Task: 05

Performed By: Tanish Choudhary

Title - Networking and Security Operations with SIEM, Forensics, and Traffic Analysis

Objective

The goal of this task is to help you:

- Design and calculate a subnet for a small network.
- Analyze network traffic patterns using packet capture tools.
- Troubleshoot network protocol issues in a simulated environment.
- Set up a SIEM system using ELK Stack for log monitoring.
- Simulate an incident and perform network forensics to investigate it.
- Conduct threat hunting using network and log data.

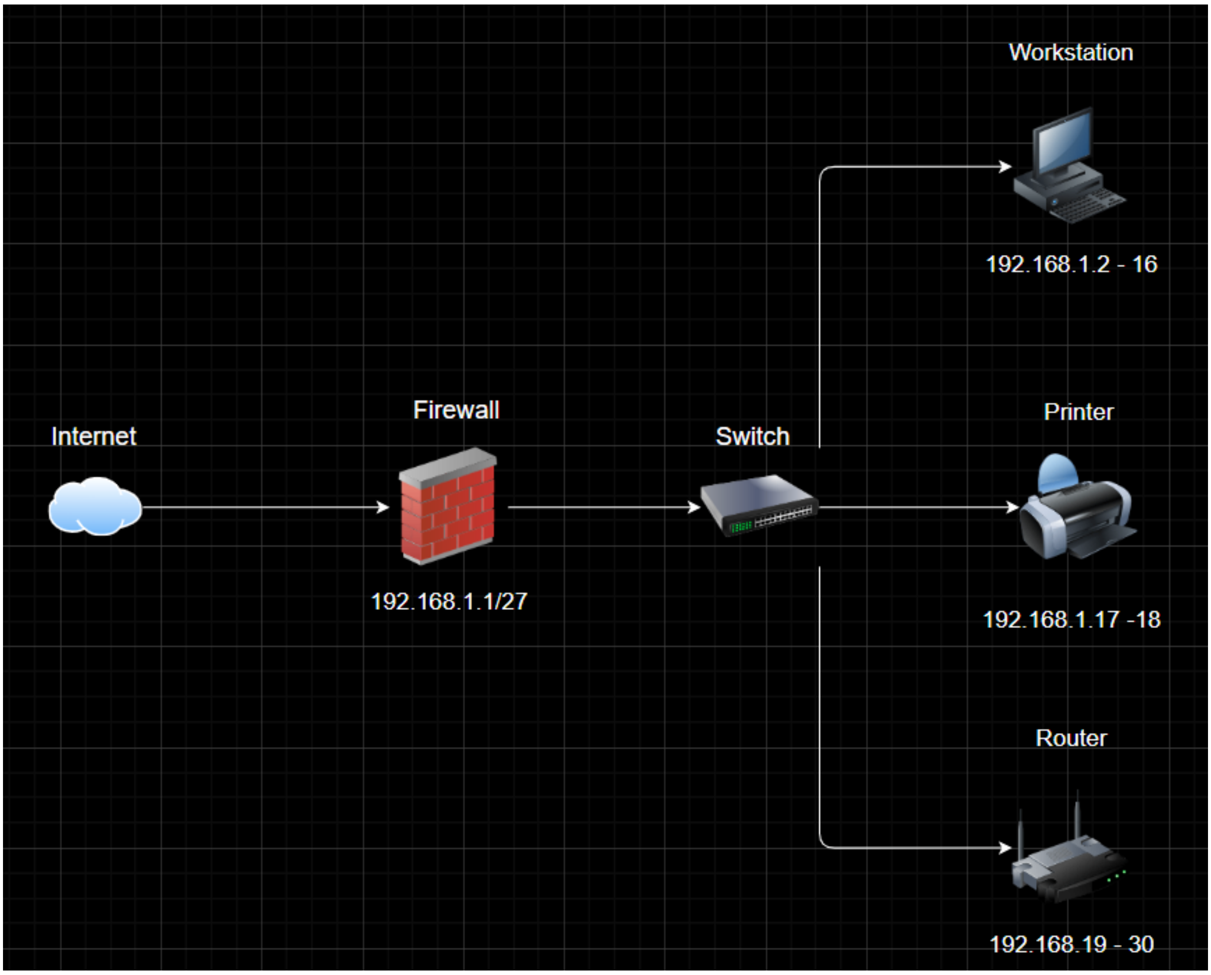
