Index

List of Practical		Date	Signature
1.	Encrypting and Decrypting Data Using OpenSSL	20/2/23	
2.	Demonstrate the use of Snort and Firewall Rules.	1/3/23	
3.	Demonstrate Extract an Executable from a PCAP.	9/3/23	
4.	Demonstrate Analysis of DNS Traffic	18/3/23	
5.	Create your own syslog server	21/3/23	
6.	Install and Run Splunk on Linux	3/4/23	
7.	Install and Configure ELK on Linux	10/4/23	
8.	Install and Configure GrayLog on Linux	17/4/23	

Table of Contents

Practical 1: Encrypting and Decrypting Data Using OpenSSL

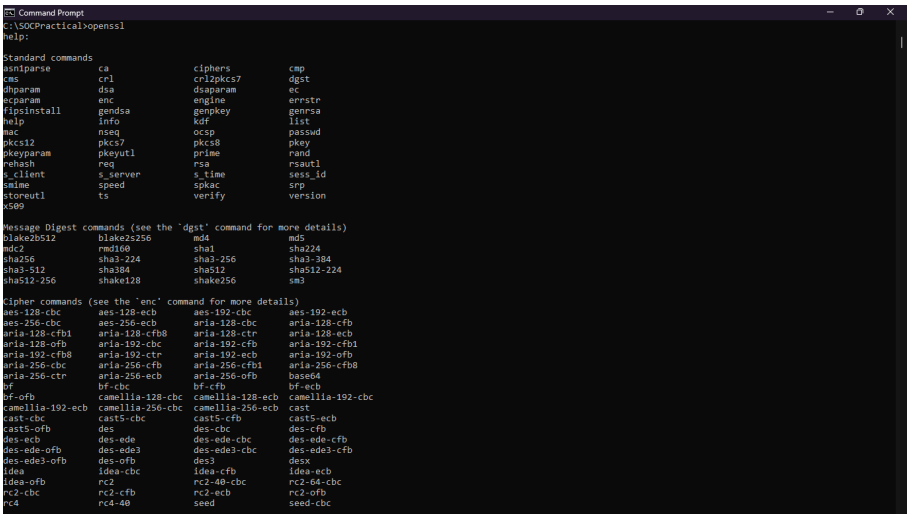


Figure 1 - openssl commands

Create a text file named input.txt and add some text or data in it. Encrypt the file with file command. After encrypting the file, a .enc file will be generated. And decrypt it into another file and name it output.txt.

```
C:\SOCPractical>openssl enc -aes-256-ecb -a -salt -in input.txt -out cipher.enc
enter AES-256-ECB encryption password:
Verifying - enter AES-256-ECB encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\SOCPractical>openssl enc -d -aes-256-ecb -a -salt -in cipher.enc -out output.txt
enter AES-256-ECB decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\SOCPractical>_
```

Figure 2 - Encryption and decryption using aes 256

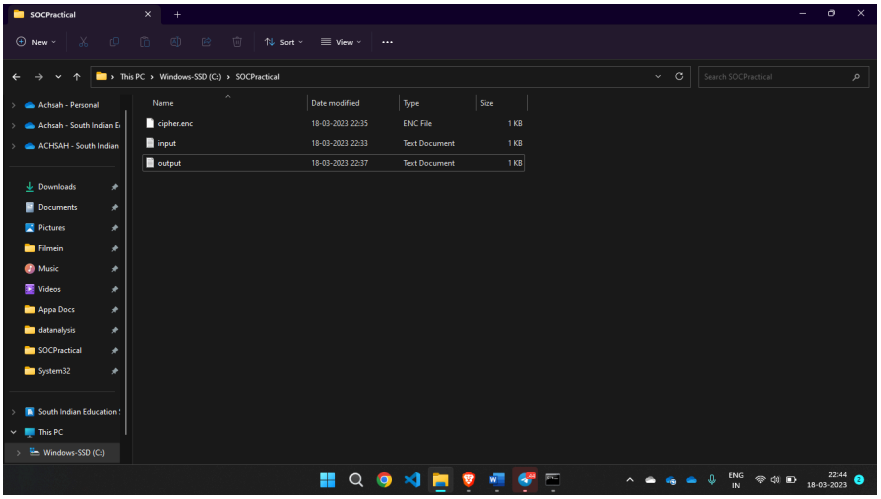


Figure 3 - all files generated by openssl

Practical 2: Demonstrate the use of Snort and Firewall Rules

Step-1-> Download snort application from https://snort.org/downloads from binaries section Snort_2_9_20_Installer.x64.exe

Step-2-> Download rules tar file from the registered section snortrules-snapshot-29200.tar.gz.

Step-3-> Install snort.exe file

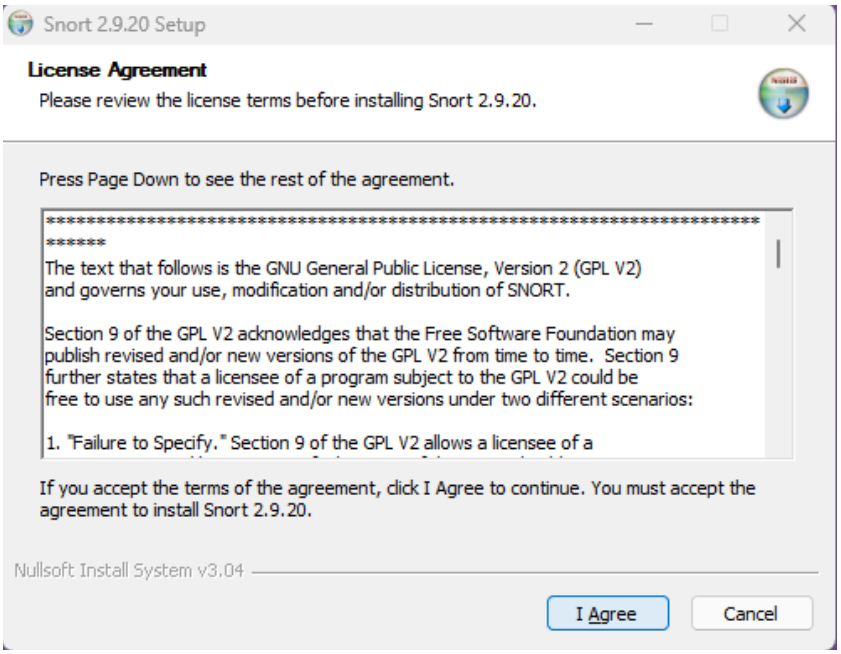


Figure 4 - Snort installation

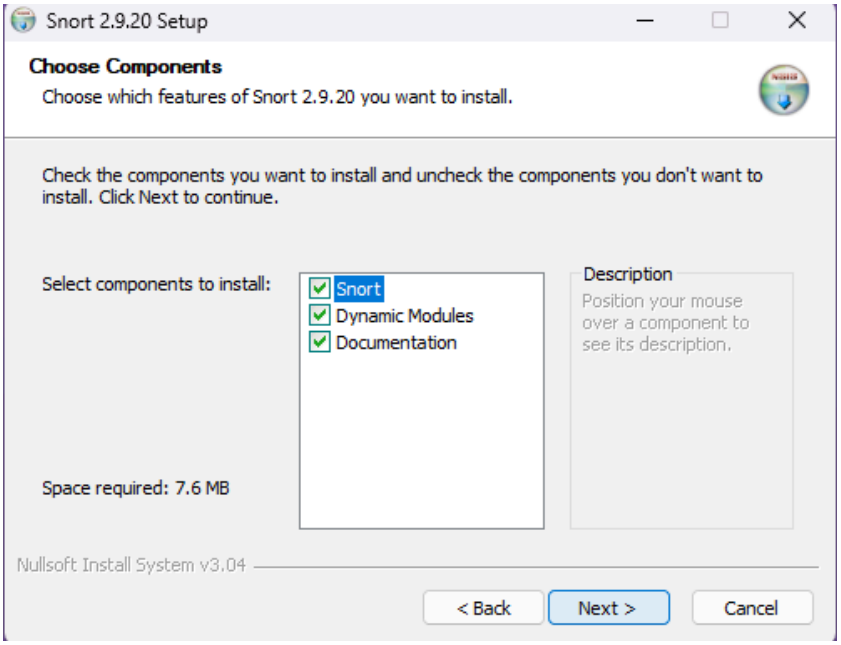


Figure 5 - choose all options

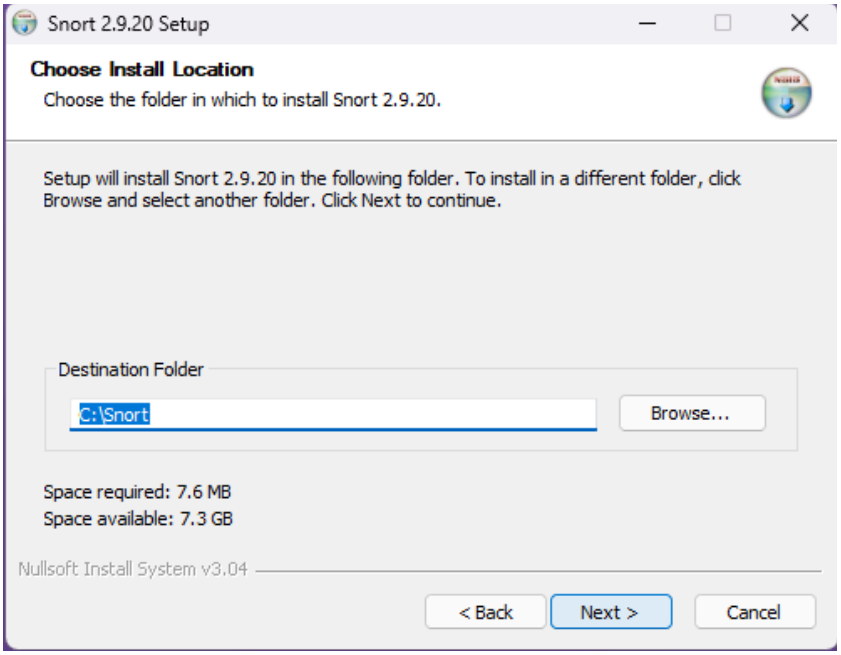


Figure 6 - Destination folder

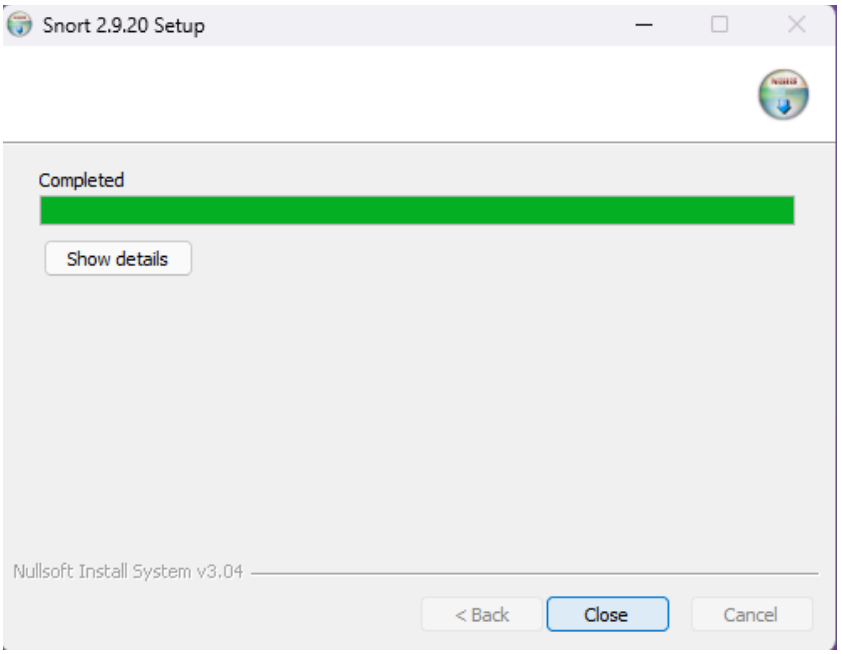


Figure 7 - installation succesful

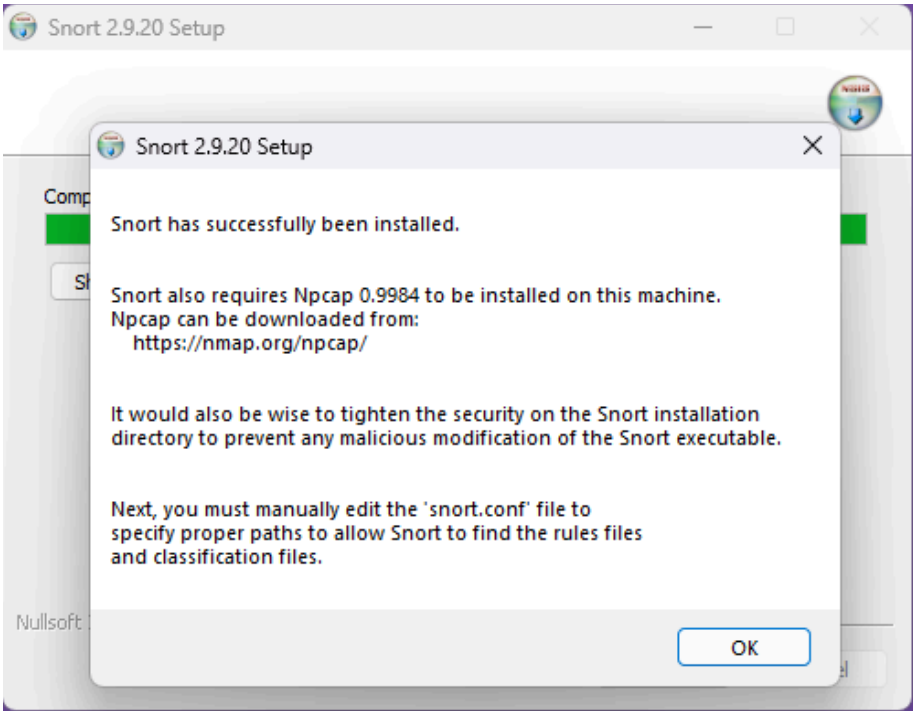


FIGURE 8 - NPCAP REQUIREMENT MESSAGE

Step-4-> Unzip and extract the .tar file

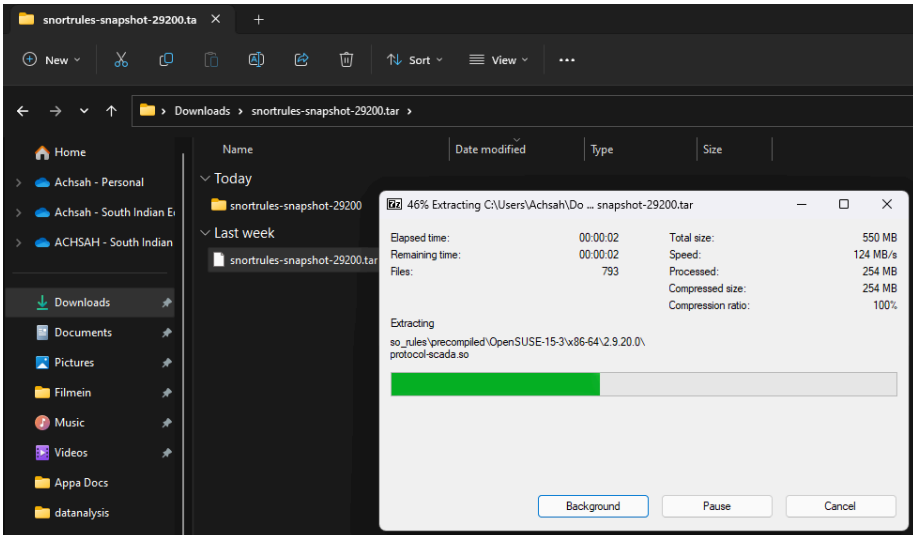


FIGURE 9 - EXTRACTING RULES ARCHIVE

Step-5-> Copy rules and preproc_rules folder from the extracted folder and paste it to the Snort folder in C:\Snort thus replacing the existing folder in the same.

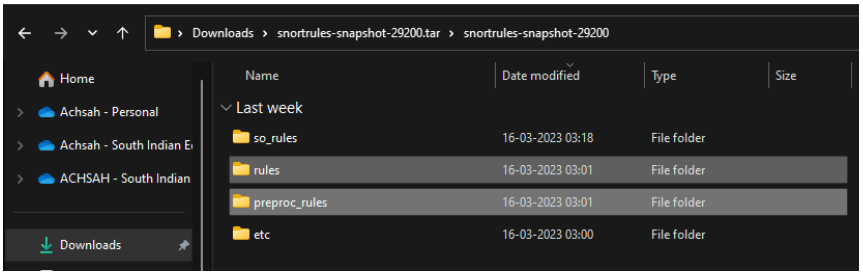


FIGURE 10 - UNZIPPED ARCHIVE

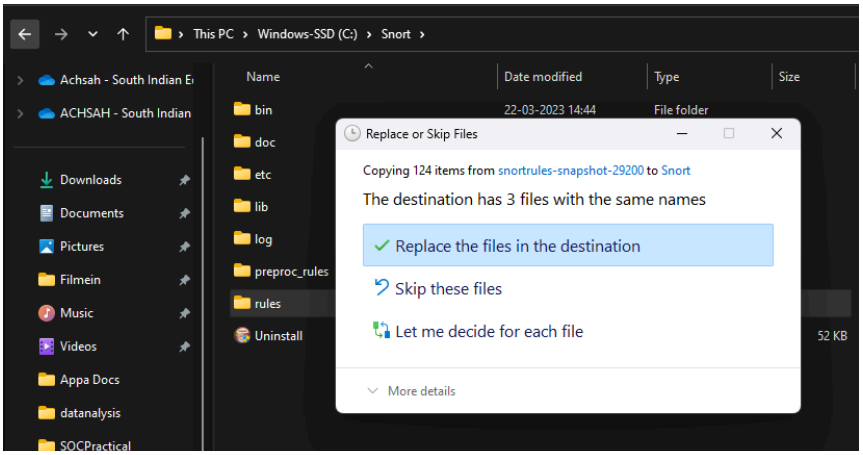


FIGURE 11 - REPLACING PREVIOUS FILES

Step-6-> Download and install notepad++ from https://notepad-plus-plus.org/downloads/ and open the snort.conf file from C:\Snort\etc in the same.

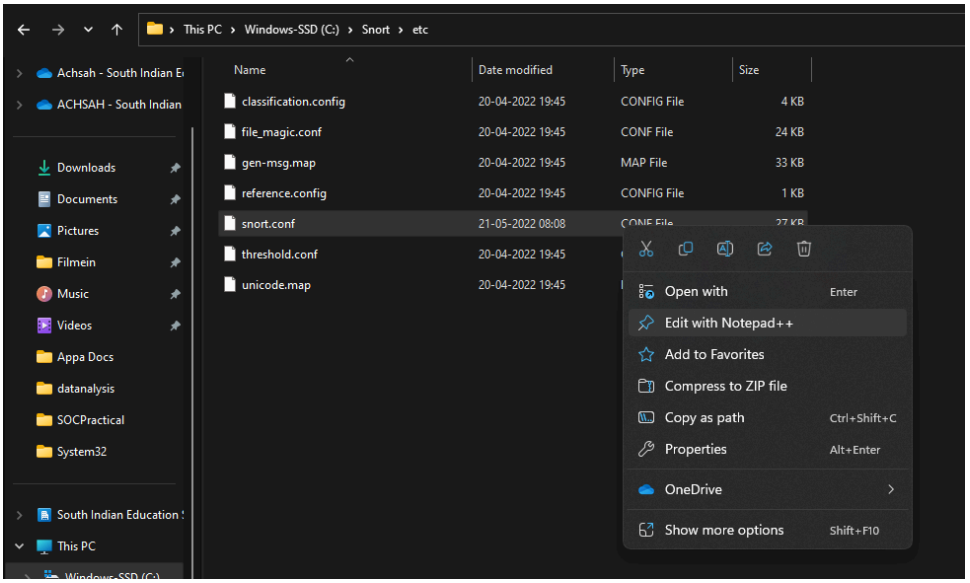


FIGURE 12 - EDIT SNORT.CONF WITH NOTEPAD++

Step-7-> Search for blacklist from the snort folder in C:\Snort, open in notepad++, do save as and name it to whitelist.rules now you will have files with name blacklist.rules and whitelist.rules in the folder.

