported on this platform. In the latter c
uld configure Tepl with --disable-gvfs-metadata.
root@ubuntu2004:/home/ubuntu/Downloads# echo -n password | sha256sum
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8  -
```

Step 7: Edit the server.conf file again.in terminal
"sudo nano /etc/graylog/server/server.conf"

```
# This password cannot be changed using the API or via the web interface. If you need to c
# modify it in this file.
# Create one by using for example: echo -n yourpassword | shasum -a 256
# and put the resulting hash value into the following line
root_password_sha2 = 5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
# The email address of the root user.
# Default is empty
#root_email = ""
```

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/

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@ubuntu2004:~/Downloads$ sudo systemctl daemon-reload
ubuntu@ubuntu2004:~/Downloads$ sudo systemctl start graylog-server
ubuntu@ubuntu2004:~/Downloads$ 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.service → /lib/systemd/system/graylog-se
rver.service.
ubuntu@ubuntu2004:~/Downloads$
```

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

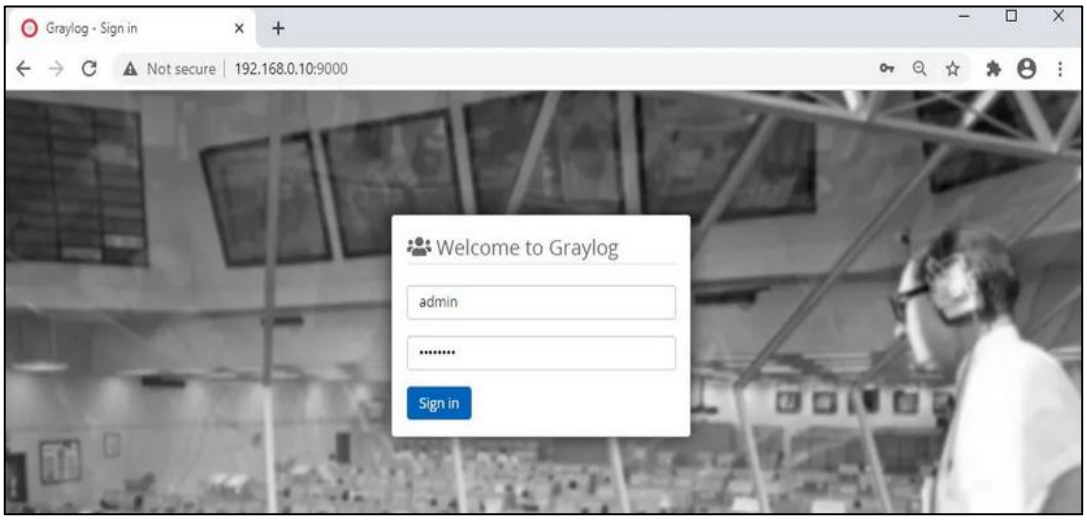
“sudo tail -f /var/log/graylog-server/server.log”

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 and point it to.

“**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"**

```
Terminal - analyst@secOps:~/lab.support.files
File  Edit  View  Terminal  Tabs  Help
[analyst@secOps lab.support.files]$ ls -l
total 584
-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     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
-rw-r--r-- 1 analyst analyst    255 Mar 21  2018 letter_to_grandma.txt
-rw-r--r-- 1 analyst analyst  24464 Mar 21  2018 logstash-tutorial.log
```

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

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

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
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" 2
00 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.

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

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"

```
[analyst@secOps lab.support.files]$  awk 'BEGIN {FS=OFS=" "} {gsub(/[|]/,"",$4)}{print}{$4=strftime("%c",$4)}{print}' apach
e_in_epoch.log
198.51.100.213 - - [1219071600] "GET /twiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP
/1.1" 401 12846
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.Configur
ationVariables HTTP/1.1" 401 12846
198.51.100.213 - - [1219158000] "GET /twiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1" 200 4523
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1"
200 4523
198.51.100.213 - - [1220799600] "GET /mailman/listinfo/hsdivision HTTP/1.1" 200 6291
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /mailman/listinfo/hsdivision HTTP/1.1" 200 6291
198.51.100.213 - - [1220886000] "GET /twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
198.51.100.213 - - [1220972400] "GET /twiki/bin/view/Main/DCCAndPostFix HTTP/1.1" 200 5253
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/view/Main/DCCAndPostFix HTTP/1.1" 200 5253
198.51.100.213 - - [1221058800] "GET /twiki/bin/oops/TWiki/AppendixFileSystem?template=oopsmore&m1=1.12&m2=1.12 HTTP/1.1" 2
00 11382
198.51.100.213 - - Wed 31 Dec 1969 07:00:00 PM EST "GET /twiki/bin/oops/TWiki/AppendixFileSystem?template=oopsmore&m1=1.12&
m2=1.12 HTTP/1.1" 200 11382
[analyst@secOps lab.support.files]$
```