NETWORK SUBNET MASK

Range: **192.168.1.0/27**

Why: **32 addresses, 30 usable (enough for 20 devices)**

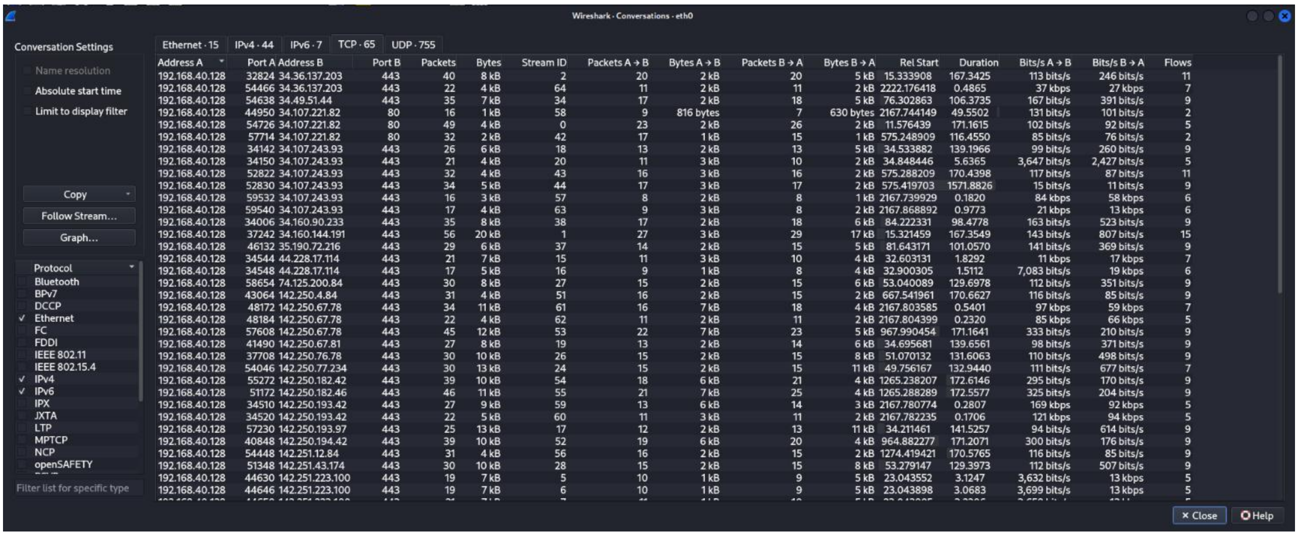
Subnet Mask: **255.255.255.224**

Broadcast: **192.168.1.31**



TRAFFIC ANALYSIS WITH WIRESHARK

1. **TCP** Conversations



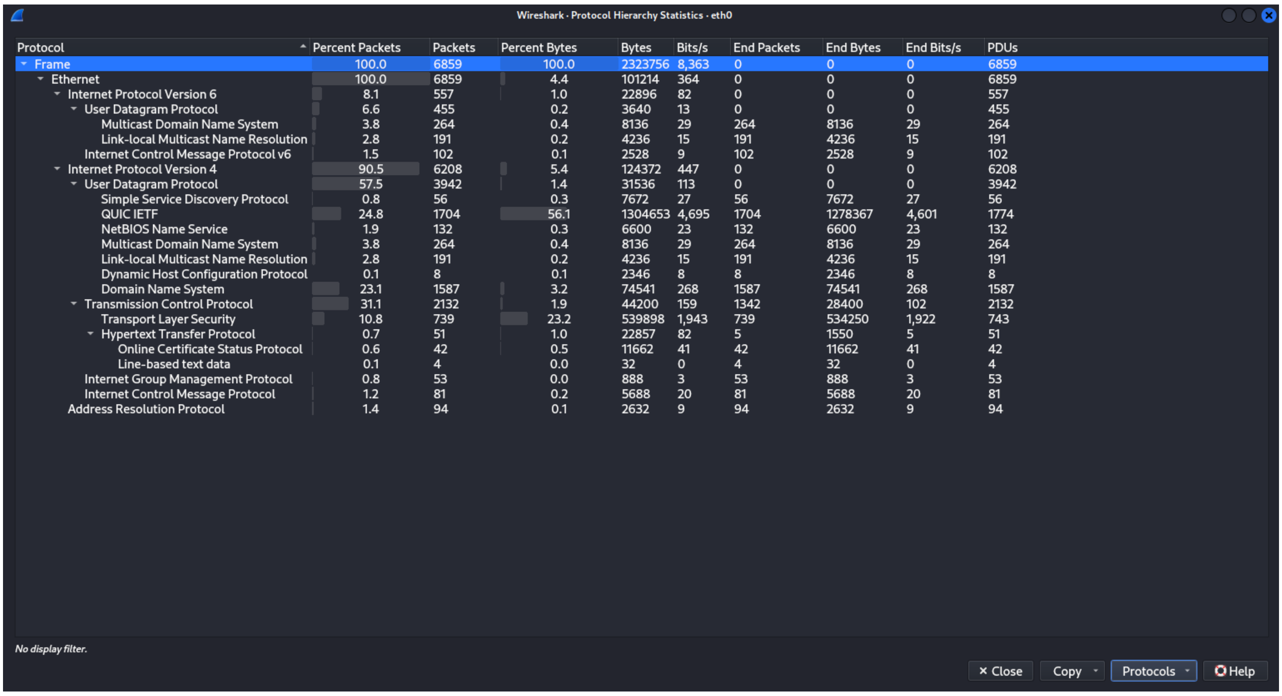
Conversation Settings

☐ Name resolution
☐ Absolute start time
☐ Limit to display filter

Protocol
☐ Bluetooth
☐ BPv7
☐ DCCP
✓ Ethernet
☐ FC
☐ FDDI
☐ IEEE 802.11
☐ IEEE 802.15.4
✓ IPv4
✓ IPv6
☐ IPX
☐ JXTA
☐ LTP
☐ MPTCP
☐ NCP
☐ openSAFETY