Step-8-> Edit the config file in the following lines

```
44  # Setup the network addresses you a
45  ipvar HOME_NET 192.168.0.0/24
46
47  # Set up the external network addre
48  ipvar EXTERNAL_NET !$HOME_NET
49
```

```
104 var RULE_PATH C:\Snort\rules
105 # var SO_RULE_PATH ../so_rules
106 var PREPROC_RULE_PATH C:\Snort\preproc_rules
```

```
112 # Set the absolute path appropriately
113 var WHITE_LIST_PATH C:\Snort\rules
114 var BLACK_LIST_PATH C:\Snort\rules
115
```

```
185 #
186 config logdir: C:\Snort\log
187
```

```
246 # path to dynamic preprocessor libraries
247 dynamicpreprocessor directory C:\Snort\lib\snort_dynamicpreprocessor
248
249 # path to base preprocessor engine
250 dynamicengine C:\Snort\lib\snort_dynamicengine\sf_engine.dll
251
252 # path to dynamic rules libraries
253 # dynamicdetection directory /usr/local/lib/snort_dynamicrules
254
```

```
264 # Does nothing in IDS mode
265 # preprocessor normalize_ip4
266 # preprocessor normalize_tcp: ips ecn stream
267 # preprocessor normalize_icmp4
268 # preprocessor normalize_ip6
269 # preprocessor normalize_icmp6
270
```

Step-9-> From line no. 546 to 651 replace forward slash to backslash of the path given.

```
# site specific rules
include $RULE_PATH\local.rules

include $RULE_PATH\app-detect.rules
include $RULE_PATH\attack-responses.rules
include $RULE_PATH\backdoor.rules
include $RULE_PATH\bad-traffic.rules
include $RULE_PATH\blacklist.rules
include $RULE_PATH\botnet-cnc.rules
include $RULE_PATH\browser-chrome.rules
include $RULE_PATH\browser-firefox.rules
include $RULE_PATH\browser-ie.rules
include $RULE_PATH\browser-other.rules
include $RULE_PATH\browser-plugins.rules
include $RULE_PATH\browser-webkit.rules
include $RULE_PATH\chat.rules
include $RULE_PATH\content-replace.rules
include $RULE_PATH\ddos.rules
include $RULE_PATH\dns.rules
include $RULE_PATH\dos.rules
include $RULE_PATH\experimental.rules
include $RULE_PATH\exploit-kit.rules
include $RULE_PATH\exploit.rules
include $RULE_PATH\file-executable.rules
include $RULE_PATH\file-flash.rules
include $RULE_PATH\file-identify.rules
include $RULE_PATH\file-image.rules
include $RULE_PATH\file-multimedia.rules
include $RULE_PATH\file-office.rules
include $RULE_PATH\file-other.rules
```

```

# decoder and preprocessor event rules
include $PREPROC_RULE_PATH/preprocessor.rules
include $PREPROC_RULE_PATH/decoder.rules
include $PREPROC_RULE_PATH/sensitive-data.rules

```

Step-10-> Open local.rules from C:\Snort\rules in notepad++ and add the following lines of code in it.

```

alert icmp any any -> any any (msg:"Testing ICMP"; sid:1000001;)

alert udp any any -> any any (msg:"Testing UDP"; sid:1000002;)

alert tcp any any -> any any (msg:"Testing TCP"; sid:1000003;)
```

Step-11-> Open command line in the path C:\Snort\bin and run the command snort -W

```
C:\Snort\bin>snort -W

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Index   Physical Address        IP Address      Device Name     Description
-----   ----------------        ----------      -----------     -----------
    1   00:00:00:00:00:00       disabled        \Device\NPF_{70B1D23C-CB1B-4DA2-AACE-6053BA239C62}      WAN Miniport (Network Monitor)
    2   00:00:00:00:00:00       disabled        \Device\NPF_{7D41CBC9-C450-4414-9246-4557C6AF3787}      WAN Miniport (IPv6)
    3   00:00:00:00:00:00       disabled        \Device\NPF_{A02C2DF4-6A26-47AB-841F-EFA013BD4B4D}      WAN Miniport (IP)
    4   E8:D0:FC:E1:EF:50       169.254.39.155  \Device\NPF_{5F24DE34-CF9E-468C-91A3-7FCE4440E6CF}      Bluetooth Device (Personal Area Network)
    5   E8:D0:FC:E1:EF:4F       192.168.0.102   \Device\NPF_{8DD3EE5E-90F0-45E9-B649-C54C390F5069}      Qualcomm Atheros QCA9377 Wireless Network Adapter
    6   FA:D0:FC:E1:EF:4F       169.254.184.44  \Device\NPF_{05E7CDB5-AC63-4D2C-B689-73EB8461E09B}      Microsoft Wi-Fi Direct Virtual Adapter #4
    7   EA:D0:FC:E1:EF:4F       169.254.7.234   \Device\NPF_{FFDD3A86-D792-4D13-A9AF-599631C8FBBA}      Microsoft Wi-Fi Direct Virtual Adapter #3
    8   00:00:00:00:00:00       0000:0000:0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback    Adapter for loopback traffic capture

C:\Snort\bin>
```

FIGURE 13 - WIRELESS ADAPTERS

Step-12-> Run the command: snort -i 5 -c C:\Snort\etc\snort.conf -A console

```
C:\Windows\System32\cmd.exe
C:\Snort\bin>snort -i 5 -c C:\Snort\etc\snort.conf -A console
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "C:\Snort\etc\snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
PortVar 'SSH_PORTS' defined :  [ 22 ]
PortVar 'FTP_PORTS' defined :  [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined :  [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined :  [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined :  [ 2123 2152 3386 ]
Detection:
   Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method-Optimizations = enabled
    Maximum pattern length = 20
Tagged Packet Limit: 256
Loading dynamic engine C:\Snort\lib\snort_dynamicengine\sf_engine.dll... done
Loading all dynamic preprocessor libs from C:\Snort\lib\snort_dynamicpreprocessor...
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dce2.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dnp3.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dns.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_ftptelnet.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_gtp.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imap.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_modbus.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_pop.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_reputation.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_sdf.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_sip.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_smtp.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_ssh.dll... done
  Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_ssl.dll... done
  Finished Loading all dynamic preprocessor libs from C:\Snort\lib\snort_dynamicpreprocessor
Log directory = C:\Snort\log
Frag3 global config:
    Max frags: 65536
    Fragment memory cap: 4194304 bytes
```

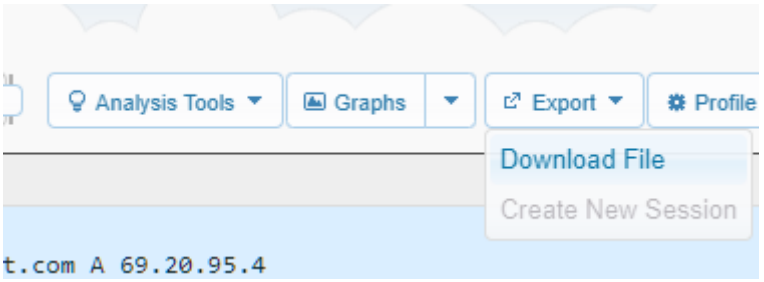
```
03/22-16:38:50.119481  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.121941  [**] [1:10000003:0] Testing TCP [**] [Priority: 0] {TCP} 192.168.0.102:36713 -> 35.223.238.178:443
03/22-16:38:50.136726  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:63398 -> 35.207.209.40:50009
```

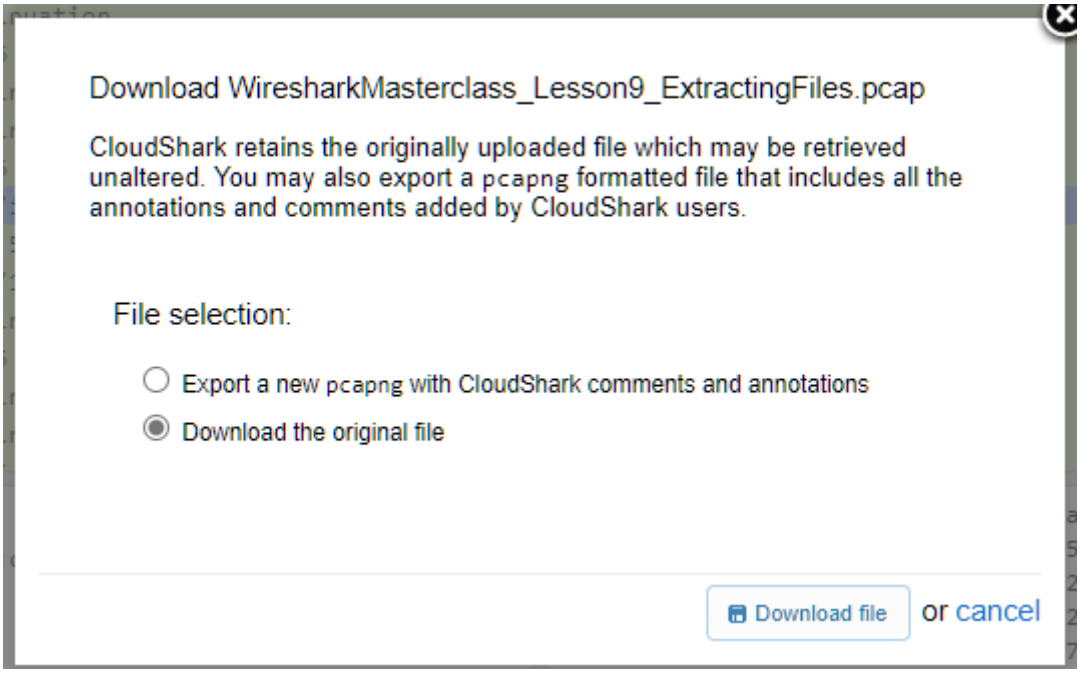
```
C:\Windows\System32\cmd.exe
03/22-16:38:50.774351  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.774437  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.774516  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.774564  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.774648  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.774779  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.774840  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.774930  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.775018  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.775086  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.776983  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:63398 -> 35.207.209.40:50009
03/22-16:38:50.777260  [**] [1:10000001:0] Testing ICMP [**] [Priority: 0] {IPV6-ICMP} fe80:0000:0000:0000:0000:ea48:b8ff:fec3:8930 -> ff02:0000:0000:0000:0000:0000:0000:000
1
03/22-16:38:50.777260  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 35.207.209.40:50009 -> 192.168.0.102:63398
03/22-16:38:50.794425  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 35.207.209.40:50009 -> 192.168.0.102:63398
03/22-16:38:50.796745  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:63398 -> 35.207.209.40:50009
03/22-16:38:50.809704  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 66.22.239.4:50007 -> 192.168.0.102:56570
03/22-16:38:50.809704  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 66.22.239.4:50007 -> 192.168.0.102:56570
03/22-16:38:50.809704  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 66.22.239.4:50007 -> 192.168.0.102:56570
03/22-16:38:50.812264  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 35.207.209.40:50009 -> 192.168.0.102:63398
03/22-16:38:50.816585  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:63398 -> 35.207.209.40:50009
03/22-16:38:50.819367  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.819823  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.819905  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.820020  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.820123  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.820251  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.820365  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.824960  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 66.22.239.4:50007 -> 192.168.0.102:56570
03/22-16:38:50.829846  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 35.207.209.40:50009 -> 192.168.0.102:63398
03/22-16:38:50.839988  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:63398 -> 35.207.209.40:50009
03/22-16:38:50.857744  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:63398 -> 35.207.209.40:50009
03/22-16:38:50.858706  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 35.207.209.40:50009 -> 192.168.0.102:63398
03/22-16:38:50.863512  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.863779  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.863870  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.863929  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.863987  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.864043  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.864106  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.864177  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.864228  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:56570 -> 66.22.239.4:50007
03/22-16:38:50.870796  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 35.207.209.40:50009 -> 192.168.0.102:63398
03/22-16:38:50.877059  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:63398 -> 35.207.209.40:50009
03/22-16:38:50.895213  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 35.207.209.40:50009 -> 192.168.0.102:63398
03/22-16:38:50.896940  [**] [1:10000002:0] Testing UDP [**] [Priority: 0] {UDP} 192.168.0.102:63398 -> 35.207.209.40:50009
```

FIGURE 14 - UDP TCP ICMP ALERTS TRIGGERED BY SNORT

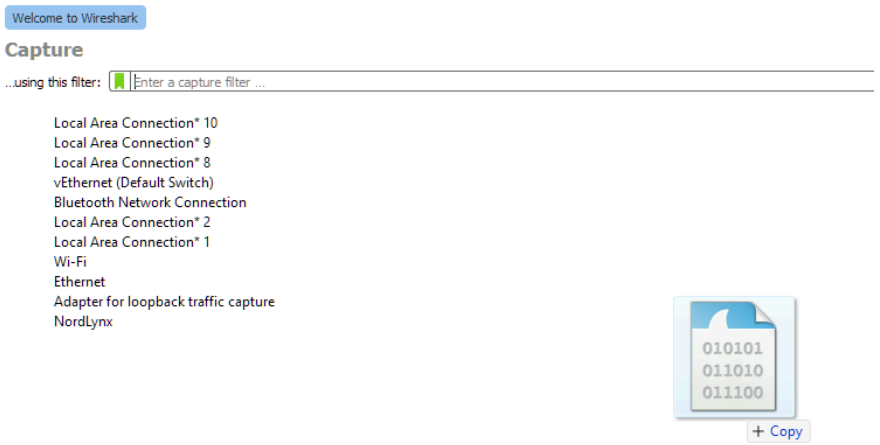
Practical 3: Demonstrate Extract an Executable from a PCAP

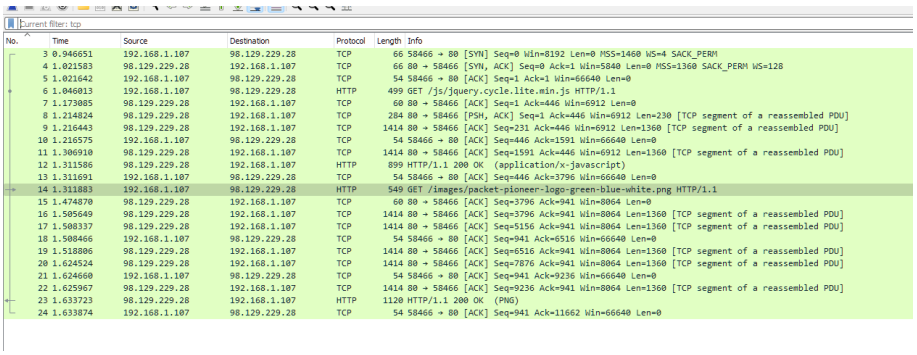
Step-1-> Download packet capture file from : https://www.cloudshark.org/captures/a9472fbe700a





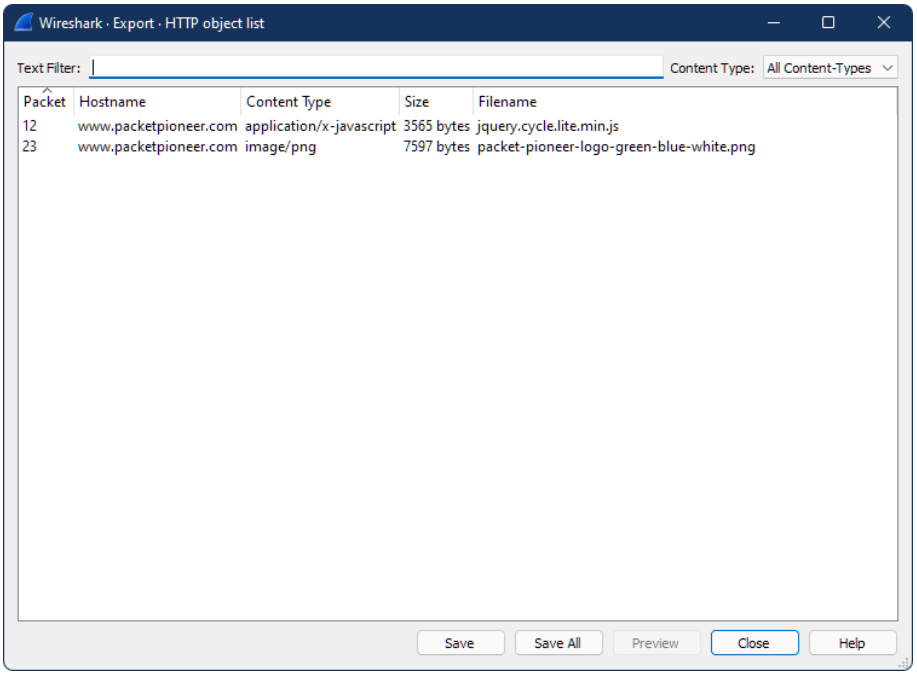
Step-2-> Open the packet capture file in Wireshark by dragging and dropping the capture file into the Wireshark interface.



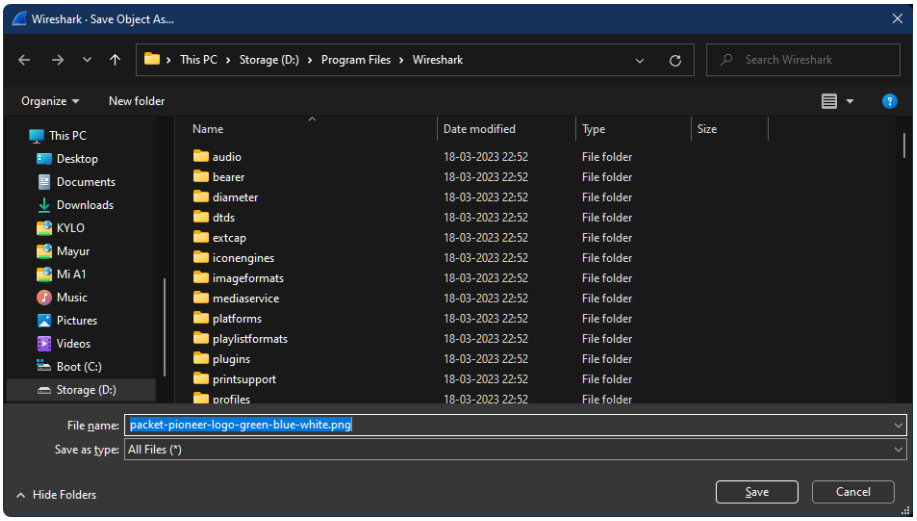


No.	Time	Source	Destination	Protocol	Length	Info
3	0.946651	192.168.1.107	98.129.229.28	TCP	66	58466 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
4	1.021583	98.129.229.28	192.168.1.107	TCP	66	80 → 58466 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1360 SACK_PERM WS=128
5	1.021642	192.168.1.107	98.129.229.28	TCP	54	58466 → 80 [ACK] Seq=1 Ack=1 Win=66640 Len=0
6	1.046013	192.168.1.107	98.129.229.28	HTTP	499	GET /js/jquery.cycle.lite.min.js HTTP/1.1
7	1.173085	98.129.229.28	192.168.1.107	TCP	60	80 → 58466 [ACK] Seq=1 Ack=446 Win=6912 Len=0
8	1.214824	98.129.229.28	192.168.1.107	TCP	284	80 → 58466 [PSH, ACK] Seq=1 Ack=446 Win=6912 Len=230 [TCP segment of a reassembled PDU]
9	1.216443	98.129.229.28	192.168.1.107	TCP	1414	80 → 58466 [ACK] Seq=231 Ack=446 Win=6912 Len=1360 [TCP segment of a reassembled PDU]
10	1.216575	192.168.1.107	98.129.229.28	TCP	54	58466 → 80 [ACK] Seq=446 Ack=1591 Win=66640 Len=0
11	1.306910	98.129.229.28	192.168.1.107	TCP	1414	80 → 58466 [ACK] Seq=1591 Ack=446 Win=6912 Len=1360 [TCP segment of a reassembled PDU]
12	1.311586	98.129.229.28	192.168.1.107	HTTP	899	HTTP/1.1 200 OK (application/x-javascript)
13	1.311691	192.168.1.107	98.129.229.28	TCP	54	58466 → 80 [ACK] Seq=446 Ack=3796 Win=66640 Len=0
14	1.311883	192.168.1.107	98.129.229.28	HTTP	549	GET /images/packet-pioneer-logo-green-blue-white.png HTTP/1.1
15	1.474870	98.129.229.28	192.168.1.107	TCP	60	80 → 58466 [ACK] Seq=3796 Ack=941 Win=8064 Len=0
16	1.505649	98.129.229.28	192.168.1.107	TCP	1414	80 → 58466 [ACK] Seq=3796 Ack=941 Win=8064 Len=1360 [TCP segment of a reassembled PDU]
17	1.508337	98.129.229.28	192.168.1.107	TCP	1414	80 → 58466 [ACK] Seq=5156 Ack=941 Win=8064 Len=1360 [TCP segment of a reassembled PDU]
18	1.508466	192.168.1.107	98.129.229.28	TCP	54	58466 → 80 [ACK] Seq=941 Ack=6516 Win=66640 Len=0
19	1.518806	98.129.229.28	192.168.1.107	TCP	1414	80 → 58466 [ACK] Seq=6516 Ack=941 Win=8064 Len=1360 [TCP segment of a reassembled PDU]
20	1.624524	98.129.229.28	192.168.1.107	TCP	1414	80 → 58466 [ACK] Seq=7876 Ack=941 Win=8064 Len=1360 [TCP segment of a reassembled PDU]
21	1.624660	192.168.1.107	98.129.229.28	TCP	54	58466 → 80 [ACK] Seq=941 Ack=9236 Win=66640 Len=0
22	1.625967	98.129.229.28	192.168.1.107	TCP	1414	80 → 58466 [ACK] Seq=9236 Ack=941 Win=8064 Len=1360 [TCP segment of a reassembled PDU]
23	1.633723	98.129.229.28	192.168.1.107	HTTP	1120	HTTP/1.1 200 OK (PNG)
24	1.633874	192.168.1.107	98.129.229.28	TCP	54	58466 → 80 [ACK] Seq=941 Ack=11662 Win=66640 Len=0

Step-3-> Click on File->Export Objects-> HTTP



Step-4-> Select image/png and click on Save. Select Save Location.



Practical 4: Demonstrate Analysis of DNS Traffic

Step-1-> Open Wireshark, select Wi-Fi and click on Capture to capture the DNS Traffic

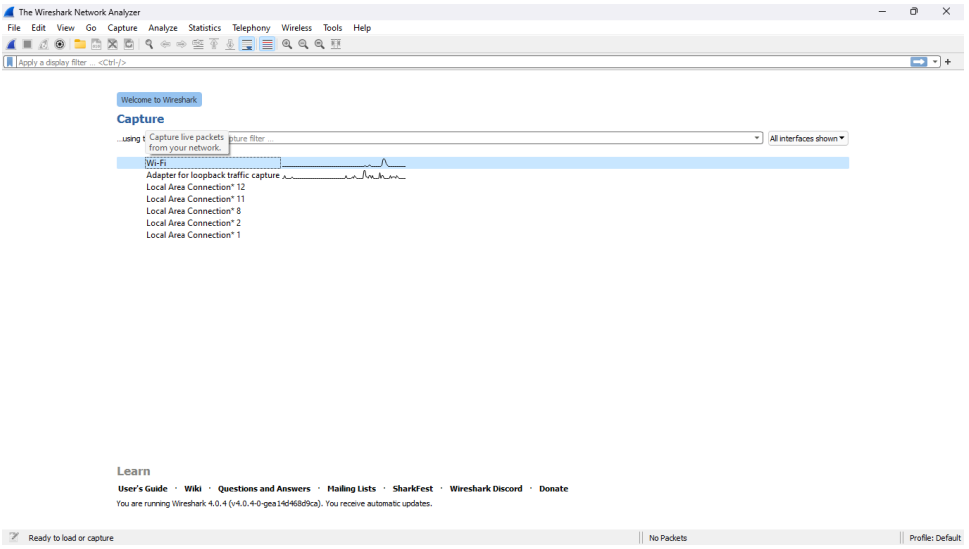


Figure 15 - All network adapters

Step-2-> Click start

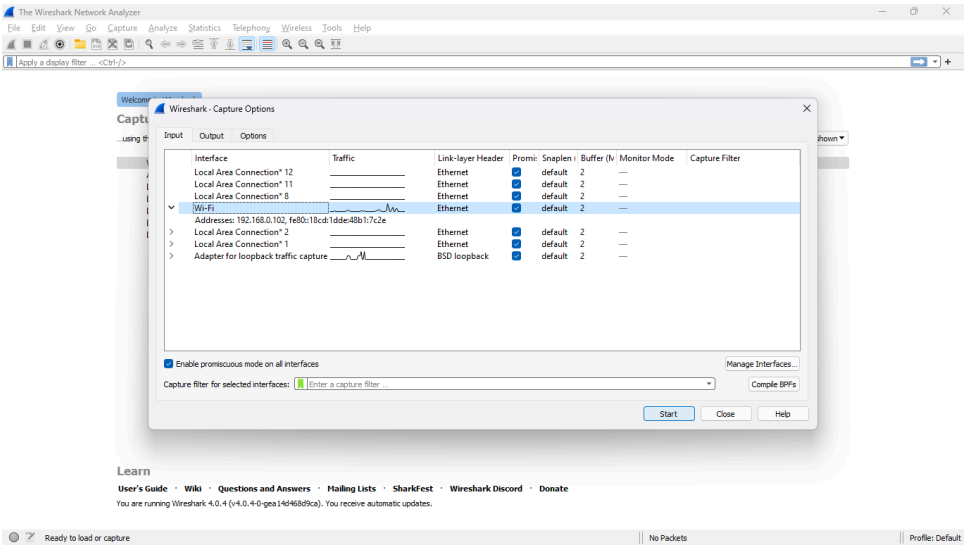


Figure 16 - selecting adapter to capture packets

Step-3-> Following window will open

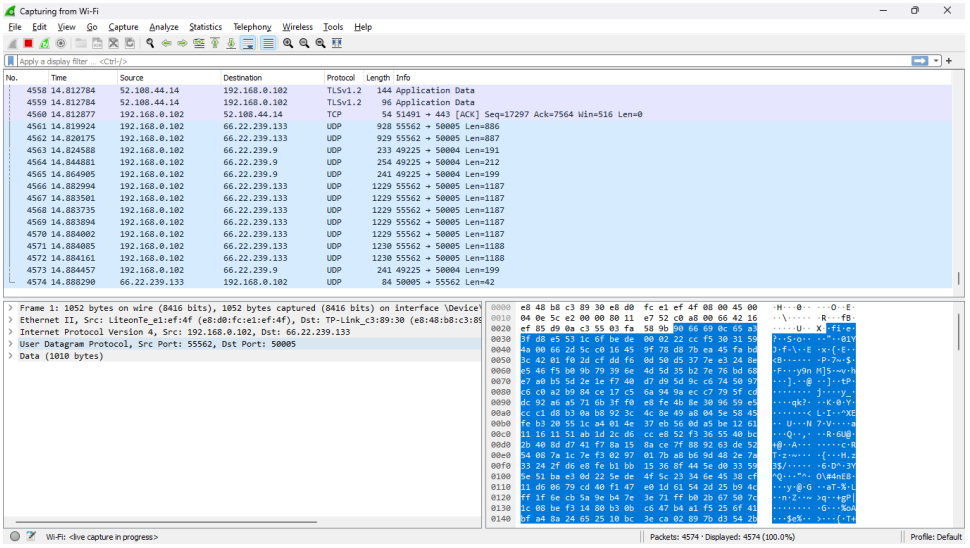


FIGURE 17 - ALL PACKETS CAPTURED BY WIRESHARK

Step-4-> Search for dns in the search bar

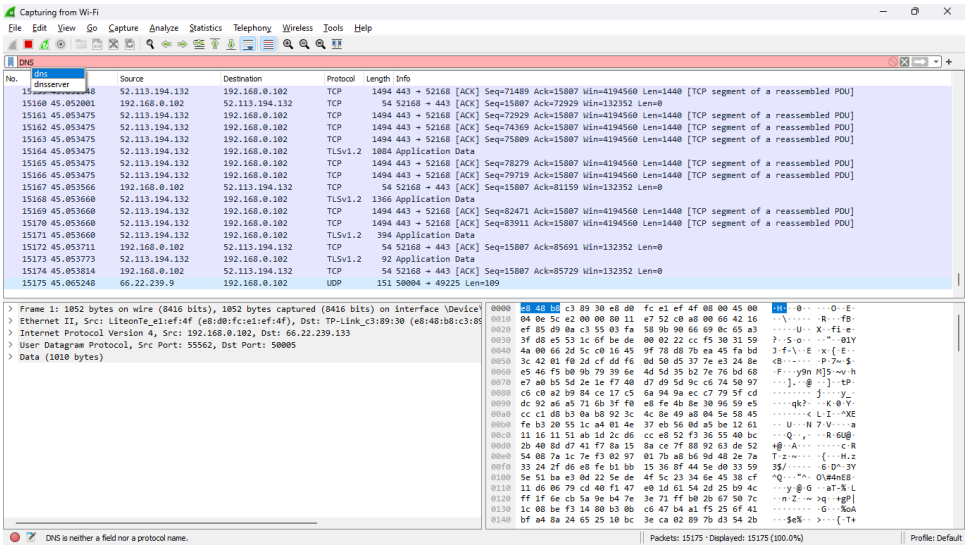


FIGURE 18 DNS FILTER

Step-5-> Search for some websites and check them on Wireshark, it will give you the following output

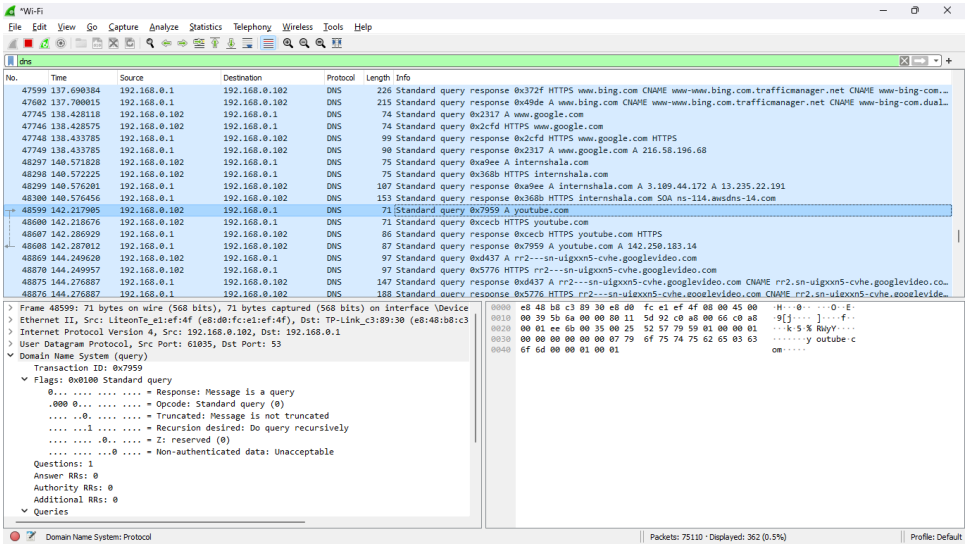


FIGURE 19 - PACKETS INCOMING FROM YOUTUBE.COM

Step-6-> Double click on that and open the DNS arrow and analyse it

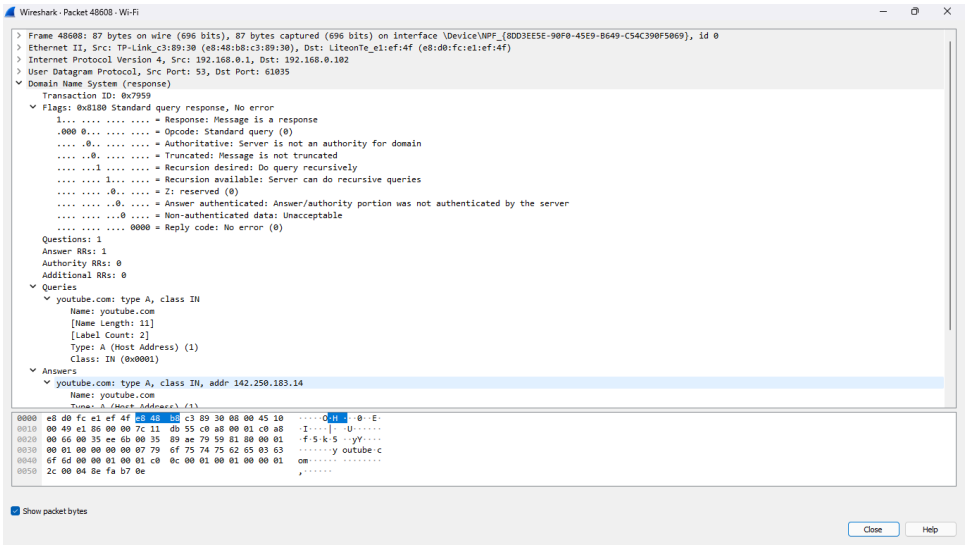


FIGURE 20 - OPENING A PACKET AND ANALYZING

Practical 5: Create your own syslog Server

Step 1-> Run **sudo apt-get install openssh-server** command to install ssh

```
ubuntu@ubuntu:~$ sudo apt-get install openssh-server
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-
0.5).
The following package was automatically installed and
onger required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 32 not
d.
```

Step 2-> Run **sudo apt-get install net-tools** to install net-tools

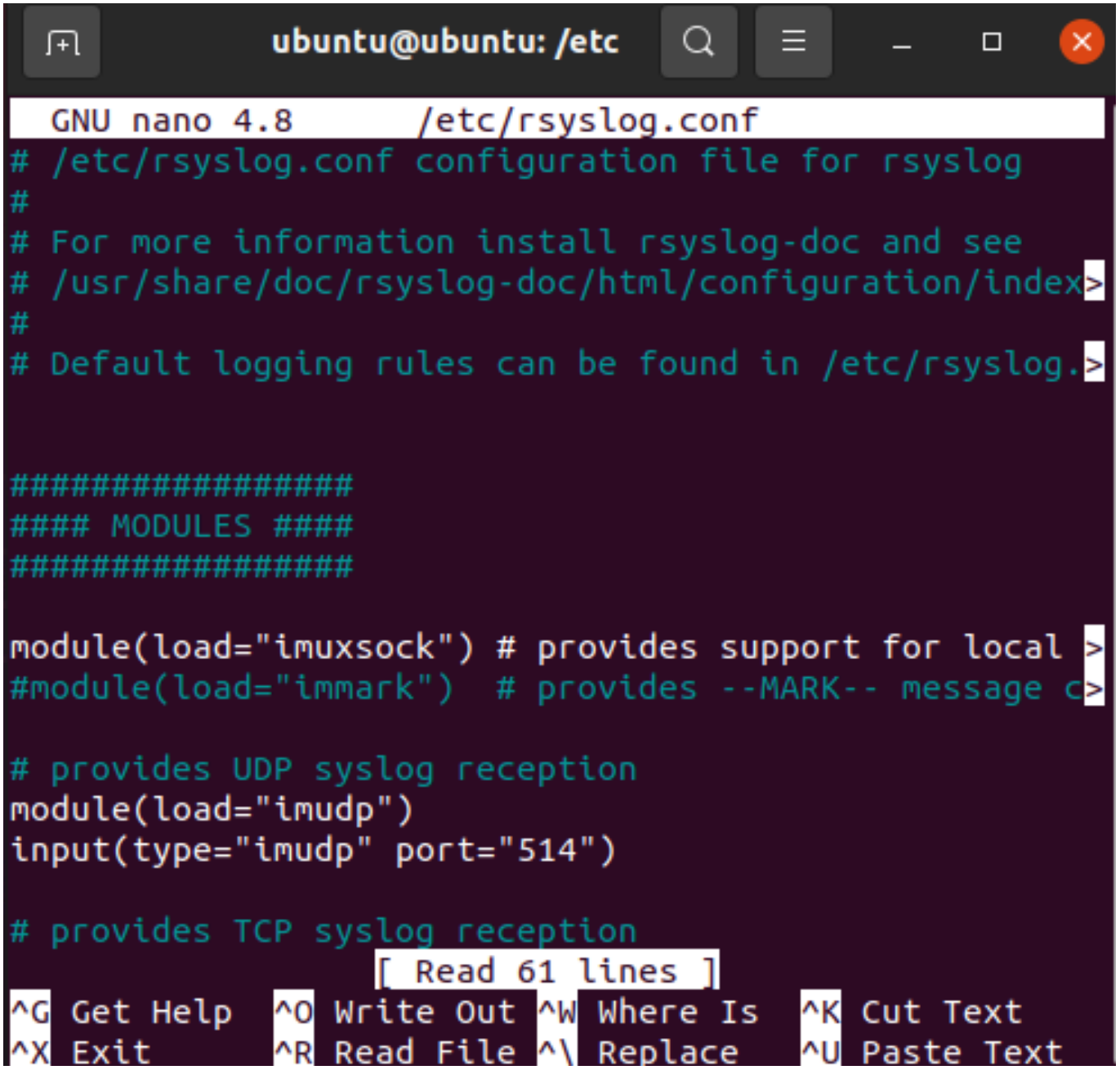
```
ubuntu@ubuntu:~$ sudo apt-get install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
net-tools is already the newest version (1.60+gi
d88e-1ubuntu1).
The following package was automatically installe
onger required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 3
d
```

Step 3-> Run **sudo service rsyslog restart** to restart rsyslog service

```
ubuntu@ubuntu:/etc$ sudo service rsyslog restart
ubuntu@ubuntu:/etc$
```

Step 4-> Edit rsyslog config using **sudo nano /etc/rsyslog.conf** command and uncomment the lines
module(load="imudp") input(type="imudp" port="514")
module(load="imtcp") input(type="imtcp" port="514")
Press **Ctrl+X** & **Enter** to save file

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



```
# 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-> Restart rsyslog service with **sudo service rsyslog restart** command

```
ubuntu@ubuntu:/etc$ sudo service rsyslog restart
```

Step 6-> Check status of rsyslog service with **sudo service rsyslog status** command

```
ubuntu@ubuntu:/etc$ sudo service rsyslog status
● rsyslog.service - System Logging Service
     Loaded: loaded (/lib/systemd/system/rsyslog.serv>
     Active: active (running) since Fri 2023-04-21 04>
TriggeredBy: ● syslog.socket
       Docs: man:rsyslogd(8)
             https://www.rsyslog.com/doc/
   Main PID: 14060 (rsyslogd)
      Tasks: 9 (limit: 3449)
     Memory: 1.2M
     CGroup: /system.slice/rsyslog.service
             └─14060 /usr/sbin/rsyslogd -n -iNONE