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A	Flows
192.168.40.128	32824	34.36.137.203	443	40	8 kB	2	20	2 kB	20	5 kB	15.333908	167.3425	113 bits/s	246 bits/s	11
192.168.40.128	54466	34.36.137.203	443	22	4 kB	64	11	2 kB	11	2 kB	2222.176418	0.4865	37 kbps	27 kbps	7
192.168.40.128	54638	34.49.51.44	443	35	7 kB	34	17	2 kB	18	5 kB	76.302863	106.3735	167 bits/s	391 bits/s	9
192.168.40.128	44950	34.107.221.82	80	16	1 kB	58	9	816 bytes	7	630 bytes	2167.744149	49.5502	131 bits/s	101 bits/s	2
192.168.40.128	54726	34.107.221.82	80	49	4 kB	0	23	2 kB	26	2 kB	11.576439	171.1615	102 bits/s	92 bits/s	5
192.168.40.128	57714	34.107.221.82	80	32	2 kB	42	17	1 kB	15	1 kB	575.248909	116.4550	85 bits/s	76 bits/s	2
192.168.40.128	34142	34.107.243.93	443	26	6 kB	18	13	2 kB	13	5 kB	34.533882	139.1966	99 bits/s	260 bits/s	9
192.168.40.128	34150	34.107.243.93	443	21	4 kB	20	11	3 kB	10	2 kB	34.848446	5.6365	3,647 bits/s	2,427 bits/s	5
192.168.40.128	52822	34.107.243.93	443	32	4 kB	43	16	3 kB	16	2 kB	575.288209	170.4398	117 bits/s	87 bits/s	11
192.168.40.128	52830	34.107.243.93	443	34	5 kB	44	17	3 kB	17	2 kB	575.419703	1571.8826	15 bits/s	11 bits/s	9
192.168.40.128	59532	34.107.243.93	443	16	3 kB	57	8	2 kB	8	1 kB	2167.739929	0.1820	84 kbps	58 kbps	6
192.168.40.128	59540	34.107.243.93	443	17	4 kB	63	9	3 kB	8	2 kB	2167.868892	0.9773	21 kbps	13 kbps	6
192.168.40.128	34006	34.160.90.233	443	35	8 kB	38	17	2 kB	18	6 kB	84.222331	98.4778	163 bits/s	523 bits/s	9
192.168.40.128	37242	34.160.144.191	443	56	20 kB	1	27	3 kB	29	17 kB	15.321459	167.3549	143 bits/s	807 bits/s	15
192.168.40.128	46132	35.190.72.216	443	29	6 kB	37	14	2 kB	15	5 kB	81.643171	101.0570	141 bits/s	369 bits/s	9
192.168.40.128	34544	44.228.17.114	443	21	7 kB	15	11	3 kB	10	4 kB	32.603131	1.8292	11 kbps	17 kbps	7
192.168.40.128	34548	44.228.17.114	443	17	5 kB	16	9	1 kB	8	4 kB	32.900305	1.5112	7,083 bits/s	19 kbps	6