Apr 21 04:16:20 ubuntu systemd[1]: Starting System Lo>
Apr 21 04:16:20 ubuntu systemd[1]: Started System Log>
Apr 21 04:16:20 ubuntu rsyslogd[14060]: imuxsock: Acq>
Apr 21 04:16:20 ubuntu rsyslogd[14060]: rsyslogd's gr>
Apr 21 04:16:20 ubuntu rsyslogd[14060]: rsyslogd's us>
Apr 21 04:16:20 ubuntu rsyslogd[14060]: [origin softw>
lines 1-18/18 (END)
```

Terminal

Practical 6: Install and Run Splunk on Linux

Step-1-> Run **sudo apt-get update , sudo apt-get upgrade** primarily. Next run **sudo apt-get install net-tools wget openssh-server** . This install necessary tools for this practical.

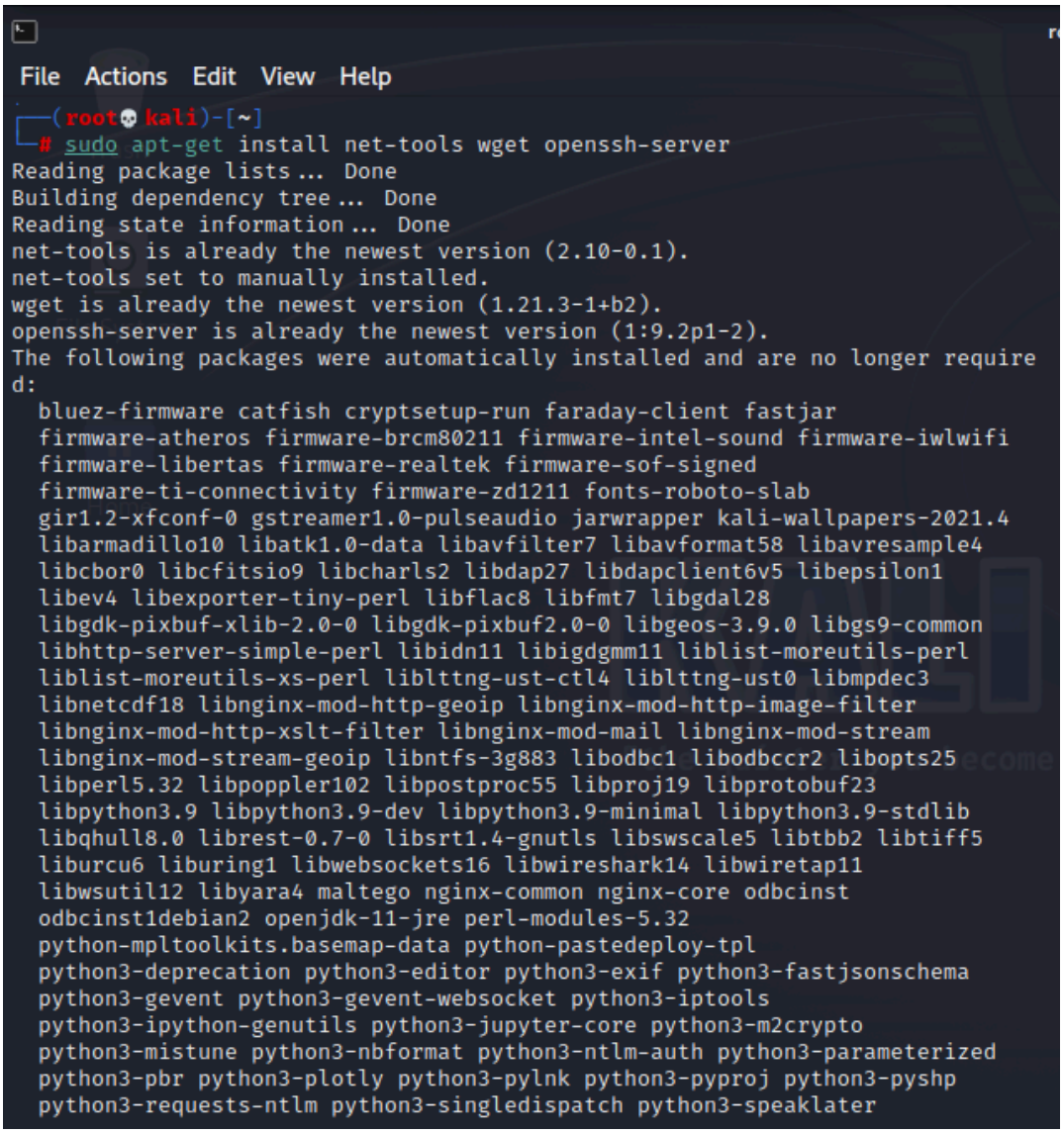


Figure 21 - updating and upgrading kali

Step-2-> Next restart ssh server using **`sudo systemctl restart ssh`** and run **`sudo nano /etc/rsyslog.conf`** to edit rsyslog configuration file.

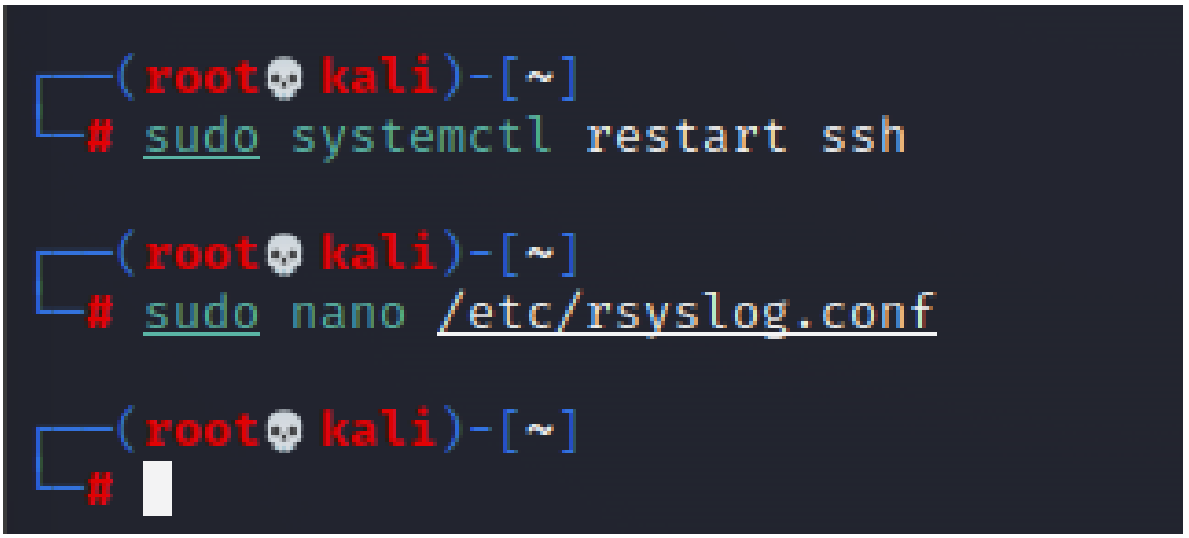


FIGURE 22 - RESTARTING SSH SERVICE AND EDITING RSYSLOG CONFIGURATION FILE

Step-3-> Uncomment lines **`module(load="imudp") input(type="imudp" port="514") & module(load="imtcp") input(type="imtcp" port="514")`** and Press `Ctrl+X`, then Y, and Enter

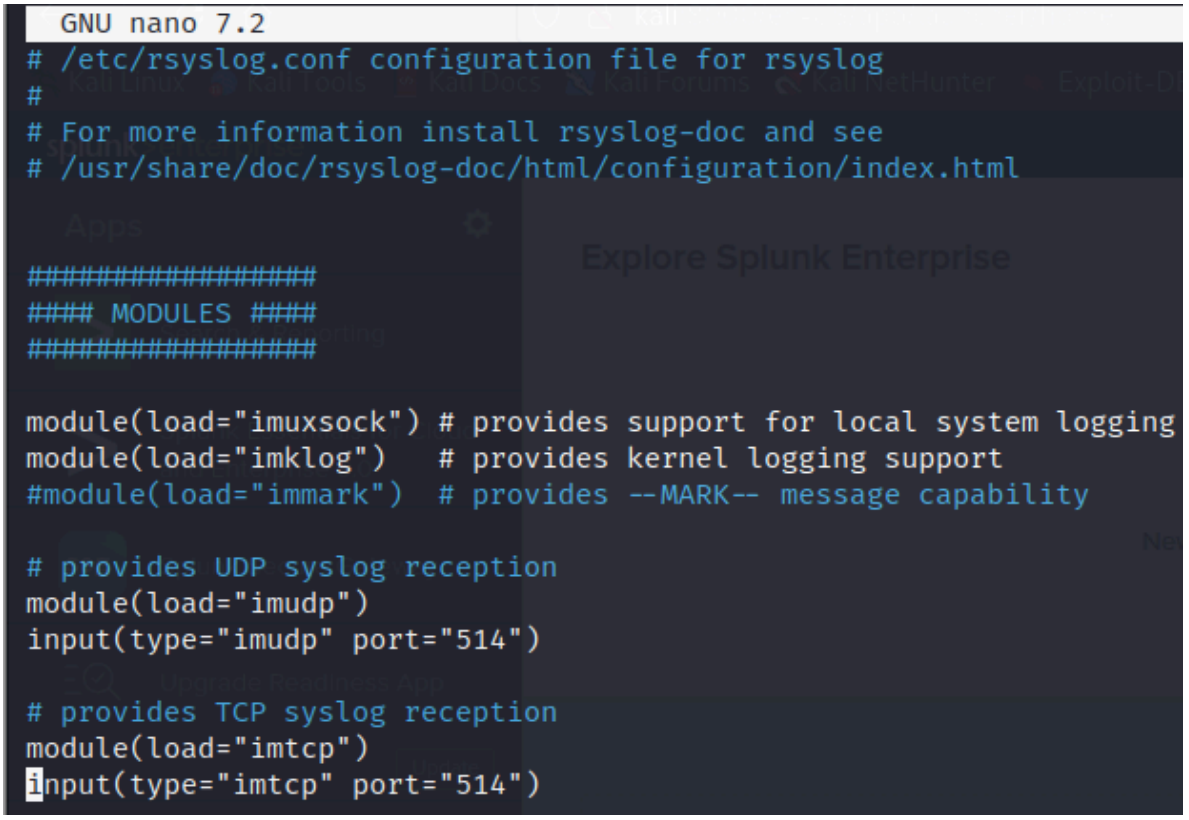


FIGURE 23 -UNCOMMENTING LINES

Step-4-> Run **`sudo systemctl restart rsyslog`** to restart rsyslog service. Run **`sudo systemctl rsyslog status`** to check status of service. If service is active, you can proceed.

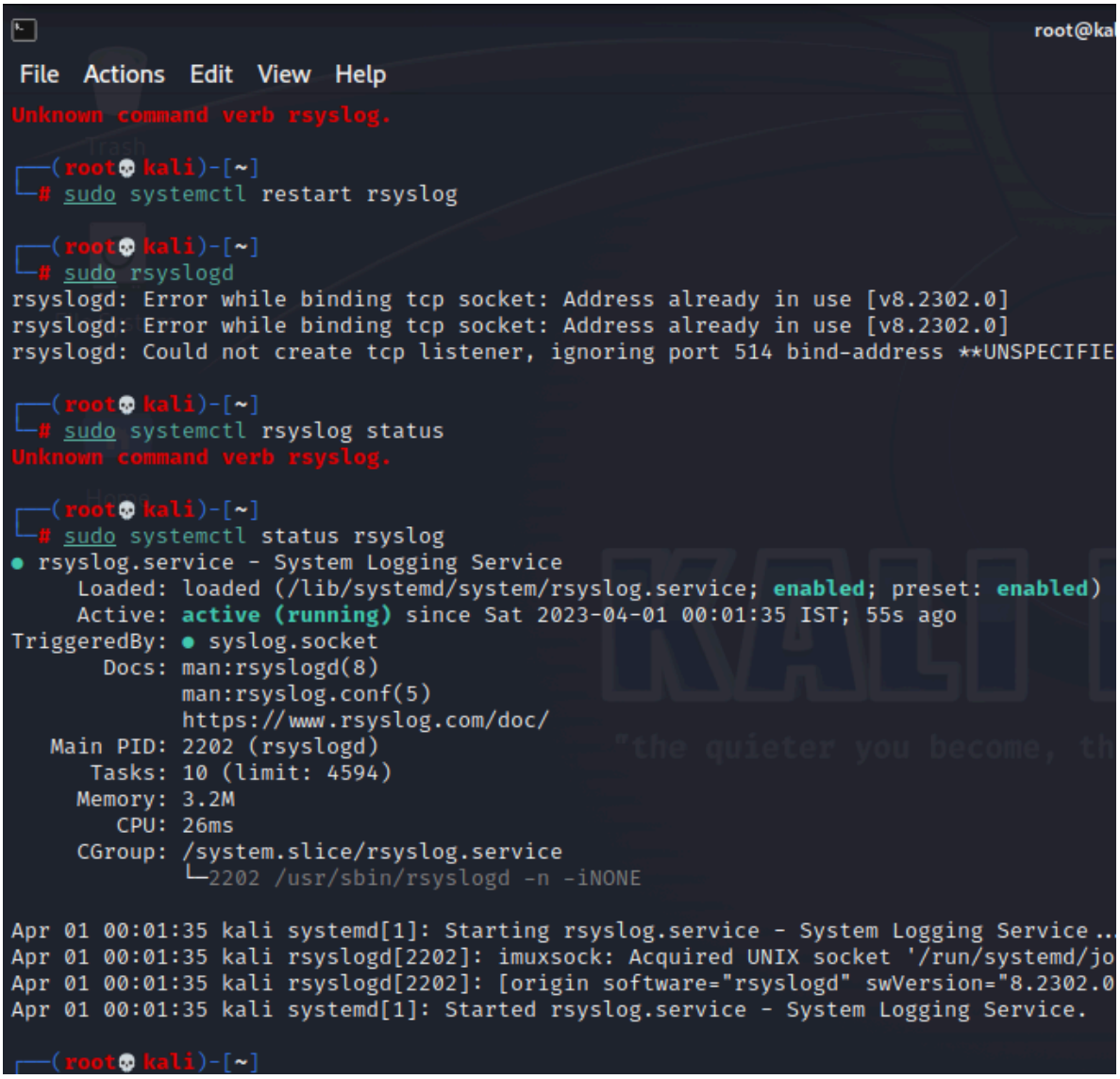


FIGURE 24 - RESTARTING RSYSLOG SERVICE AND CHECKING STATUS

Step-5-> Add a group using **`sudo groupadd <group_name>`** . Add a user to that group using **`sudo useradd -d /opt/<group_name> -m -g splunk splunkdemo`**

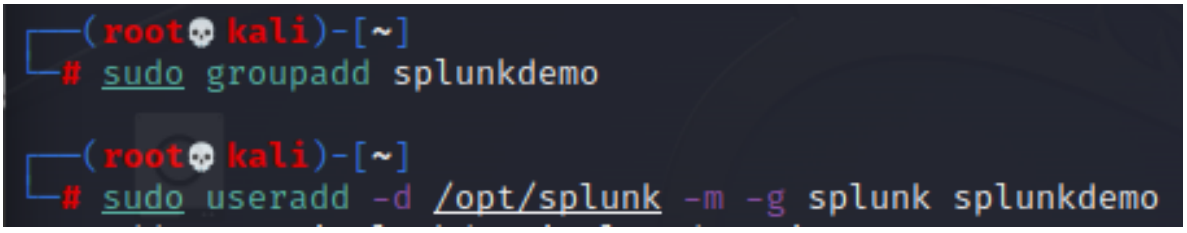


FIGURE 25 - ADDING GROUP AND NEW USER

Step-6-> Now go to https://www.splunk.com/ . Register and Click on Free Splunk at top right of page. Click on Free Trials and Downloads page link. On the next screen Click on Get My Free Trial under Splunk Enterprise. Next Choose Linux and click on Download for .tgz. Stop the download and click

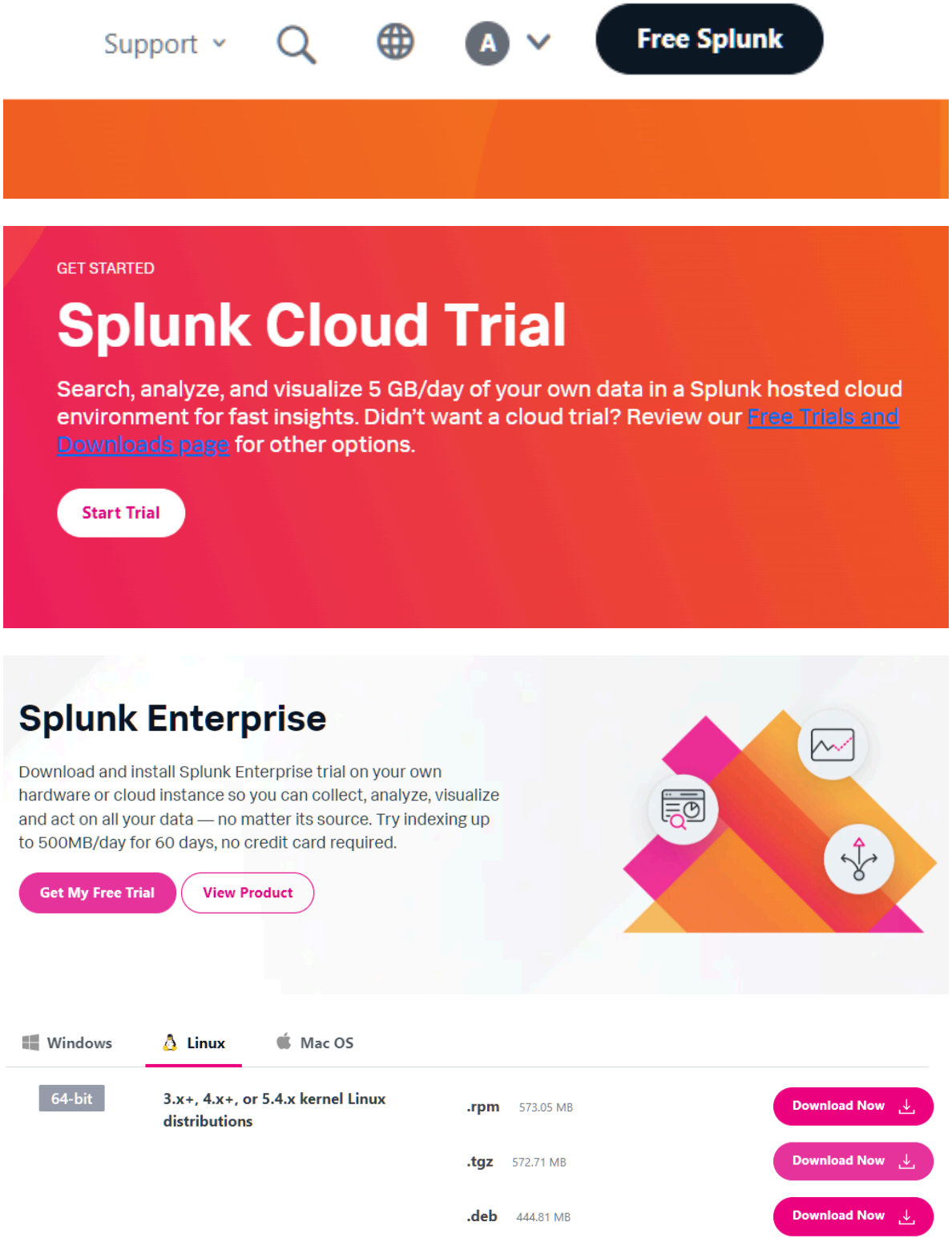


FIGURE 26 - PROCESS TO DOWNLOAD SPLUNK ON LINUX

Step-7-> Stop the download and click on Dowload via Command Line (wget) and copy the command.

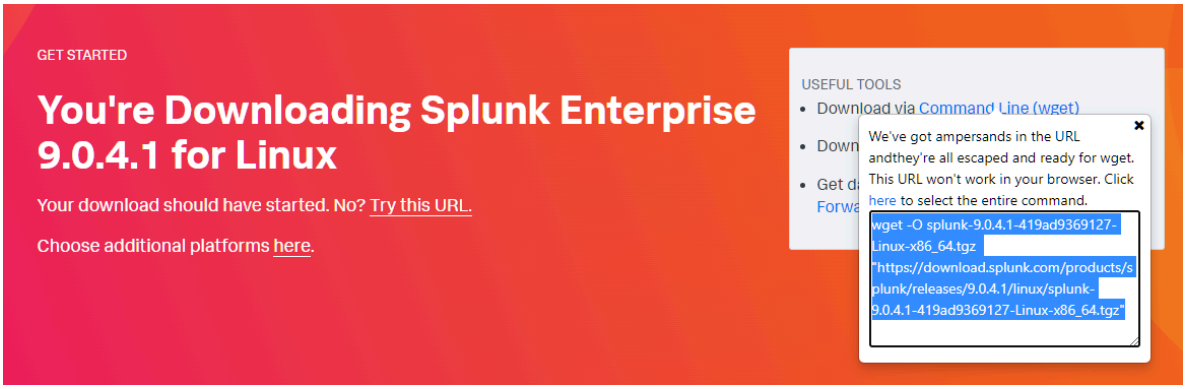


FIGURE 27 - COPYING WGET COMMAND TO DOWNLOAD SPLUNK

Step-8-> Now change user using **`su - <username>`** and paste the command you copied before.

Step-9-> Now run **`tar xvzf splunk-9.0.4.1-419ad9369127-Linux-x86_64.tgz -C /opt`** (Note: the splunk version will differ with time. Change version number accordingly.)

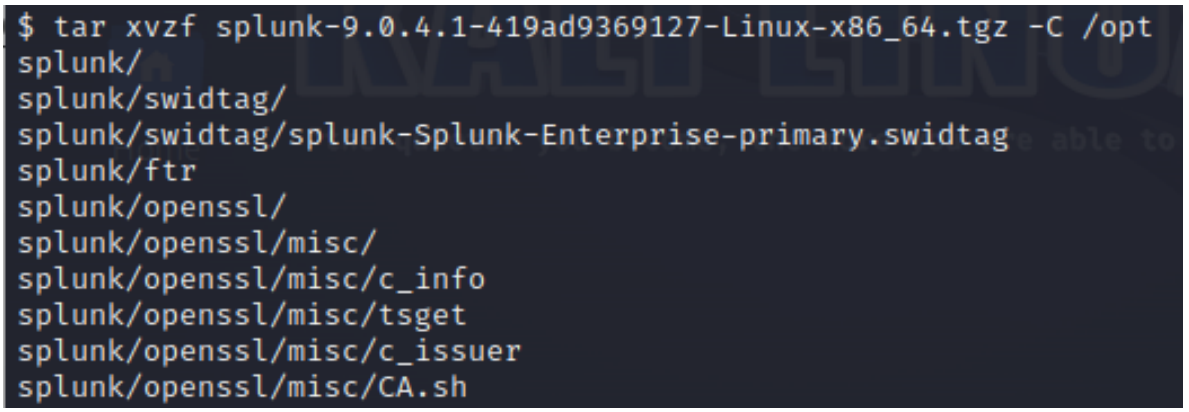


FIGURE 28 - UNZIPPING TAR FILE

Step-10-> Now access root using su – and **`run chown -R splunk: /opt/splunk/`** and change directory to **`/opt/splunk/bin`**

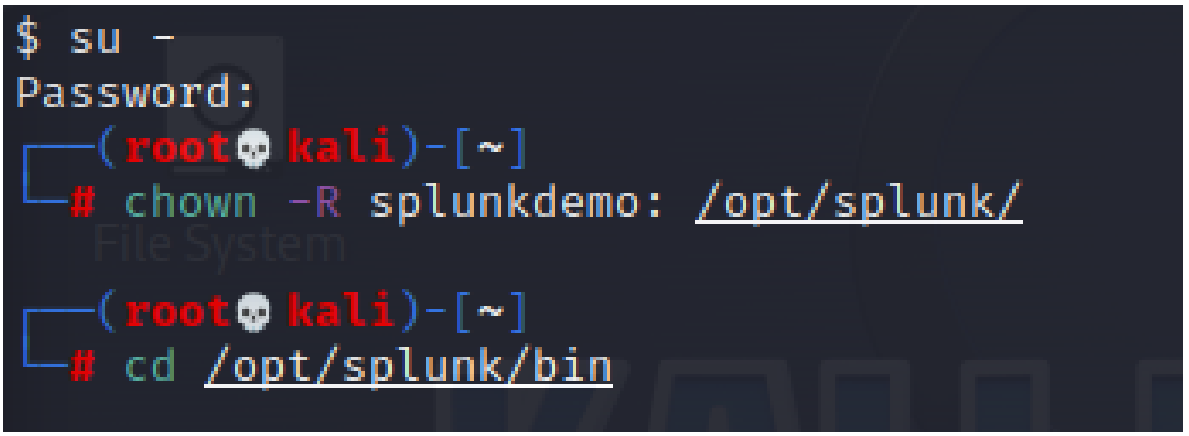


FIGURE 29 - CHANGING OWNERSHIP

Step-11-> Start splunk service using **`./splunk start`** . Hold spacebar to skip through the EULA.

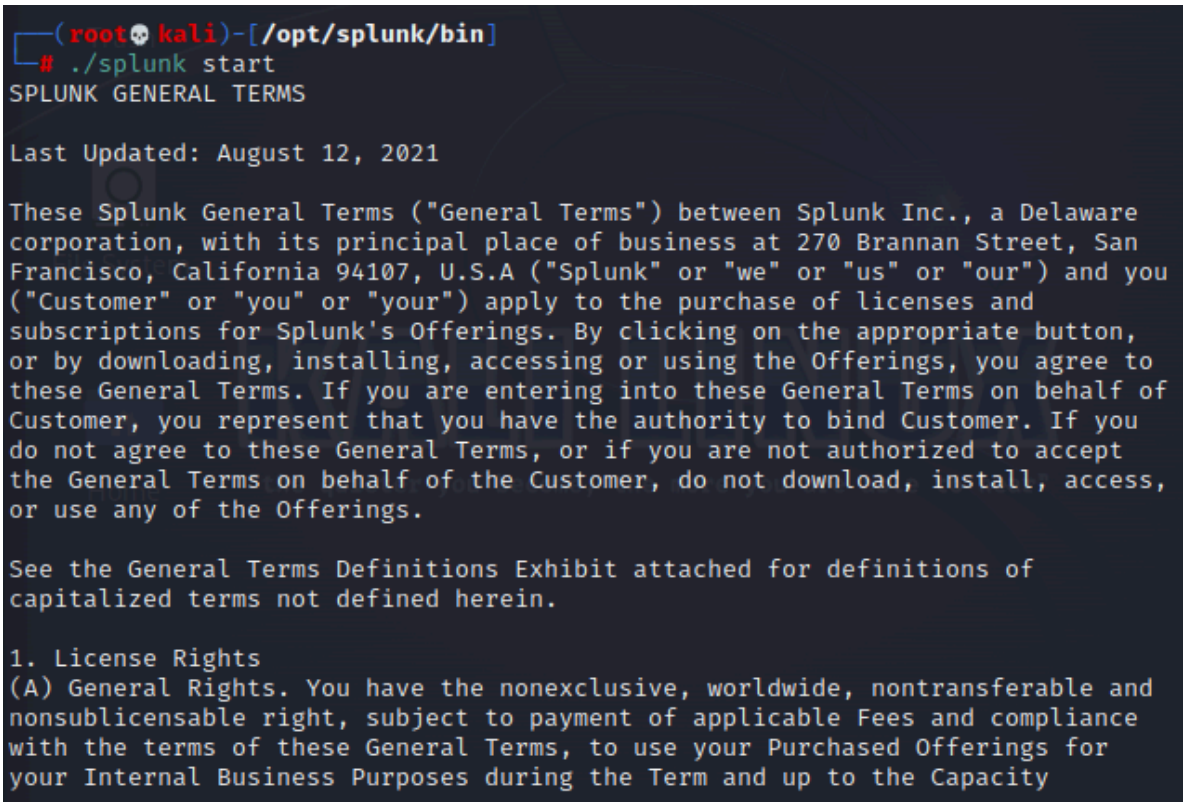


FIGURE 30 – GENERAL TERMS & CONDITIONS OF SPLUNK

Step-12-> To accept splunk license run **`./splunk start --accept-license`**. You will be prompted to provide a username and password. This will be your login credentials to splunk on your linux system.

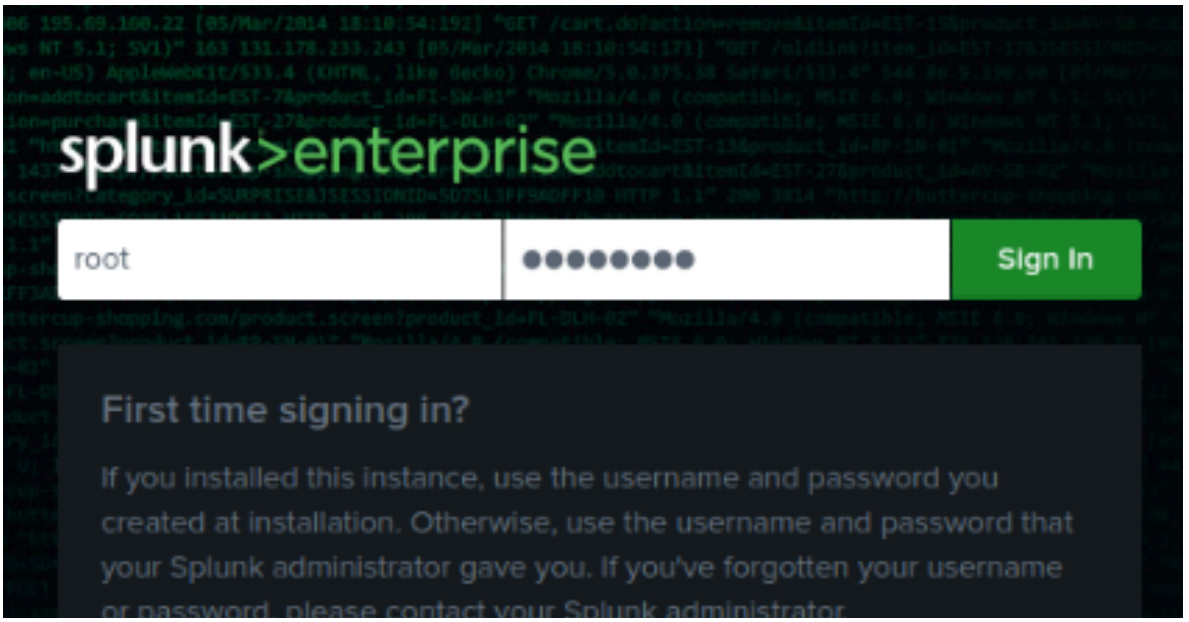
```
┌──(root💀kali)-[/opt/splunk/bin]
└─# ./splunk start --accept-license

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

Splunk software must create an administrator account during startup. Otherwise, you c
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: root
Password must contain at least:
   * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
ERROR: Password did not meet complexity requirements. 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/lda
Generating RSA private key, 2048 bit long modulus
.........................+++++
.....................................................................................
e is 65537 (0×10001)
```

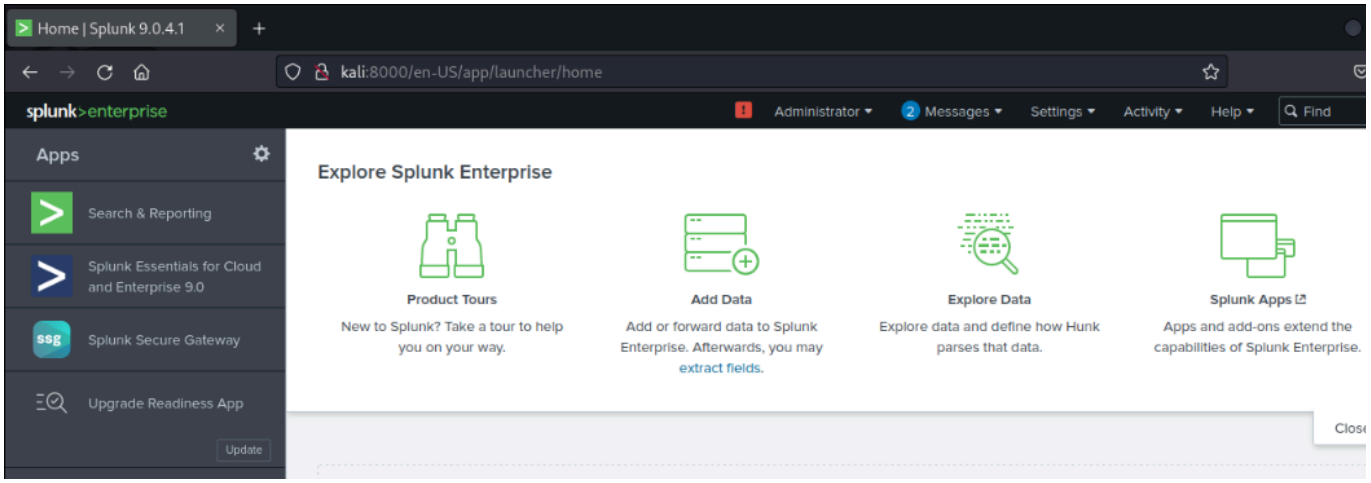
Figure 31 - accepting license

```
File  Actions  Edit  View  Help
        All installed files intact.
        Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Generating a RSA private key
.....................................................+++++
.....................................................+++++
writing new private key to 'privKeySecure.pem'
-----
Signature ok
subject=/CN=kali/O=SplunkUser
Getting CA Private Key
writing RSA key
PYTHONHTTPSVERIFY is set to 0 in splunk-launch.conf disabling ce
must be set to "1" for increased security
Done


Waiting for web server at http://127.0.0.1:8000 to be available.


If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://kali:8000


┌──(root💀kali)-[/opt/splunk/bin]
└─#

┌──(root💀kali)-[/opt/splunk/bin]
└─#
```

Figure 32 - splunk interface is now accessible on kal:8000

Step-13-> Access the splunk web interface on http://kali:8000. You will see the following screen if you've followed every step to the mark. Now login using the username and password you were asked to provide while accepting license.





Practical 7: Install and configure ELK on linux

NOTE: To save a file Press CTRL+X, Y and Press Enter.

Step-1-> Run **sudo apt-get update** & **sudo apt-get upgrade** in terminal

```
┌──(root💀kali)-[~]
└─# sudo apt-get update
0% [Connecting to http.kali.org (192.99.200.113)] [Connected to brave-browser-apt-release.s3.^
Hit:1 https://brave-browser-apt-release.s3.brave.com stable InRelease
Get:2 https://kali.download/kali kali-rolling InRelease [41.2 kB]
Get:3 https://kali.download/kali kali-rolling/main amd64 Packages [19.3 MB]
Get:4 https://kali.download/kali kali-rolling/main amd64 Contents (deb) [44.4 MB]
Get:5 https://kali.download/kali kali-rolling/non-free amd64 Packages [217 kB]
Get:6 https://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [927 kB]
Fetched 64.8 MB in 29s (2,211 kB/s)
Reading package lists ... Done

┌──(root💀kali)-[~]
└─#
```

```
┌──(root💀kali)-[~]
└─# sudo apt-get upgrade
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Calculating upgrade ... Done
The following packages were automatically installed and are no longer required:
  bluez-firmware catfish cryptsetup-run faraday-client fastjar firmware-atheros
  firmware-brcm80211 firmware-intel-sound firmware-iwlwifi firmware-libertas
  firmware-realtek firmware-sof-signed firmware-ti-connectivity firmware-zd1211
  fonts-roboto-slab gir1.2-xfconf-0 gstreamer1.0-pulseaudio jarwrapper
  kali-wallpapers-2021.4 libarmadillo10 libatk1.0-data libavfilter7 libavformat58
  libavresample4 libcbor0 libcfitsio9 libcharls2 libdap27 libdapclient6v5 libepsilon1 libev4
  libexporter-tiny-perl libflac8 libfmt7 libgdal28 libgdk-pixbuf-xlib-2.0-0
  libgdk-pixbuf2.0-0 libgeos-3.9.0 libgs9-common libhttp-server-simple-perl libidn11
  libigdgmm11 liblist-moreutils-perl liblist-moreutils-xs-perl liblttng-ust-ctl4
  liblttng-ust0 libmpdec3 libnetcdf18 libnginx-mod-http-geoip libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip libntfs-3g883 libodbc1 libodbccr2 libopts25 libperl5.32
  libpoppler102 libpostproc55 libproj19 libprotobuf23 libpython3.9 libpython3.9-dev
  libpython3.9-minimal libpython3.9-stdlib libqhull8.0 librest-0.7-0 libsrt1.4-gnutls
  libswscale5 libtbb2 libtiff5 liburcu6 liburing1 libwebsockets16 libwireshark14
  libwiretap11 libwsutil12 libyara4 maltego nginx-common nginx-core odbcinst
  odbcinst1debian2 openjdk-11-jre perl-modules-5.32 python-mpltoolkits.basemap-data
  python-pastedeploy-tpl python3-deprecation python3-editor python3-exif
  python3-fastjsonschema python3-gevent python3-gevent-websocket python3-iptools
  python3-ipython-genutils python3-jupyter-core python3-m2crypto python3-mistune
```

Figure 33 - Updating and upgrading kali linux

Step-2-> Install java JDK using **sudo apt-get install default-jdk default-jre** command

```
┌──(root💀 kali)-[~]
└─# sudo apt-get install default-jdk default-jre
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
default-jre is already the newest version (2:1.17-74).
default-jre set to manually installed.
The following packages were automatically installed and are no longer required:
  bluez-firmware catfish cryptsetup-run faraday-client fastjar firmware-atheros
  firmware-brcm80211 firmware-intel-sound firmware-iwlwifi firmware-libertas
  firmware-realtek firmware-sof-signed firmware-ti-connectivity firmware-zd1211
  fonts-roboto-slab gir1.2-xfconf-0 gstreamer1.0-pulseaudio jarwrapper
  kali-wallpapers-2021.4 libarmadillo10 libatk1.0-data libavfilter7 libavformat58
  libavresample4 libcbor0 libcfitsio9 libcharls2 libdap27 libdapclient6v5 libepsilon1 libev4
  libexporter-tiny-perl libflac8 libfmt7 libgdal28 libgdk-pixbuf-xlib-2.0-0
  libgdk-pixbuf2.0-0 libgeos-3.9.0 libgs9-common libhttp-server-simple-perl libidn11
  libigdgmm11 liblist-moreutils-perl liblist-moreutils-xs-perl liblttng-ust-ctl4
  liblttng-ust0 libmpdec3 libnetcdf18 libnginx-mod-http-geoip libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip libntfs-3g883 libodbc1 libodbccr2 libopts25 libperl5.32
  libpoppler102 libpostproc55 libproj19 libprotobuf23 libpython3.9 libpython3.9-dev
  libpython3.9-minimal libpython3.9-stdlib libqhull8.0 librest-0.7-0 libsrt1.4-gnutls
```

FIGURE 34 - INSTALLING JAVA JDK AND JRE

Step-3-> Check java version using **java -version** command

```
┌──(root💀 kali)-[~]
└─# java -version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
openjdk version "17.0.6" 2023-01-17
OpenJDK Runtime Environment (build 17.0.6+10-Debian-1)
OpenJDK 64-Bit Server VM (build 17.0.6+10-Debian-1, mixed mode, sharing)
```

FIGURE 35 - CHECKING JAVA INSTALLATION AND VERSION

Step-4-> Set the java configuration to use using **update-alternatives –config java** command and select the latest version installed.

```
┌──(root💀 kali)-[~]
└─# update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority   Status
------------------------------------------------------------
* 0            /usr/lib/jvm/java-17-openjdk-amd64/bin/java   1711      auto mode
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      manual mode
  2            /usr/lib/jvm/java-17-openjdk-amd64/bin/java   1711      manual mode

Press <enter> to keep the current choice[*], or type selection number: 0

┌──(root💀 kali)-[~]
└─#
```

FIGURE 36 - SET JAVA VERSION TO USE

Step-5-> Now to update the system environment configuration enter **sudo nano /etc/environment** command.

FIGURE 37 - UPDATING SYSTEM CONFIGURATION AND ADD JAVA_HOME VARIABLE

Step-6-> Now to access remote repositories install APT Transport package using **sudo apt-get install apt-transport-https** command.

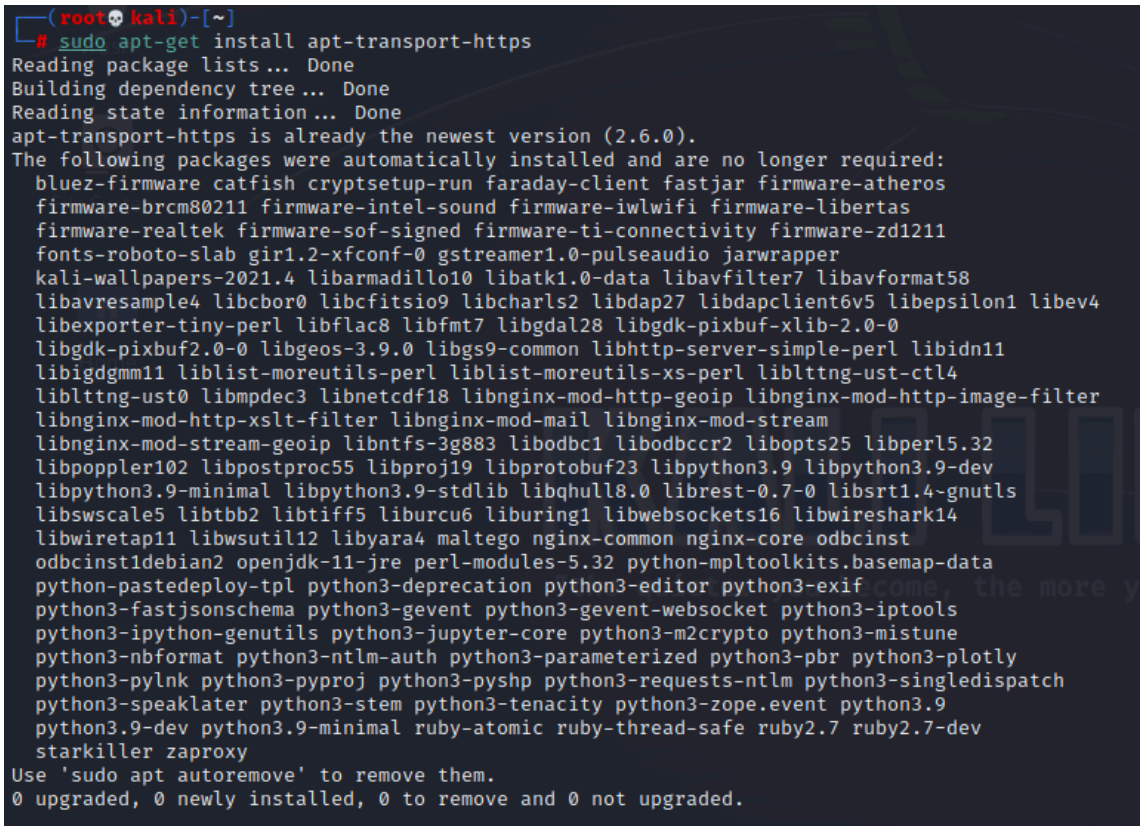


FIGURE 38 - INSTALLING APT-TRANSPORT HTTPS

Step-7-> Now run **wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg** command. This command stores a gpg key fetched from the repository URL and decrypts and store it in keyrings folder.

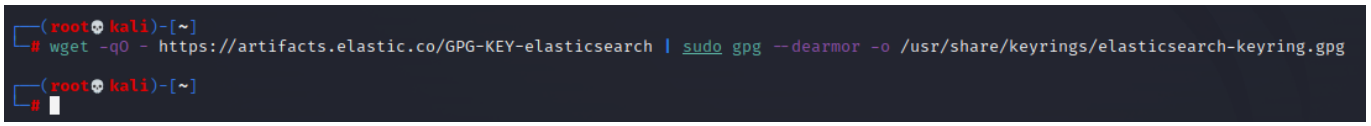


FIGURE 39 - STORING GPG KEY FROM ELASTIC SEARCH KEYSTORE

Step-8-> Now run **echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list** command. Also run **sudo apt-get update** command

```
┌──(root💀kali)-[~]
└─# echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/el
astic-8.x.list
deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main
```

```
┌──(root💀kali)-[~]
└─# sudo apt-get update
Hit:1 https://brave-browser-apt-release.s3.brave.com stable InRelease
Get:2 https://artifacts.elastic.co/packages/8.x/apt stable InRelease [10.4 kB]
Get:3 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 Packages [48.4 kB]
Get:4 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 Contents (deb) [1,217 kB]
Hit:5 https://kali.download/kali kali-rolling InRelease
Fetched 1,275 kB in 1s (889 kB/s)
Reading package lists ... Done
```

FIGURE 40 - SETTING REPOSITORY LOCATION AND UPDATING KALI AGAIN

Step-9-> Now install elasticsearch using **sudo apt-get install elasticsearch** command

```
┌──(root💀kali)-[~]
└─# sudo apt-get install elasticsearch
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  bluez-firmware catfish cryptsetup-run faraday-client fastjar firmware-atheros firmware-brcm80211 firmware-intel-sound firmware-iwlwifi firmware-libertas
  firmware-realtek firmware-sof-signed firmware-ti-connectivity firmware-zd1211 fonts-roboto-slab gir1.2-xfconf-0 gstreamer1.0-pulseaudio jarwrapper
  kali-wallpapers-2021.4 libarmadillo10 libatk1.0-data libavfilter7 libavformat58 libavresample4 libcbor0 libcfitsio9 libcharls2 libdap27 libdapclient6v5 libepsilon1
  libev4 libexporter-tiny-perl libflac8 libfmt7 libgdal28 libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 libgeos-3.9.0 libgs9-common libhttp-server-simple-perl libidn11
  libigdgmm11 liblist-moreutils-perl liblist-moreutils-xs-perl libltdl-ust-ctl4 libltdl-ust0 libmpdec3 libnetcdf18 libnginx-mod-http-geoip
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip libntfs-3g883 libodbc1 libodbccr2
  libopts25 libperl5.32 libpoppler102 libpostproc55 libproj19 libprotobuf23 libpython3.9 libpython3.9-dev libpython3.9-minimal libpython3.9-stdlib libqhull8.0
  librest-0.7-0 libsrt1.4-gnutls libswscale5 libtbb2 libtiff5 liburcu6 liburing1 libwebsockets16 libwireshark14 libwiretap11 libwsutil12 libyara4 maltego nginx-common
  nginx-core odbcinst odbcinst1debian2 openjdk-11-jre perl-modules-5.32 python-mpltoolkits.basemap-data python-pastedeploy-tpl python3-deprecation python3-editor
  python3-exif python3-fastjsonschema python3-gevent python3-gevent-websocket python3-iptools python3-ipython-genutils python3-jupyter-core python3-m2crypto
  python3-mistune python3-nbformat python3-ntlm-auth python3-parameterized python3-pbr python3-plotly python3-pylnk python3-pyproj python3-pyshp python3-requests-ntlm
  python3-singledispatch python3-speaklater python3-stem python3-tenacity python3-zope.event python3.9 python3.9-dev python3.9-minimal ruby-atomic ruby-thread-safe
  ruby2.7 ruby2.7-dev starkiller zaproxy
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 596 MB of archives.
After this operation, 1,234 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 elasticsearch amd64 8.7.0 [596 MB]
Fetched 596 MB in 1min 44s (5,716 kB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 420593 files and directories currently installed.)
Preparing to unpack .../elasticsearch_8.7.0_amd64.deb ...
Creating elasticsearch group ... OK
Creating elasticsearch user ... OK
Unpacking elasticsearch (8.7.0) ...
Setting up elasticsearch (8.7.0) ...
--------------------------- Security autoconfiguration information ------------------------------
```

FIGURE 41 - INSTALLING ELASTIC SEARCH

Step-10-> Now we have to restart linux daemon, start and enable elasticsearch service

```
┌──(root💀kali)-[~]
└─# sudo systemctl daemon-reload

┌──(root💀kali)-[~]
└─# sudo systemctl enable elasticsearch.service
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elasticsearch.service.

┌──(root💀kali)-[~]
└─# sudo systemctl start elasticsearch.service

┌──(root💀kali)-[~]
└─# sudo systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; preset: disabled)
     Active: active (running) since Mon 2023-04-03 21:43:21 IST; 2min 10s ago
       Docs: https://www.elastic.co
   Main PID: 3776 (java)
      Tasks: 74 (limit: 4594)
     Memory: 2.3G
        CPU: 1min 27.191s
     CGroup: /system.slice/elasticsearch.service
             ├─3776 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/usr/share/elasticsearch/bin/elasticsearch -D>
             ├─3835 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -Djava.security.manager=allow -XX:>
             └─3857 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Apr 03 21:42:32 kali systemd[1]: Starting elasticsearch.service - Elasticsearch...
Apr 03 21:43:21 kali systemd[1]: Started elasticsearch.service - Elasticsearch.
lines 1-15/15 (END)
```

FIGURE 42 - STARTING AND ENABLING ELASTIC SEARCH SERVICE

Step-11-> Now edit the elasticsearch.yml file using the **sudo nano /etc/elasticsearch/elasticsearch.yml** command. Uncomment **network.host** and give address **0.0.0.0**. In discovery section add this line in it **discovery.seed_hosts: []** as it is. Also in same file change the value to false for line **xpack.security.enabled:** to false.

```
┌──(root💀kali)-[~]
└─# sudo nano /etc/elasticsearch/elasticsearch.yml
```

```
  GNU nano 7.2                                  /etc/elasticsearch/elasticsearch.yml *
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 0.0.0.0
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
#http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
discovery.seed_hosts: []
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
```

```
  GNU nano 7.2                                  /etc/elasticsearch/elasticsearch.yml *
#
#action.destructive_requires_name: false

#----------------------- BEGIN SECURITY AUTO CONFIGURATION -----------------------
#
# The following settings, TLS certificates, and keys have been automatically
# generated to configure Elasticsearch security features on 03-04-2023 16:09:59
#
# ---------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: false
```

Figure 43 - editing elasticsearch.yml file

Step-12-> Now, restart elastic search service using **sudo systemctl restart elasticsearch.service** command

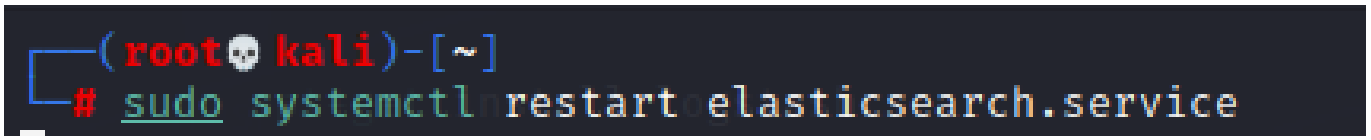
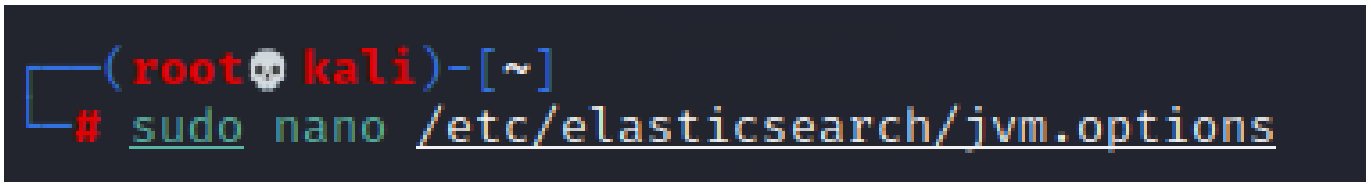


FIGURE 44 - RESTARTING ELASTIC SEARCH SERVICE

Step-13-> Now edit jvmoptions for elastic search using **sudo nano /etc/elasticsearch/jvm.options** command. Change line **-Xms4g** and **-Xmx4g** to -**Xms512m** and **-Xmx512m**.



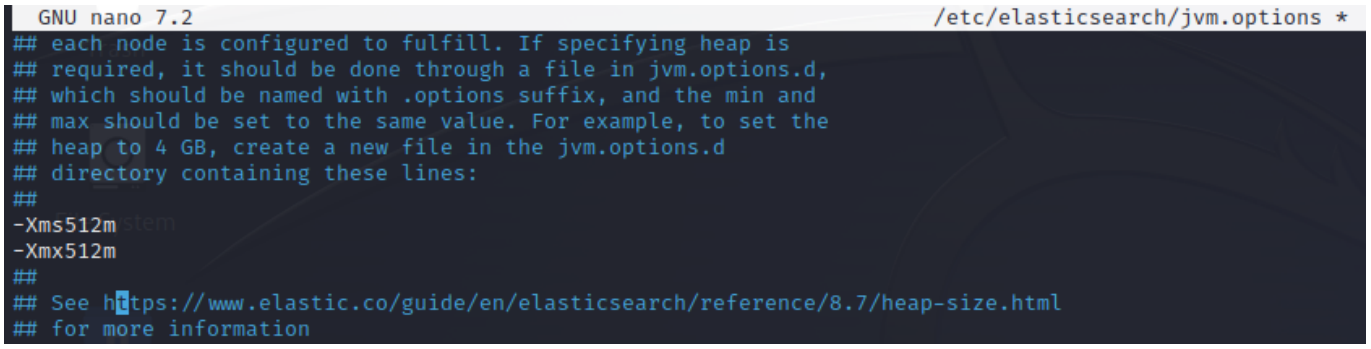


FIGURE 45 - EDITING JVM OPTIONS FOR ELASTIC SEARCH

Step-14-> Now GET the data at localhost:9200 using **curl -X GET "localhost:9200"** command. This command is used to check if elastic search is running on localhost.

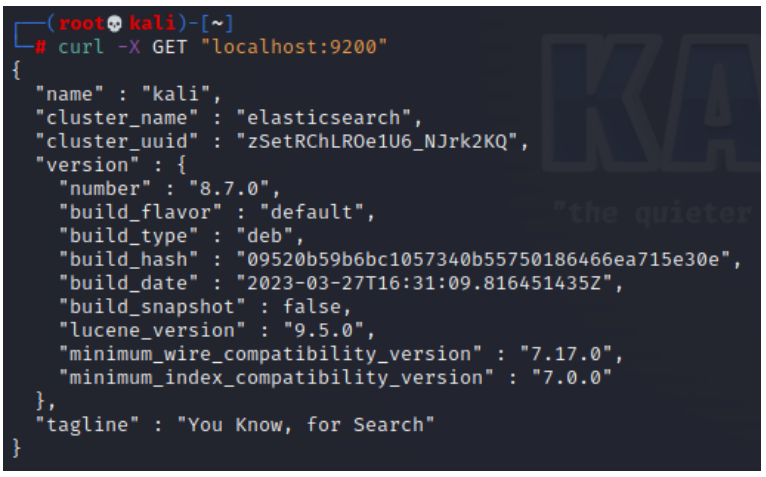


FIGURE 46 - CURL COMMAND TO CHECK STATUS OF ELASTIC SERARCH SERVICE

Step-15-> Now copy the **inet** address of ethernet using **ifconfig** command. Now

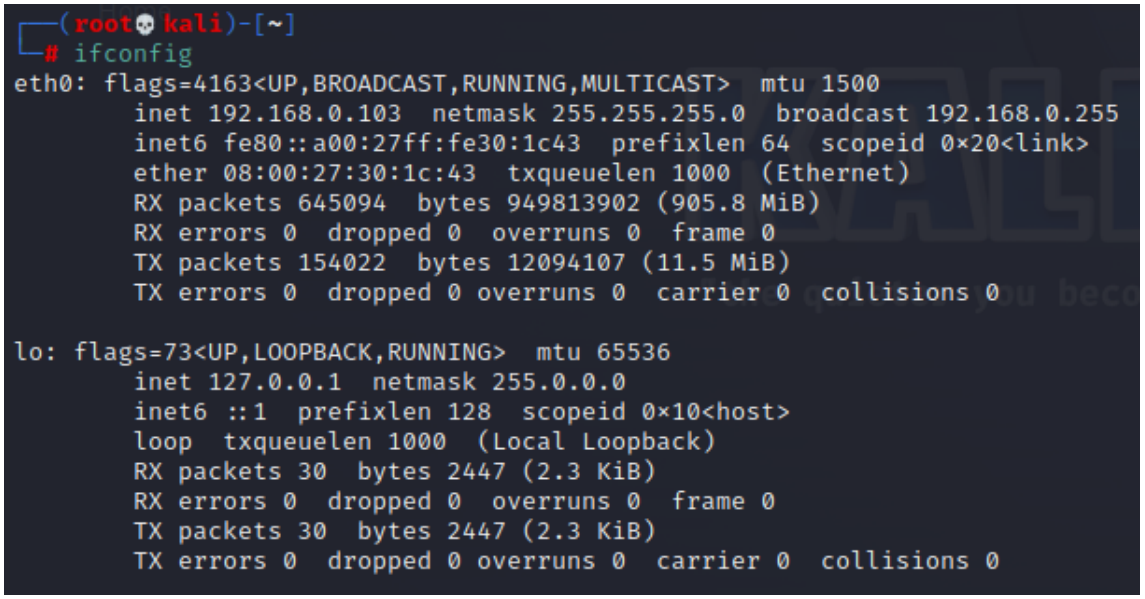


FIGURE 47 - INTERFACE CONFIGURATION COMMAND

Step-16-> Enter the address into a browser and append port :9200 to the IP address

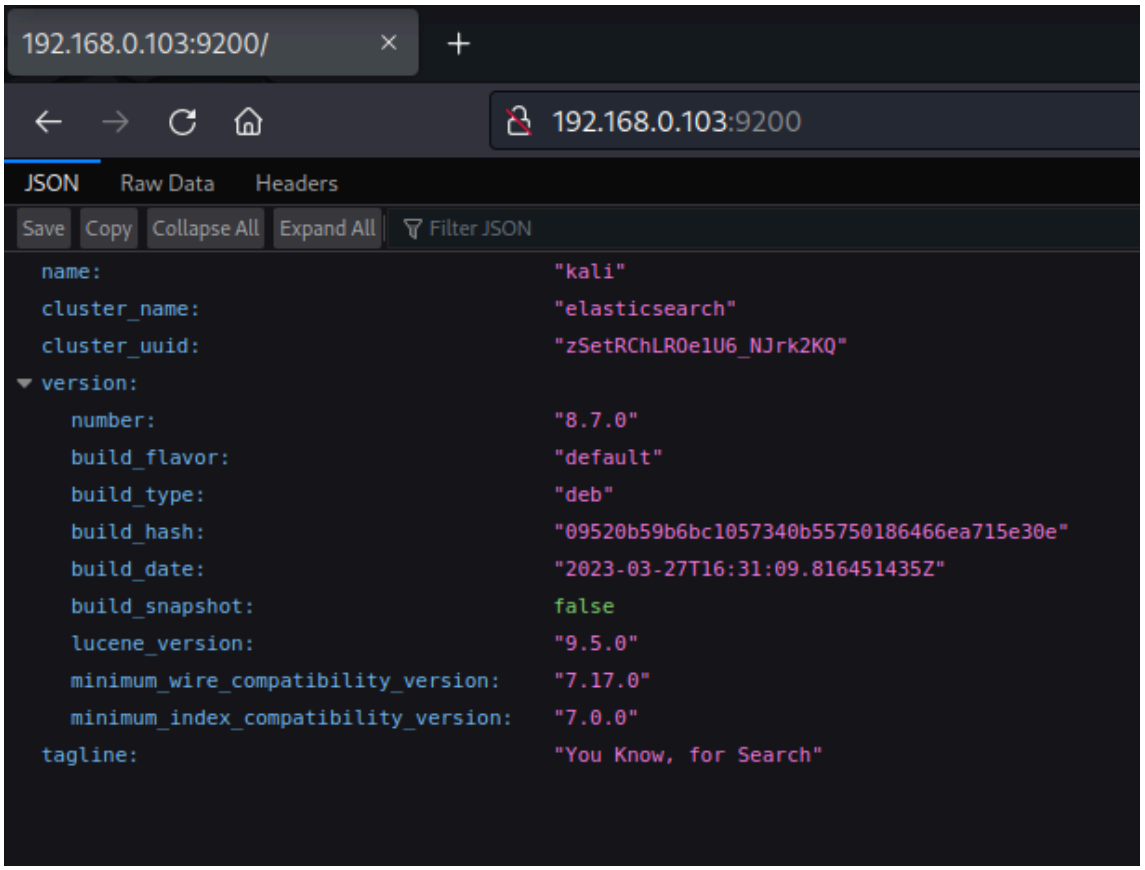


FIGURE 48 - LOCALHOST:9200 IN A BROWSER

Step-17-> Now to install and enable Logstash use the commands **sudo apt-get install logstash**, **sudo systemctl start logstash**, **sudo systemctl enable logstash**, **sudo systemctl status logstash**.