192.168.40.128	58654	74.125.200.84	443	30	8 kB	27	15	2 kB	15	6 kB	53.040089	129.6978	112 bits/s	351 bits/s	9
192.168.40.128	43064	142.250.4.84	443	31	4 kB	51	16	2 kB	15	2 kB	667.541961	170.6627	116 bits/s	85 bits/s	9
192.168.40.128	48172	142.250.67.78	443	34	11 kB	61	16	7 kB	18	4 kB	2167.803585	0.5401	97 kbps	59 kbps	7
192.168.40.128	48184	142.250.67.78	443	22	4 kB	62	11	2 kB	11	2 kB	2167.804399	0.2320	85 kbps	66 kbps	5
192.168.40.128	57608	142.250.67.78	443	45	12 kB	53	22	7 kB	23	5 kB	967.990454	171.1641	333 bits/s	210 bits/s	9
192.168.40.128	41490	142.250.67.81	443	27	8 kB	19	13	2 kB	14	6 kB	34.695681	139.6561	98 bits/s	371 bits/s	9
192.168.40.128	37708	142.250.76.78	443	30	10 kB	26	15	2 kB	15	8 kB	51.070132	131.6063	110 bits/s	498 bits/s	9
192.168.40.128	54046	142.250.77.234	443	30	13 kB	24	15	2 kB	15	11 kB	49.756167	132.9440	111 bits/s	677 bits/s	7
192.168.40.128	55272	142.250.182.42	443	39	10 kB	54	18	6 kB	21	4 kB	1265.238207	172.6146	295 bits/s	170 bits/s	9
192.168.40.128	51172	142.250.182.46	443	46	11 kB	55	21	7 kB	25	4 kB	1265.288289	172.5577	325 bits/s	204 bits/s	9
192.168.40.128	34510	142.250.193.42	443	27	9 kB	59	13	6 kB	14	3 kB	2167.780774	0.2807	169 kbps	92 kbps	5
192.168.40.128	34520	142.250.193.42	443	22	5 kB	60	11	3 kB	11	2 kB	2167.782235	0.1706	121 kbps	94 kbps	5
192.168.40.128	57230	142.250.193.97	443	25	13 kB	17	12	2 kB	13	11 kB	34.211461	141.5257	94 bits/s	614 bits/s	9
192.168.40.128	40848	142.250.194.42	443	39	10 kB	52	19	6 kB	20	4 kB	964.882277	171.2071	300 bits/s	176 bits/s	9
192.168.40.128	54448	142.251.12.84	443	31	4 kB	56	16	2 kB	15	2 kB	1274.419421	170.5765	116 bits/s	85 bits/s	9
192.168.40.128	51348	142.251.43.174	443	30	10 kB	28	15	2 kB	15	8 kB	53.279147	129.3973	112 bits/s	507 bits/s	9
192.168.40.128	44630	142.251.223.100	443	19	7 kB	5	10	1 kB	9	5 kB	23.043552	3.1247	3,632 bits/s	13 kbps	5
192.168.40.128	44646	142.251.223.100	443	19	7 kB	6	10	1 kB	9	5 kB	23.043898	3.0683	3,699 bits/s	13 kbps	5