```
┌──(root💀kali)-[~]
└─# sudo apt-get install logstash
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  bluez-firmware catfish cryptsetup-run faraday-client fastjar firmware-atheros firmware-brcm80211 firmw
  firmware-realtek firmware-sof-signed firmware-ti-connectivity firmware-zd1211 fonts-roboto-slab gir1.2
  kali-wallpapers-2021.4 libarmadillo10 libatk1.0-data libavfilter7 libavformat58 libavresample4 libcbor
  libev4 libexporter-tiny-perl libflac8 libfmt7 libgdal28 libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 li
  libigdgmm11 liblist-moreutils-perl liblist-moreutils-xs-perl liblttng-ust-ctl4 liblttng-ust0 libmpdec3
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream lib
  libopts25 libperl5.32 libpoppler102 libpostproc55 libproj19 libprotobuf23 libpython3.9 libpython3.9-de
  librest-0.7-0 libsrt1.4-gnutls libswscale5 libtbb2 libtiff5 liburcu6 liburing1 libwebsockets16 libwire
  nginx-common nginx-core odbcinst odbcinst1debian2 openjdk-11-jre perl-modules-5.32 python-mpltoolkits.
  python3-editor python3-exif python3-fastjsonschema python3-gevent python3-gevent-websocket python3-ipt
  python3-m2crypto python3-mistune python3-nbformat python3-ntlm-auth python3-parameterized python3-pbr
  python3-requests-ntlm python3-singledispatch python3-speaklater python3-stem python3-tenacity python3-
  ruby-atomic ruby-thread-safe ruby2.7 ruby2.7-dev starkiller zaproxy
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  logstash
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 327 MB of archives.
After this operation, 579 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 logstash amd64 1:8.7.0-1 [327 MB]
Fetched 327 MB in 1min 11s (4,633 kB/s)
Selecting previously unselected package logstash.
(Reading database ... 421868 files and directories currently installed.)
Preparing to unpack .../logstash_1%3a8.7.0-1_amd64.deb ...
Unpacking logstash (1:8.7.0-1) ...
Setting up logstash (1:8.7.0-1) ...

┌──(root💀kali)-[~]
└─#
```

```
┌──(root💀kali)-[~]
└─# sudo systemctl start logstash

┌──(root💀kali)-[~]
└─# sudo systemctl enable logstash
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /lib/sy

┌──(root💀kali)-[~]
└─# sudo systemctl enable logstash

┌──(root💀kali)-[~]
└─# sudo systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/lib/systemd/system/logstash.service; enabled; preset: disabled)
     Active: active (running) since Mon 2023-04-03 22:14:53 IST; 34s ago
   Main PID: 5160 (java)
      Tasks: 22 (limit: 4594)
     Memory: 381.5M
        CPU: 55.095s
     CGroup: /system.slice/logstash.service
             └─5160 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.headless

Apr 03 22:14:53 kali systemd[1]: Started logstash.service - logstash.
Apr 03 22:14:53 kali logstash[5160]: Using bundled JDK: /usr/share/logstash/jdk
lines 1-12/12 (END)
```

FIGURE 49 - INSTALLING AND ENABLING LOGSTATSH

Step-18-> Now install and enable Kibana use the commands **sudo apt-get install kibana**, **sudo systemctl start kibana**, **sudo systemctl enable kibana**, **sudo systemctl status kibana**.

```
└─# sudo apt-get install kibana
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  bluez-firmware catfish cryptsetup-run faraday-client fastjar firmware-atheros firmware-brcm80211
  firmware-realtek firmware-sof-signed firmware-ti-connectivity firmware-zd1211 fonts-roboto-slab
  kali-wallpapers-2021.4 libarmadillo10 libatk1.0-data libavfilter7 libavformat58 libavresample4 l
  libev4 libexporter-tiny-perl libflac8 libfmt7 libgdal28 libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.
  libigdgmm11 liblist-moreutils-perl liblist-moreutils-xs-perl liblttng-ust-ctl4 liblttng-ust0 lib
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stre
  libopts25 libperl5.32 libpoppler102 libpostproc55 libproj19 libprotobuf23 libpython3.9 libpython
  librest-0.7-0 libsrt1.4-gnutls libswscale5 libtbb2 libtiff5 liburcu6 liburing1 libwebsockets16 l
  nginx-common nginx-core odbcinst odbcinst1debian2 openjdk-11-jre perl-modules-5.32 python-mpltoo
  python3-editor python3-exif python3-fastjsonschema python3-gevent python3-gevent-websocket pytho
  python3-m2crypto python3-mistune python3-nbformat python3-ntlm-auth python3-parameterized python
  python3-requests-ntlm python3-singledispatch python3-speaklater python3-stem python3-tenacity py
  ruby-atomic ruby-thread-safe ruby2.7 ruby2.7-dev starkiller zaproxy
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  kibana
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 240 MB of archives.
After this operation, 629 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 kibana amd64 8.7.0 [240 MB]
11% [1 kibana 32.4 MB/240 MB 14%]
Fetched 240 MB in 60s (3,980 kB/s)
Selecting previously unselected package kibana.
(Reading database ... 436144 files and directories currently installed.)
Preparing to unpack .../kibana_8.7.0_amd64.deb ...
Unpacking kibana (8.7.0) ...
Setting up kibana (8.7.0) ...
Creating kibana group ... OK
Creating kibana user ... OK
Created Kibana keystore in /etc/kibana/kibana.keystore
```

```
┌──(root💀kali)-[~]
└─# sudo systemctl start kibana

┌──(root💀kali)-[~]
└─# sudo systemctl enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /lib/sy

┌──(root💀kali)-[~]
└─# sudo systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; enabled; preset: disabled)
     Active: active (running) since Mon 2023-04-03 23:35:00 IST; 33s ago
       Docs: https://www.elastic.co
   Main PID: 13539 (node)
      Tasks: 11 (limit: 4594)
     Memory: 254.5M
        CPU: 28.669s
     CGroup: /system.slice/kibana.service
             └─13539 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/..

Apr 03 23:35:00 kali systemd[1]: Started kibana.service - Kibana.
Apr 03 23:35:09 kali kibana[13539]: [2023-04-03T23:35:09.227+05:30][INFO ][node] Kib
```

FIGURE 50 - INSTALLING AND ENABLING KIBANA SERVER

Step-19-> Now edit the kibana.yml file using **sudo nano /etc/kibana/kibana.yml** command. Uncomment these lines: **server.port:5601, server.host: "localhost" to server.host: "0.0.0.0" elasticsearch.hosts: ["http://localhost:9200"]**

FIGURE 51 - EDITING KIBANA.YML

Step-20-> Restart kibana service using **sudo systemctl restart kibana** command

FIGURE 52 - RESTARTING KIBANA SERVER

Step-21-> Now install filebeat using **sudo apt-get install filebeat** command.

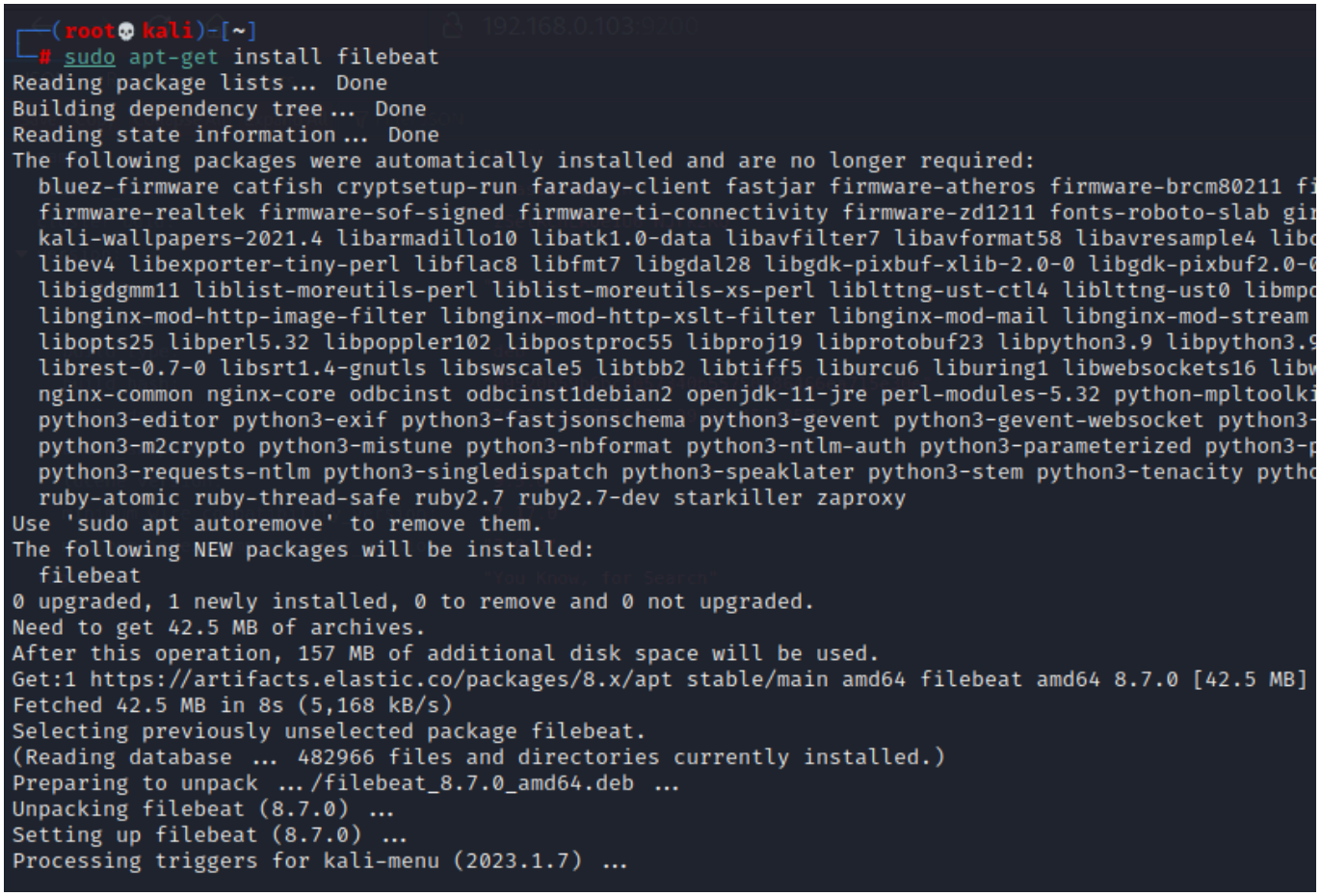


FIGURE 53 - INSTALLING FILEBEAT

Step-22-> Now edit filebeat.yml using **sudo nano /etc/filebeat/filebeat.yml** command. comment this line: **output.elasticsearch: Array of hosts to connect to. hosts: ["localhost:9200"]**. Uncomment this line: **output.logstash hosts: ["0.0.0.0:5044"]**

```
┌──(root💀kali)-[~]
└─# sudo nano /etc/filebeat/filebeat.yml
```

```
# ------------------------------ Elasticsearch Output ------------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
 #hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  #username: "elastic"
  #password: "changeme"

# ------------------------------ Logstash Output -------------------------------
output.logstash:
  # The Logstash hosts
   hosts: ["localhost:5044"]

  # Optional SSL. By default is off.
```

FIGURE 54 - EDITING KIBANA.YML

Step-23-> Now enable filebeat modules for the system using **sudo filebeat modules enable system** command

```
┌──(root💀kali)-[~]
└─# sudo filebeat modules enable system
Enabled system

┌──(root💀kali)-[~]
└─# sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["0.0.0.0:9200"]'
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.

┌──(root💀kali)-[~]
└─# sudo systemctl start filebeat

┌──(root💀kali)-[~]
└─# sudo systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.
```

FIGURE 55 - STARTING AND ENABLING FILEBEAT

Step-24-> Now run the curl command **curl -X GET** http://{your_ip_address}:9200/_cat/indices?v.

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ curl -X GET http://192.168.1.10:9200/_cat/indices?v
health status index                                uuid                   pri rep docs.count docs.dele
ted store.size pri.store.size
yellow open   .ds-filebeat-8.7.0-2023.04.03-000001 5YWVCG2DSTar21GySmMwJQ   1   1          0
  0       225b           225b
```

FIGURE 56 - CULR COMMAND TO GET ALL INDICES

Step-25-> Now open any browser, type url http://{your_ip_address}:9200/_cat/indices?v and check if you get the same output.

FIGURE 57 - SAME COMMAND IN BROWSER

Step-26-> Now change the same url to http://{your_ip_address}:5601.

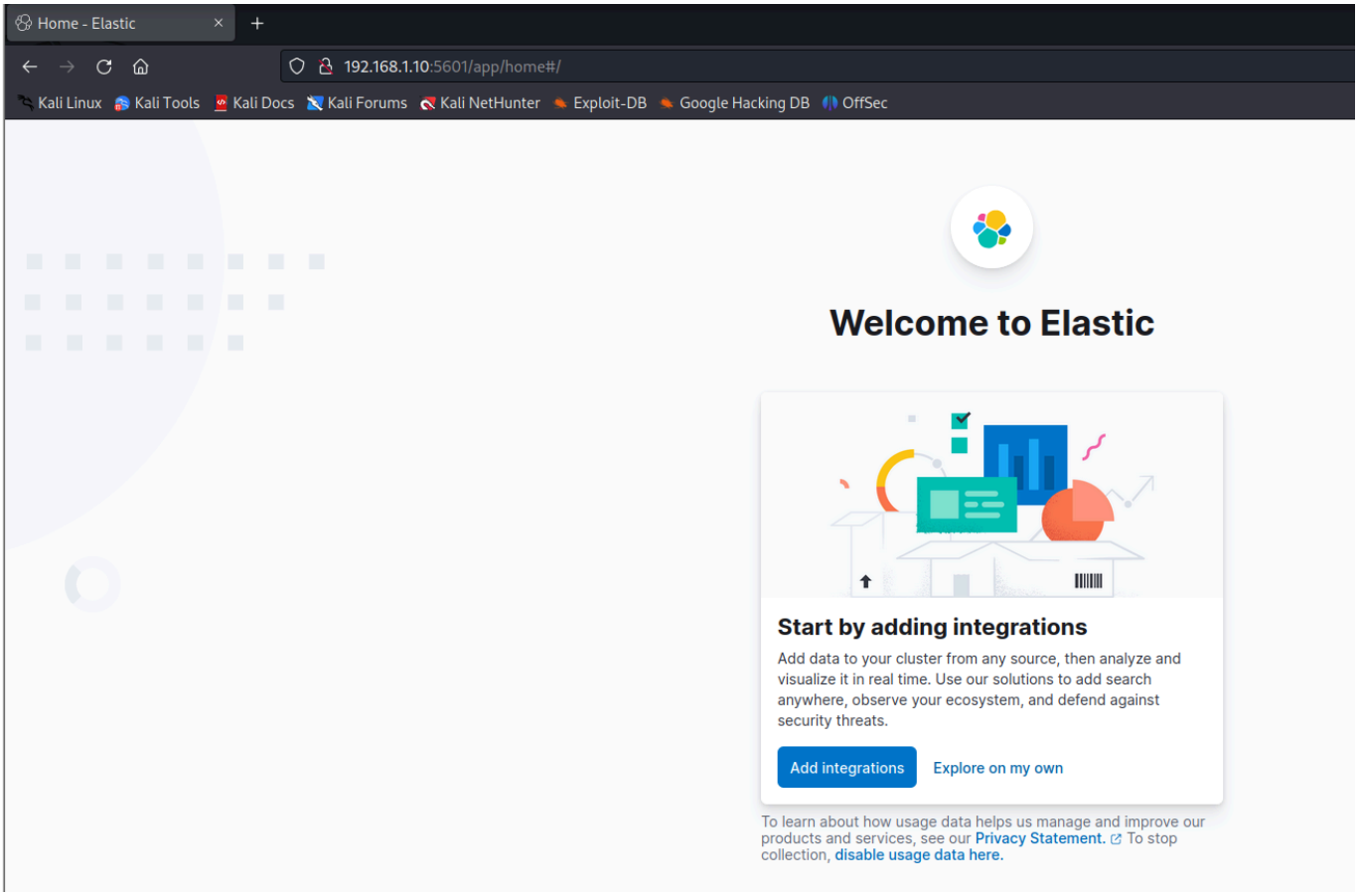


FIGURE 58 - ELASTIC SEARCH RUNNING ON PORT 5601

Practical 8: Install and Configure GrayLog on Linux

Step 1-> Run **sudo apt-get update** & **sudo apt-get upgrade** commands before procceding further

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.2 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.3 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [44.5
MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [116 kB]
Get:5 http://kali.download/kali kali-rolling/non-free amd64 Packages [217 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [9
27 kB]
Fetched 65.1 MB in 37s (1,749 kB/s)
Reading package lists ... Done
```

Figure 59 - updating packages

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get upgrade
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Calculating upgrade ... Done
The following packages were automatically installed and are no longer require
d:
  bluez-firmware firmware-ath9k-htc firmware-atheros firmware-brcm80211
  firmware-intel-sound firmware-iwlwifi firmware-libertas firmware-realtek
  firmware-sof-signed firmware-ti-connectivity firmware-zd1211
  kali-linux-firmware
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  kali-desktop-xfce libcurl3-nss libgtk-4-1 libgtk-4-bin libpocl2
  libpocl2-common linux-image-amd64 nginx pocl-opencl-icd python3-pypykatz
The following packages will be upgraded:
  adduser apache2 apache2-bin apache2-data apache2-utils bulk-extractor
  cryptcat curl dirbuster e2fsprogs easy-rsa exploitdb fakeroot faraday
  firefox-esr ipp-usb kali-desktop-core kali-linux-core kali-linux-default
  kali-linux-firmware kali-linux-headless kali-menu kali-system-cli
  kali-system-core kali-system-gui kali-tools-top10 laudanum
  libapache2-mod-php8.2 libbrlapi0.8 libc-bin libc-dev-bin libc-devtools
  libc-l10n libc6 libc6-dev libc6-i386 libcom-err2 libcups2 libcurl3-gnutls
```

Figure 60 - upgrading packages

Step 2-> Install jdk and jre using **sudo apt-get install default-jdk default-jre** command.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install default-jdk default-jre
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
default-jre is already the newest version (2:1.17-74).
default-jre set to manually installed.
The following packages were automatically installed and are no longer require
d:
  bluez-firmware firmware-ath9k-htc firmware-atheros firmware-brcm80211
  firmware-intel-sound firmware-iwlwifi firmware-libertas firmware-realtek
  firmware-sof-signed firmware-ti-connectivity firmware-zd1211
  kali-linux-firmware
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  default-jdk-headless openjdk-17-jdk openjdk-17-jdk-headless
Suggested packages:
  openjdk-17-demo openjdk-17-source visualvm
The following NEW packages will be installed:
  default-jdk default-jdk-headless openjdk-17-jdk openjdk-17-jdk-headless
0 upgraded, 4 newly installed, 0 to remove and 10 not upgraded.
Need to get 234 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-rolling/main amd64 openjdk-17-jdk-headle
ss amd64 17.0.6+10-1 [230 MB]
Get:2 http://kali.download/kali kali-rolling/main amd64 default-jdk-headless
```

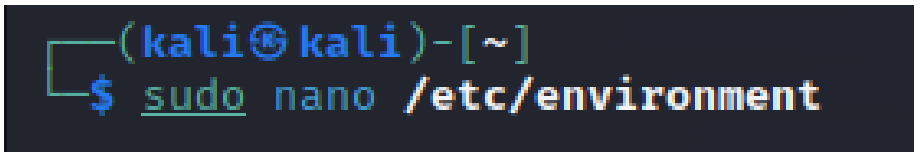
Step 3-> Check if java is installed using **java -version** command.

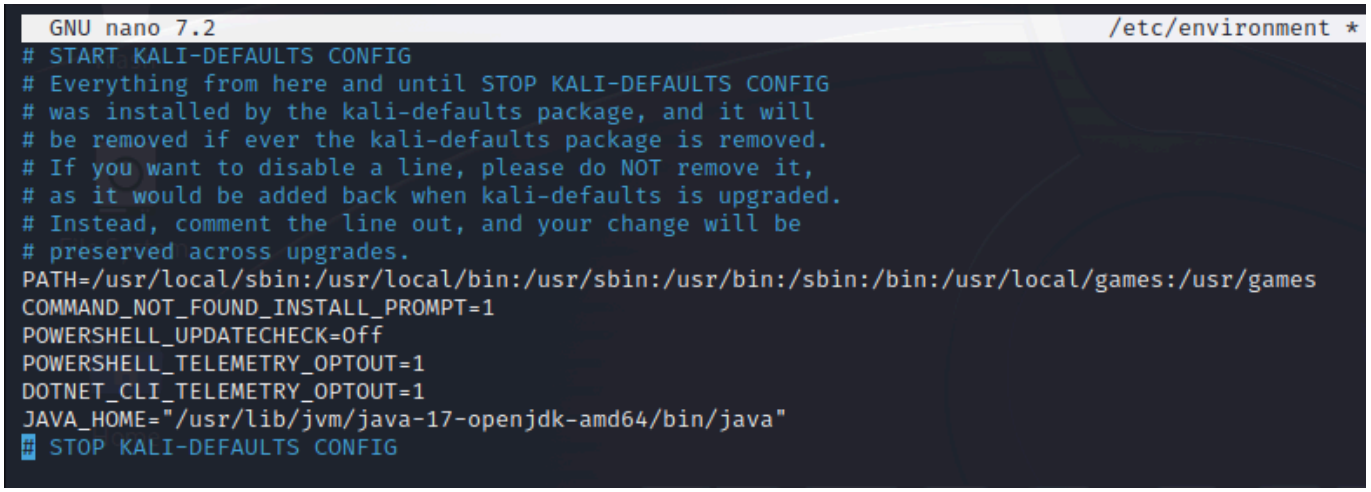
```
┌──(kali㉿kali)-[~]
└─$ java -version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
openjdk version "17.0.6" 2023-01-17
OpenJDK Runtime Environment (build 17.0.6+10-Debian-1)
OpenJDK 64-Bit Server VM (build 17.0.6+10-Debian-1, mixed mode, sharing)
```

Step 4-> Select java version to use with **update-alternatives --config** java command.

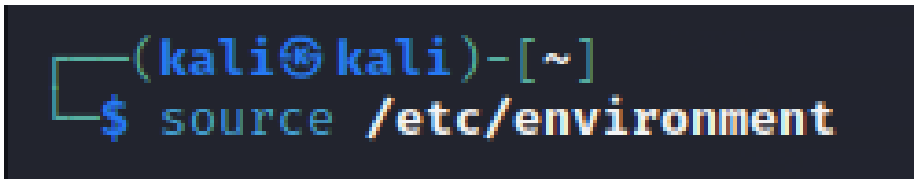
```
┌──(kali㉿kali)-[~]
└─$ update-alternatives --config java
There is 1 choice for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority   Status
------------------------------------------------------------
* 0            /usr/lib/jvm/java-17-openjdk-amd64/bin/java   1711      auto mode
  1            /usr/lib/jvm/java-17-openjdk-amd64/bin/java   1711      manual mode

Press <enter> to keep the current choice[*], or type selection number: 0
```

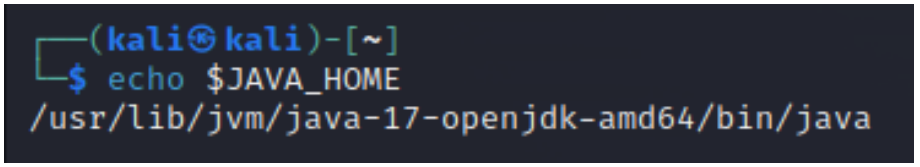
Step 5-> Run **sudo nano /etc/environment** command to edit the environment variables. Append the line **JAVA_HOME="/usr/lib/jvm/java-17-openjdk-amd64/bin/java"**



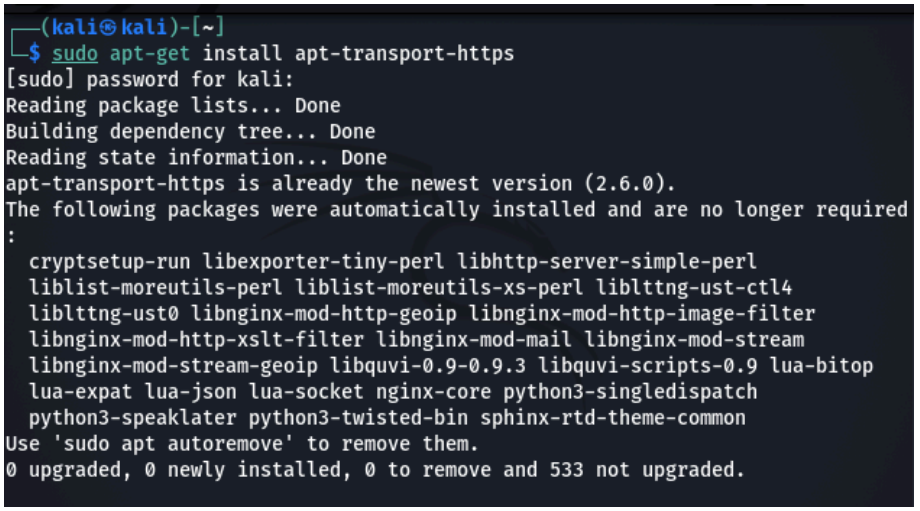


Step 6-> Now save the environment variables using **source /etc/environment** command. Check if **JAVA_HOME** variable is correctly set using **echo &JAVA_HOME**.





Step 7-> Now run **sudo apt-get install apt-transport-https** command.



Step 8-> Run the command **wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –** .

```
┌──(kali㉿kali)-[~]
└─$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-k
ey add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead
(see apt-key(8)).
OK
```

Step 9-> Run the command **echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list**.

```
┌──(kali㉿kali)-[~]
└─$ echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee
 /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
```

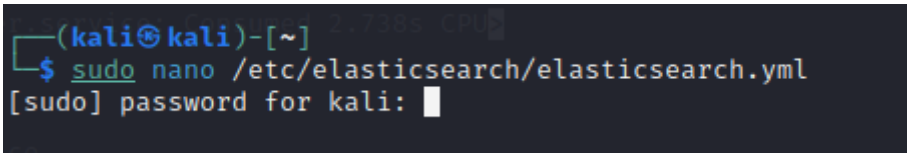
Step 10-> Now run **sudo apt-get update** command again.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Hit:2 http://kali.download/kali kali-rolling InRelease
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [109 kB]
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Contents (deb) [3,
177 kB]
Fetched 3,300 kB in 5s (637 kB/s)
Reading package lists... Done
W: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: Key is stored i
n legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-
key(8) for details.
```

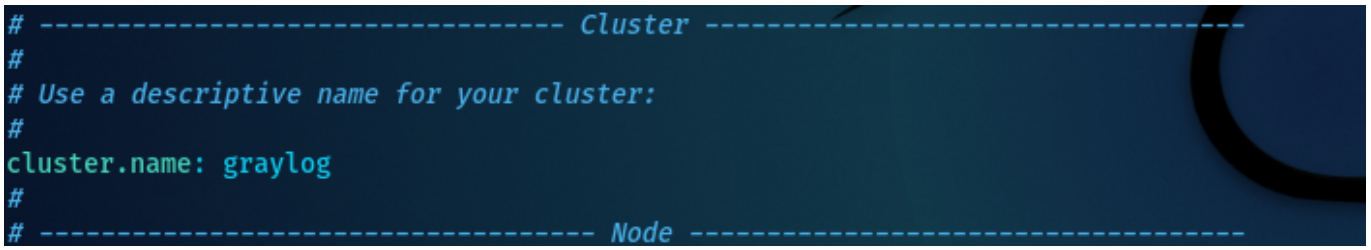
Step 11-> Now install elasticsearch using **sudo apt-get install elasticsearch**

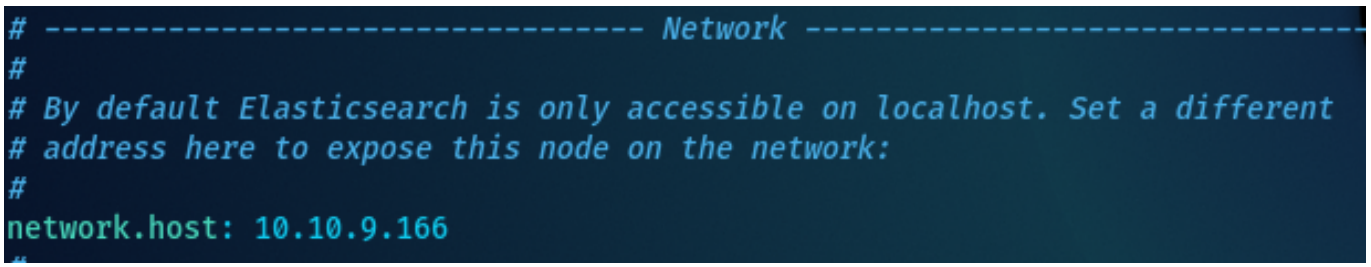
```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install elasticsearch
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
elasticsearch is already the newest version (7.17.9).
The following packages were automatically installed and are no longer required:
  cryptsetup-run libexporter-tiny-perl libhttp-server-simple-perl liblist-moreut
  libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-
  libquvi-0.9-0.9.3 libquvi-scripts-0.9 lua-bitop lua-expat lua-json lua-socket
  sphinx-rtd-theme-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 533 not upgraded.
```

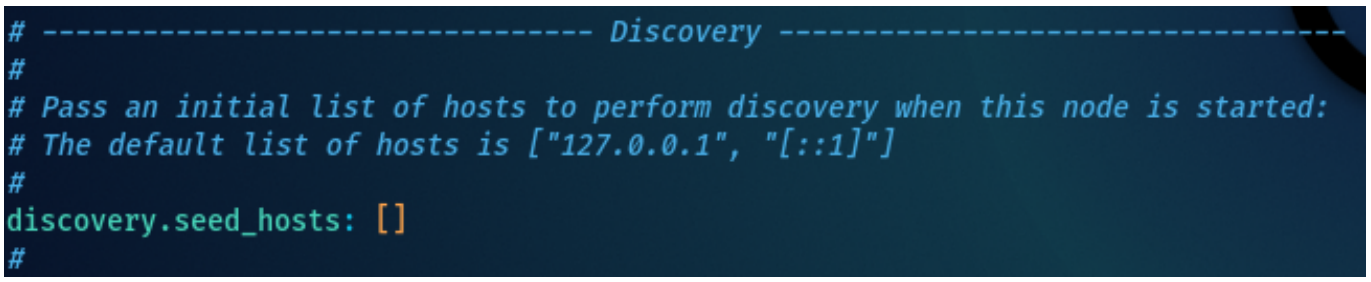
Step 12-> Now run **sudo nano /etc/elasticsearch/elasticsearch.yml** to edit the yml file.

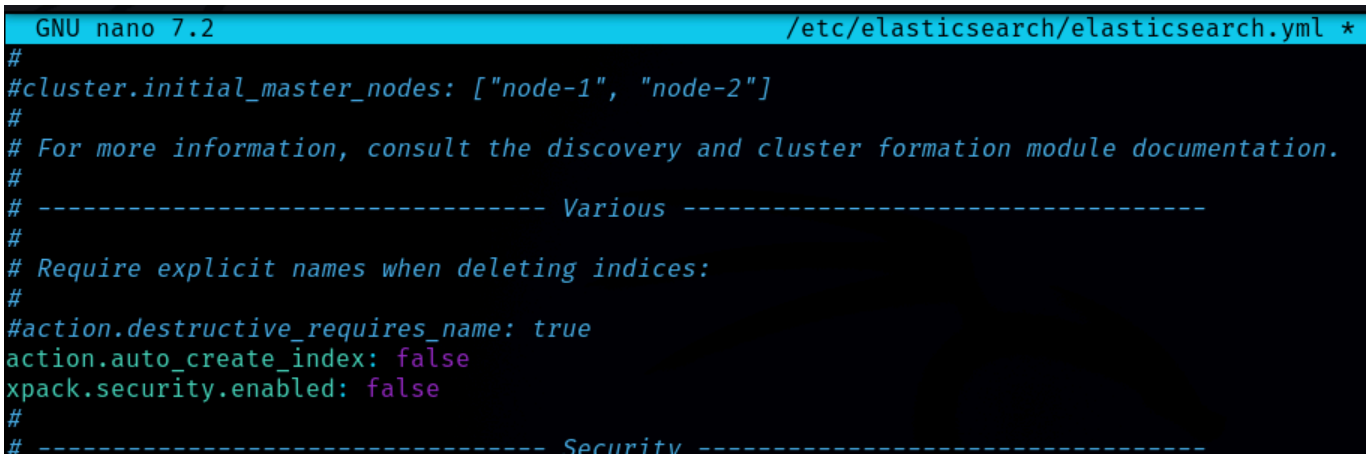


Step 13-> Add the lines **cluster.name: graylog**, **network.host: your_linux_ip_address**, **discovery.seed_host: []**. Add **action.auto_create_index: false, xpack.security.enabled: false**.

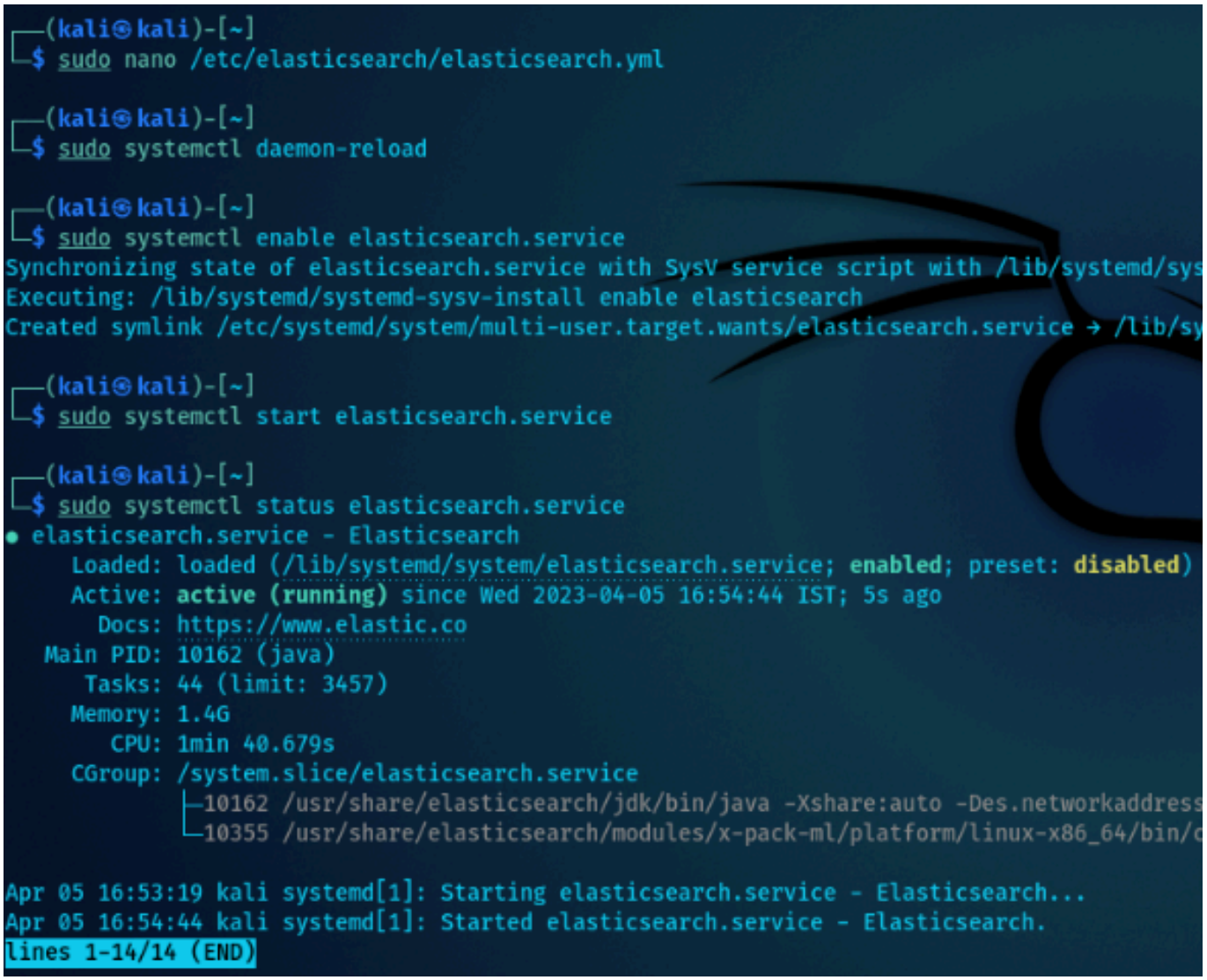






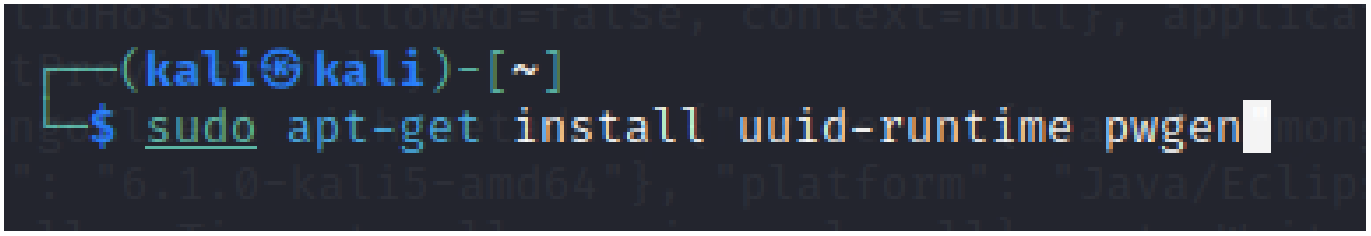


Step 14-> Run **sudo systemctl daemon-reload, sudo systemctl enable elasticsearch.service, sudo systemctl start elasticsearch.service, sudo systemctl status elasticsearch.service** commands in the given order.

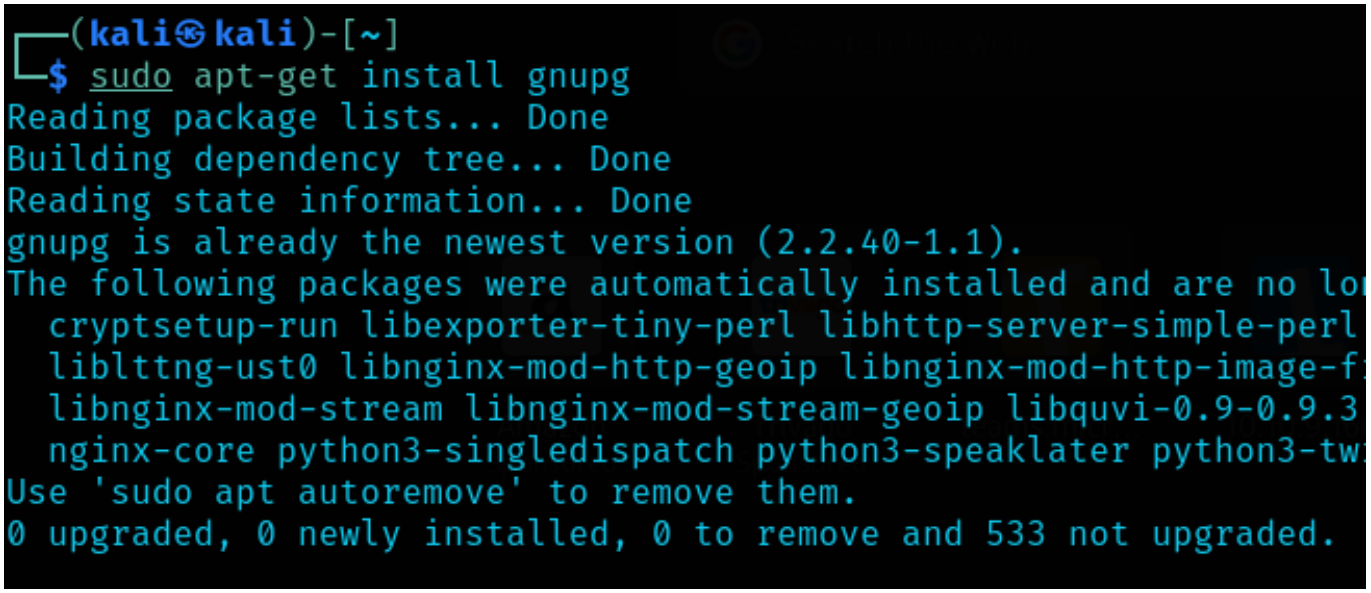


Step 15-> Run the curl command **curl -X GET "localhost:9200/?pretty"** where localhost is your ip address.

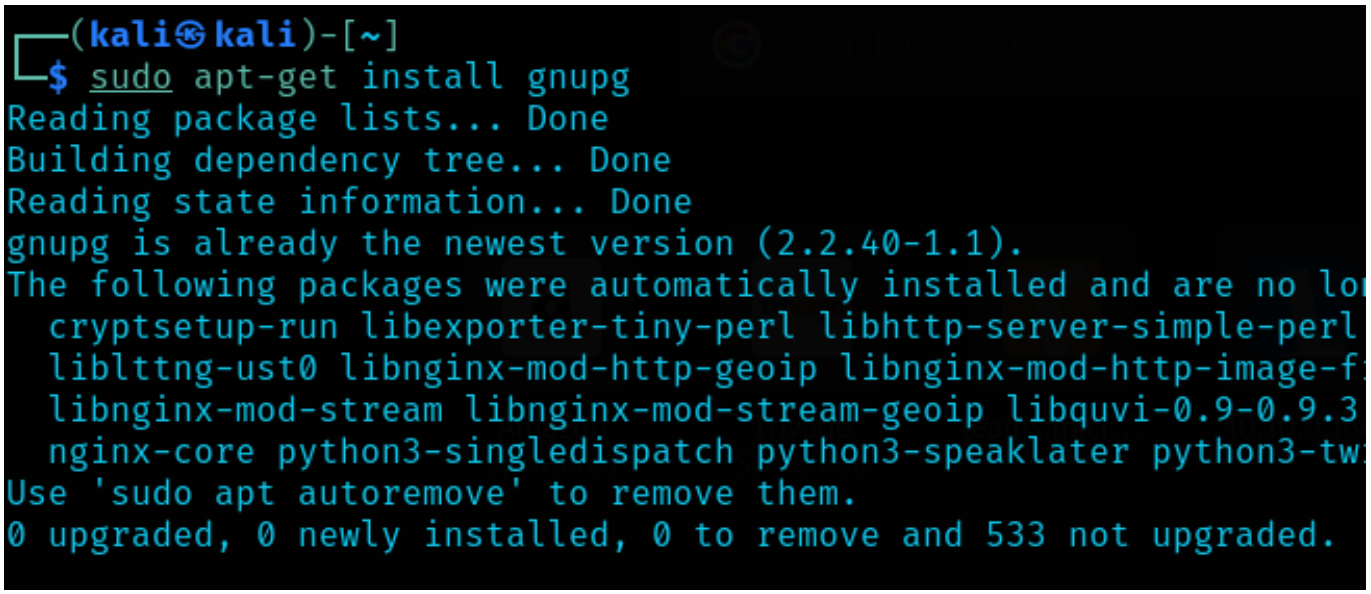
Step 16-> Now run **sudo apt-get install uuid-runtime pwgen** command.



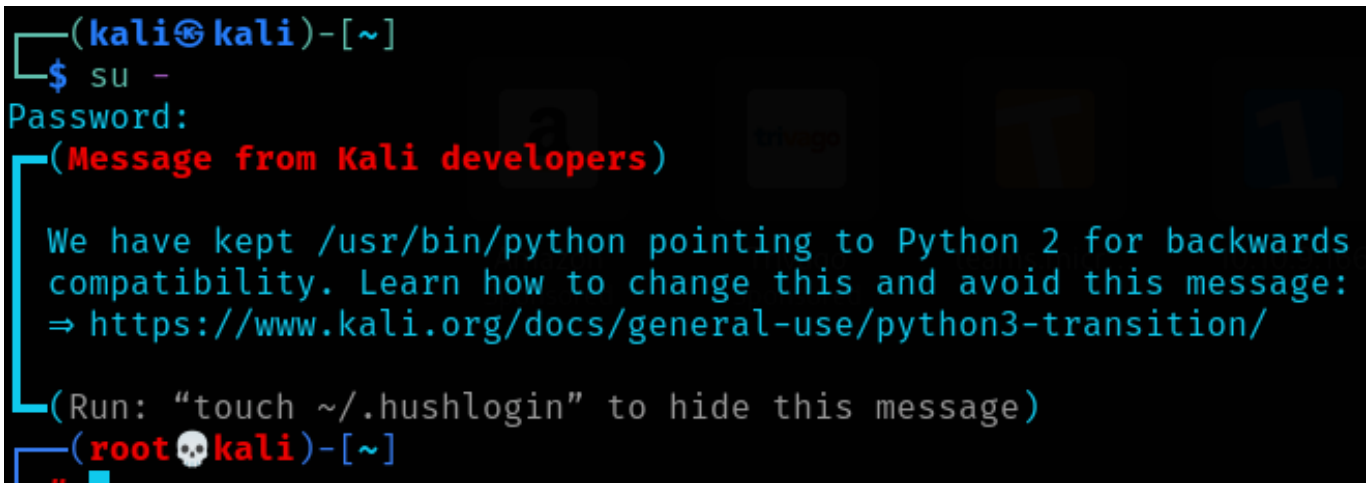
Step 17-> Now run **sudo apt-get install gnupg** command.



Step 18-> Now install gnupg using **sudo apt-get install gnupg** command.



Step 19-> Now login to root user with **su –** command.



Step 20-> Now run the command wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -

Step 21-> Next run command **echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list**. Exit from root access using exit command.

```
┌──(root💀kali)-[~]
└─# echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu ja
mmy/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongod
b-org-6.0.list
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu jammy/mongod
b-org/6.0 multiverse


┌──(root💀kali)-[~]
└─# exit
```

Step 22-> Run **sudo apt-get update** again

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Ign:2 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 InRelease
Get:3 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 Release [
3,094 B]
Get:5 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 Release.g
pg [801 B]
Hit:4 http://kali.download/kali kali-rolling InRelease
Get:6 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0/multivers
e amd64 Packages [14.3 kB]
Get:7 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0/multivers
e arm64 Packages [12.6 kB]
Fetched 30.8 kB in 2s (13.7 kB/s)
Reading package lists... Done
W: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease:
Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see t
he DEPRECATION section in apt-key(8) for details.
W: https://repo.mongodb.org/apt/ubuntu/dists/jammy/mongodb-org/6.0/Releas
e.gpg: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg)
, see the DEPRECATION section in apt-key(8) for details.
```

Step 23-> Now install mongodb using **sudo apt-get install mongodb-org** command

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install mongodb-org
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer rec
uired:
  cryptsetup-run libexporter-tiny-perl libhttp-server-simple-perl
  liblist-moreutils-perl liblist-moreutils-xs-perl liblttng-ust-ctl4
  liblttng-ust0 libnginx-mod-http-geoip libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip libquvi-0.9-0.9.3 libquvi-scripts-0.9
  lua-bitop lua-expat lua-json lua-socket nginx-core
  python3-singledispatch python3-speaklater python3-twisted-bin
  sphinx-rtd-theme-common
Use 'sudo apt autoremove' to remove them.
```

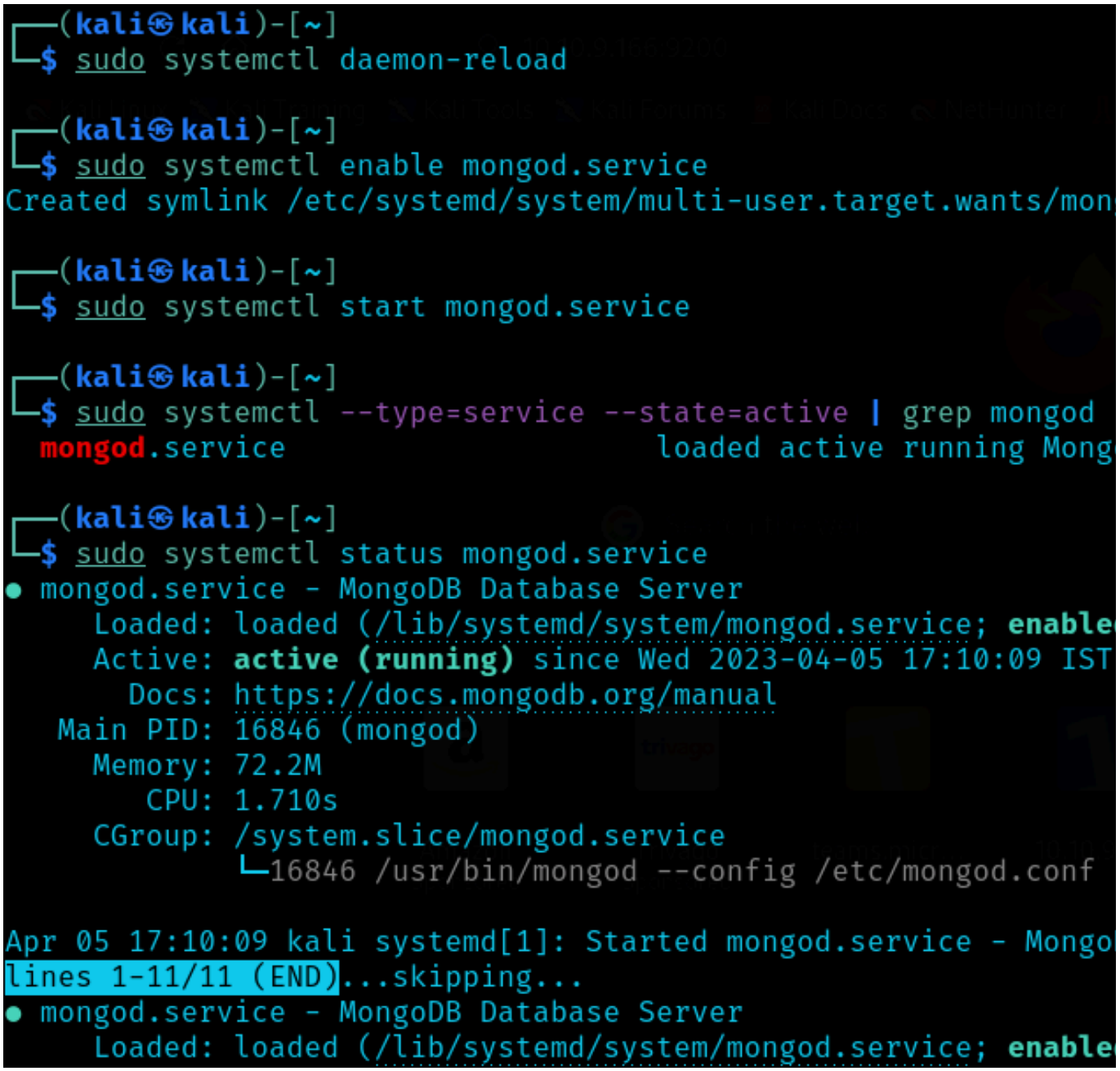
Step 24-> Enter root user using **su –** again. Run command **wget -qO- 'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0xf5679a222c647c87527c2f8cb00a0bd1e2c63c11' | sudo apt-key add -**.

```
┌──(kali㉿kali)-[~]
└─$ su -
Password:
┌──(Message from Kali developers)
│
│ We have kept /usr/bin/python pointing to Python 2 for backwards
│ compatibility. Learn how to change this and avoid this message:
│ ⇒ https://www.kali.org/docs/general-use/python3-transition/
│
└──(Run: “touch ~/.hushlogin” to hide this message)
┌──(root💀kali)-[~]
└─# wget -qO- 'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0xf56
79a222c647c87527c2f8cb00a0bd1e2c63c11' | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d ins
tead (see apt-key(8)).
OK
```

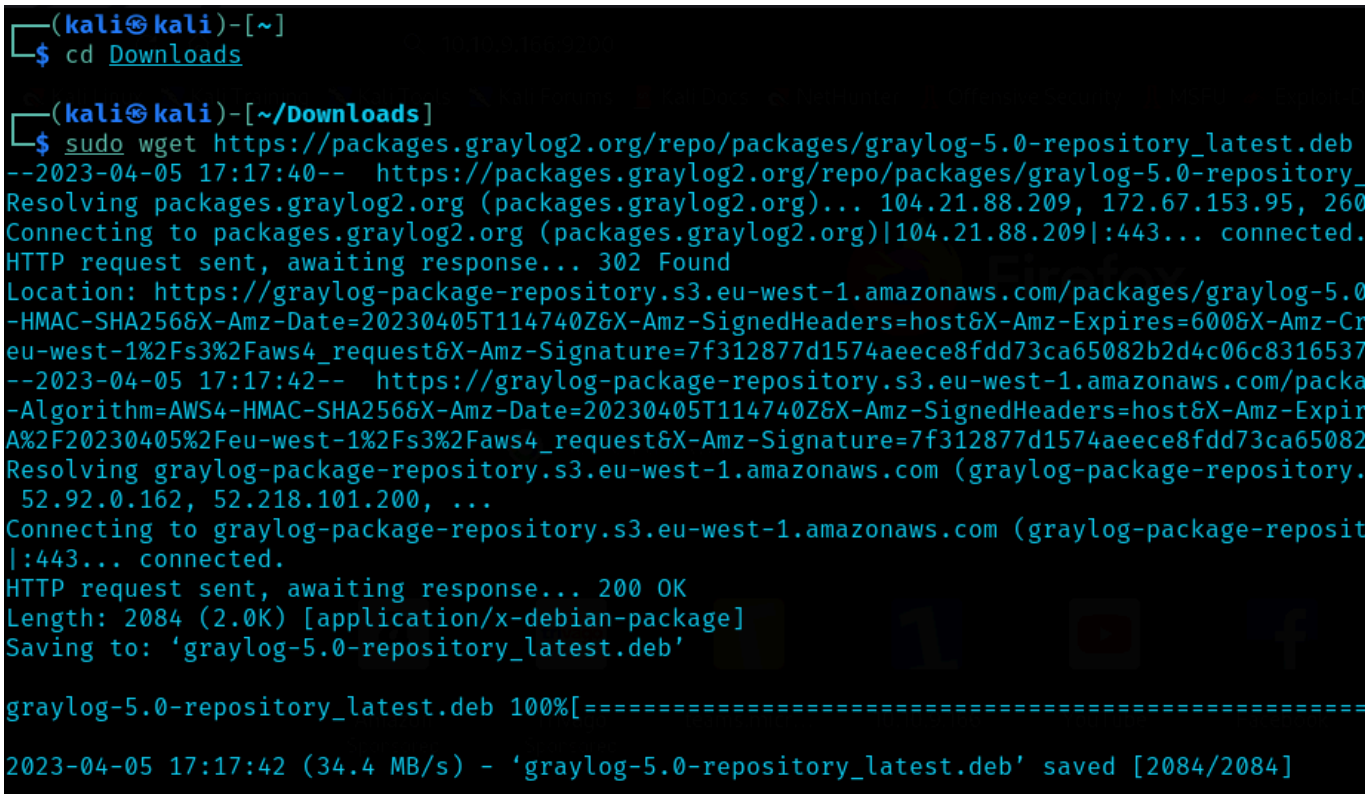
Step 25-> Exit root access using **exit** command

```
┌──(root💀kali)-[~]
└─# exit
```

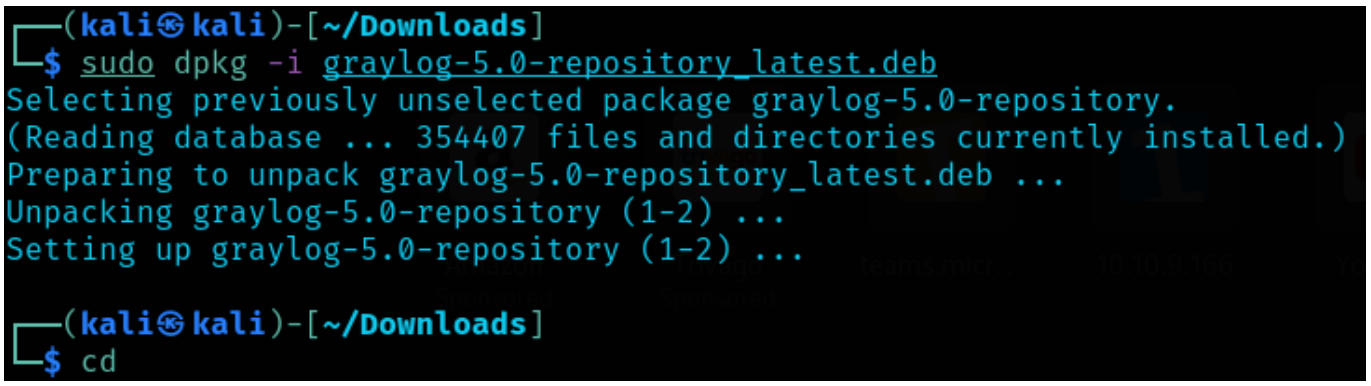
Step 26-> Now run commands **sudo systemctl daemon-reload sudo systemctl enable mongod.service sudo systemctl start mongod.service sudo systemctl --type=service --state=active | grep mongod sudo systemctl status mongod.service**.



Step 27-> Change directory to Dowloads using **cd Downloads** .Now download graylog Debian package using **sudo wget https://packages.graylog2.org/repo/packages/graylog-5.0-repository_latest.deb** command



Step 28-> Install graylog server using command **sudo dpkg -i graylog-5.0-repository_latest.deb**. Change directory once more using **cd** command.



Step 29-> Run **sudo apt-get update** once more.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Ign:2 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 InRelease
Hit:4 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 Release
Hit:3 http://kali.download/kali kali-rolling InRelease
Get:5 https://packages.graylog2.org/repo/debian stable InRelease [54.5 kB
Get:7 https://packages.graylog2.org/repo/debian stable/5.0 amd64 Packages
Fetched 60.9 kB in 4s (15.6 kB/s)
Reading package lists... Done
W: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: k
gpg), see the DEPRECATION section in apt-key(8) for details.
W: https://repo.mongodb.org/apt/ubuntu/dists/jammy/mongodb-org/6.0/Releas
rusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```

Step 30-> Now install graylog-server using **sudo apt-get install graylog-server** command.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install graylog-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer
  cryptsetup-run libexporter-tiny-perl libhttp-server-simple-perl lib
  liblttng-ust0 libnginx-mod-http-geoip libnginx-mod-http-image-filte
  libnginx-mod-stream libnginx-mod-stream-geoip libquvi-0.9-0.9.3 lib
  nginx-core python3-singledispatch python3-speaklater python3-twiste
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  util-linux util-linux-extra
Suggested packages:
  util-linux-locales
The following NEW packages will be installed:
  graylog-server util-linux-extra
The following packages will be upgraded:
  util-linux
1 upgraded, 2 newly installed, 0 to remove and 532 not upgraded.
Need to get 300 MB of archives.
After this operation, 420 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Step 31-> Run the commands **sudo systemctl enable graylog-server.service sudo systemctl start graylog-server.service sudo systemctl status graylog-server.service**.

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable graylog-server.service
Synchronizing state of graylog-server.service with SysV service script with /lib/s
Executing: /lib/systemd/systemd-sysv-install enable graylog-server
Created symlink /etc/systemd/system/multi-user.target.wants/graylog-server.service

┌──(kali㉿kali)-[~]
└─$ sudo systemctl start graylog-server.service
```

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status graylog-server.service
● graylog-server.service - Graylog server
     Loaded: loaded (/lib/systemd/system/graylog-server.service; enabled; pr>
     Active: activating (auto-restart) (Result: exit-code) since Thu 2023-04>
       Docs: http://docs.graylog.org/
    Process: 102689 ExecStart=/usr/share/graylog-server/bin/graylog-server (>
   Main PID: 102689 (code=exited, status=1/FAILURE)
        CPU: 3.240s

Apr 20 14:50:17 kali systemd[1]: graylog-server.service: Main process exited>
Apr 20 14:50:17 kali systemd[1]: graylog-server.service: Failed with result >
Apr 20 14:50:17 kali systemd[1]: graylog-server.service: Consumed 3.240s CPU>
log file:
```

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status graylog-server.service
● graylog-server.service - Graylog server
     Loaded: loaded (/lib/systemd/system/graylog-server.service; enabled; preset: disabled)
     Active: active (running) since Wed 2023-04-05 17:35:25 IST; 1min 15s ago
       Docs: http://docs.graylog.org/
   Main PID: 24447 (graylog-server)
      Tasks: 20 (limit: 3457)
     Memory: 118.0M
        CPU: 47.987s
     CGroup: /system.slice/graylog-server.service
             ├─24447 /bin/sh /usr/share/graylog-server/bin/graylog-server
             └─24449 /usr/share/graylog-server/jvm/bin/java -Xms1g -Xmx1g -server -XX:+UseG1
```

Step 32-> Now edit the server.conf file of graylog server using **sudo nano /etc/graylog/server/server.conf** command.

```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/graylog/server/server.conf
```

Step 33-> Now open another terminal and run **echo -n "Enter Password: " && head -1 </dev/stdin | tr -d '\n' | sha256sum | cut -d" " -f1** to generate root password SHA2 hash and **pwgen -N 1 -s 96** to generate password secret. Save both in a text file.

```
┌──(kali㉿kali)-[~]
└─$ echo -n "Enter Password: " && head -1 </dev/stdin | tr -d '\n' | sha256sum | cut -d" " -f1
Enter Password: kali
fc5669b52ce4e283ad1d5d182de88ff9faec6672bace84ac2ce4c083f54fe2bc

┌──(kali㉿kali)-[~]
└─$

┌──(kali㉿kali)-[~]
└─$ pwgen -N 1 -s 96
p6aHIoWCkwAsIbHRGhoZmb3atG2mGfhUUFRh6gszvqFDqaSp1kW3p3IJ5LUzrnBDfijpirblKBFsgibvm5xburONSVWngbkG
```

Step 34-> Now run **sudo nano /etc/graylog/server/server.conf**. Use the keys you generated and paste **sha256** hash after root_password_sha2 and password_secret for **pwgen** key accordingly. Uncomment line **http_bind_address** and add **http_bind_address** = localhost:9000 (in place of localhost replace with your system ip). Uncomment this line **elasticsearch_hosts = http://localhost:9200/** (in place of localhost replace with your system ip) comment this line after above line by pressing enter and then type "#" http://user:password@192.168.4.84:19200. Save the file

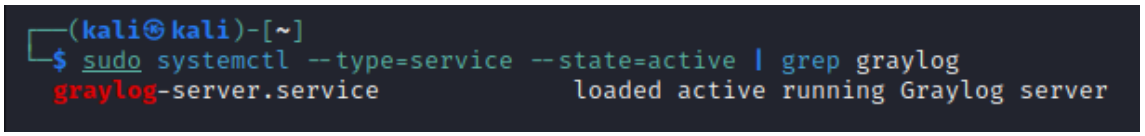
```
  GNU nano 7.2                  /etc/graylog/server/server.conf
# HTTP settings
###############

#### HTTP bind address
#
# The network interface used by the Graylog HTTP interface.
#
# This network interface must be accessible by all Graylog n
# using the Graylog web interface.
#
# If the port is omitted, Graylog will use port 9000 by defa
#
# Default: 127.0.0.1:9000
http_bind_address = 10.10.9.166:9000
#http_bind_address = [2001:db8::1]:9000

#### HTTP publish URI
```

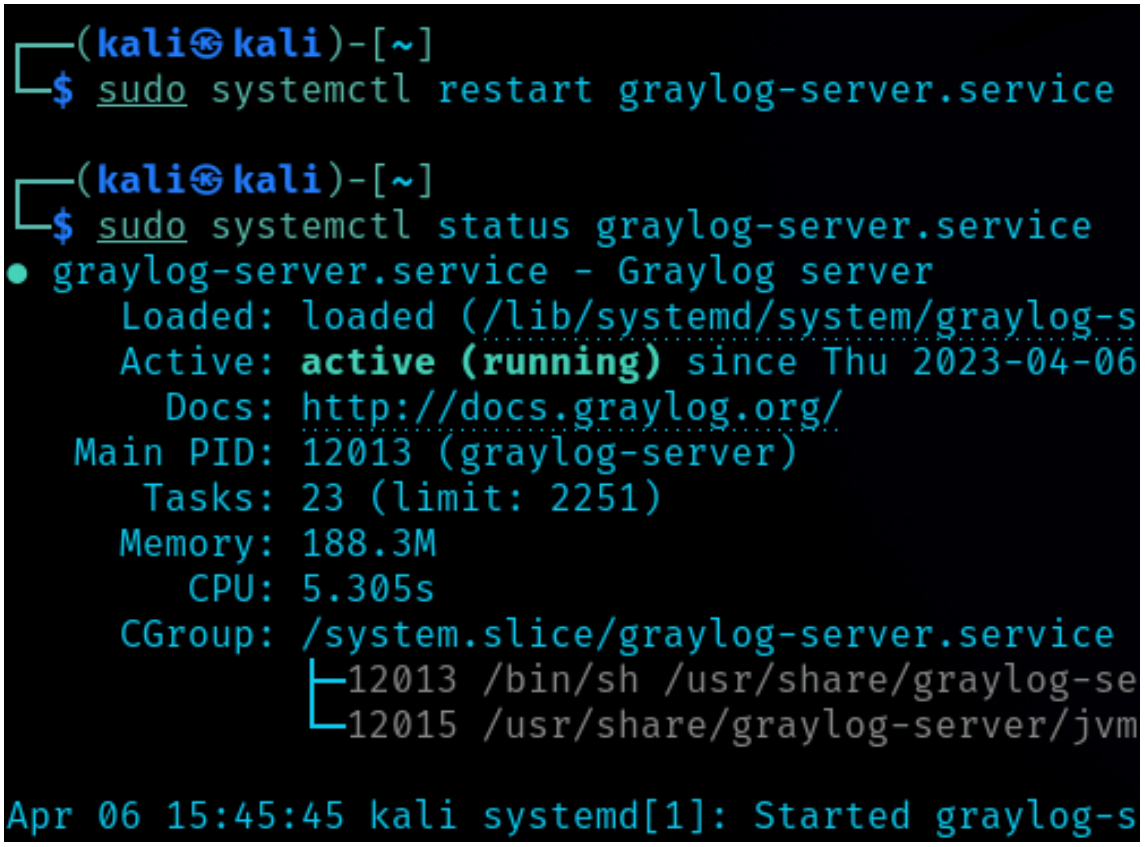
```
  GNU nano 7.2                  /etc/graylog/server/server.conf
#trusted_proxies = 127.0.0.1/32, 0:0:0:0:0:0:0:1/128

# List of Elasticsearch hosts Graylog should connect to.
# Need to be specified as a comma-separated list of valid
# If one or more of your elasticsearch hosts require authe
# requires authentication.
#
# Default: http://127.0.0.1:9200
elasticsearch_hosts = http://10.10.9.166:9200
#http://user:password@node2:19200

# Maximum number of attempts to connect to elasticsearch o
#
# Default: 0, retry indefinitely with the given delay unti
#elasticsearch_version_probe_attempts = 5
```

```
# Changing this value after installation will render all user sessions and encrypted values in the database invali
password_secret =J0KsT9LwA7ao3mfAvaveMDE1Zbz3h8LMTn4Cdierjcqj0WkQd8crDODtCC2yua9PxGWiCiCDA1pLbDQb7F7m37AQIFKgiIuW
```

```
# and put the resulting hash value into the following line
root_password_sha2 =fc5669b52ce4e283ad1d5d182de88ff9faec6672bace84ac2ce4c083f54fe2bc
```

Step 35-> Run the commands **sudo systemctl restart graylog-server.service, sudo systemctl --type=service --state=active | grep graylog, sudo systemctl restart graylog-server.service, sudo systemctl status graylog-server.service** commands in same order.





Step 36-> Now run **http://localhost:9000/dashboard** and check if graylog gui is shown as below in a browser.