2. Protocol Hierarchy



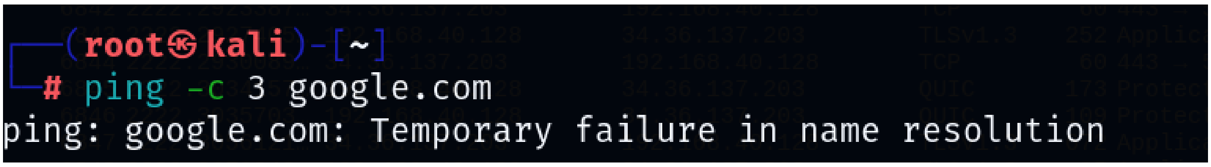
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	6859	100.0	2323756	8,363	0	0	0	6859
Ethernet	100.0	6859	4.4	101214	364	0	0	0	6859
Internet Protocol Version 6	8.1	557	1.0	22896	82	0	0	0	557
User Datagram Protocol	6.6	455	0.2	3640	13	0	0	0	455
Multicast Domain Name System	3.8	264	0.4	8136	29	264	8136	29	264
Link-local Multicast Name Resolution	2.8	191	0.2	4236	15	191	4236	15	191
Internet Control Message Protocol v6	1.5	102	0.1	2528	9	102	2528	9	102
Internet Protocol Version 4	90.5	6208	5.4	124372	447	0	0	0	6208
User Datagram Protocol	57.5	3942	1.4	31536	113	0	0	0	3942
Simple Service Discovery Protocol	0.8	56	0.3	7672	27	56	7672	27	56
QUIC IETF	24.8	1704	56.1	1304653	4,695	1704	1278367	4,601	1774
NetBIOS Name Service	1.9	132	0.3	6600	23	132	6600	23	132
Multicast Domain Name System	3.8	264	0.4	8136	29	264	8136	29	264
Link-local Multicast Name Resolution	2.8	191	0.2	4236	15	191	4236	15	191
Dynamic Host Configuration Protocol	0.1	8	0.1	2346	8	8	2346	8	8
Domain Name System	23.1	1587	3.2	74541	268	1587	74541	268	1587
Transmission Control Protocol	31.1	2132	1.9	44200	159	1342	28400	102	2132
Transport Layer Security	10.8	739	23.2	539898	1,943	739	534250	1,922	743
Hypertext Transfer Protocol	0.7	51	1.0	22857	82	5	1550	5	51
Online Certificate Status Protocol	0.6	42	0.5	11662	41	42	11662	41	42
Line-based text data	0.1	4	0.0	32	0	4	32	0	4
Internet Group Management Protocol	0.8	53	0.0	888	3	53	888	3	53
Internet Control Message Protocol	1.2	81	0.2	5688	20	81	5688	20	81
Address Resolution Protocol	1.4	94	0.1	2632	9	94	2632	9	94

No display filter.

TROUBLESHOOTING NETWORK ISSUES

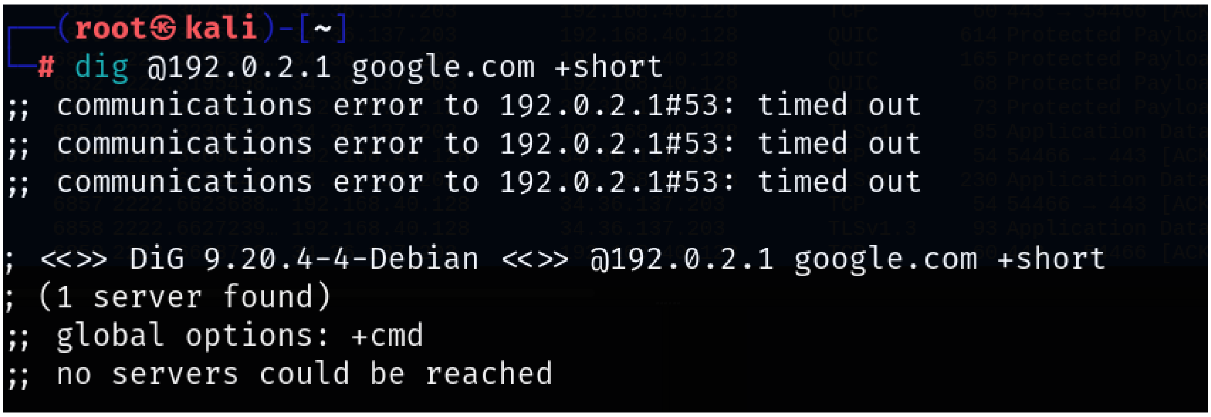
Symptoms on Client

1. Ping Fails with **Unknown Hosts**



```
┌──(root㉿kali)-[~]
└─# ping -c 3 google.com
ping: google.com: Temporary failure in name resolution
```

2. **Timeout** response

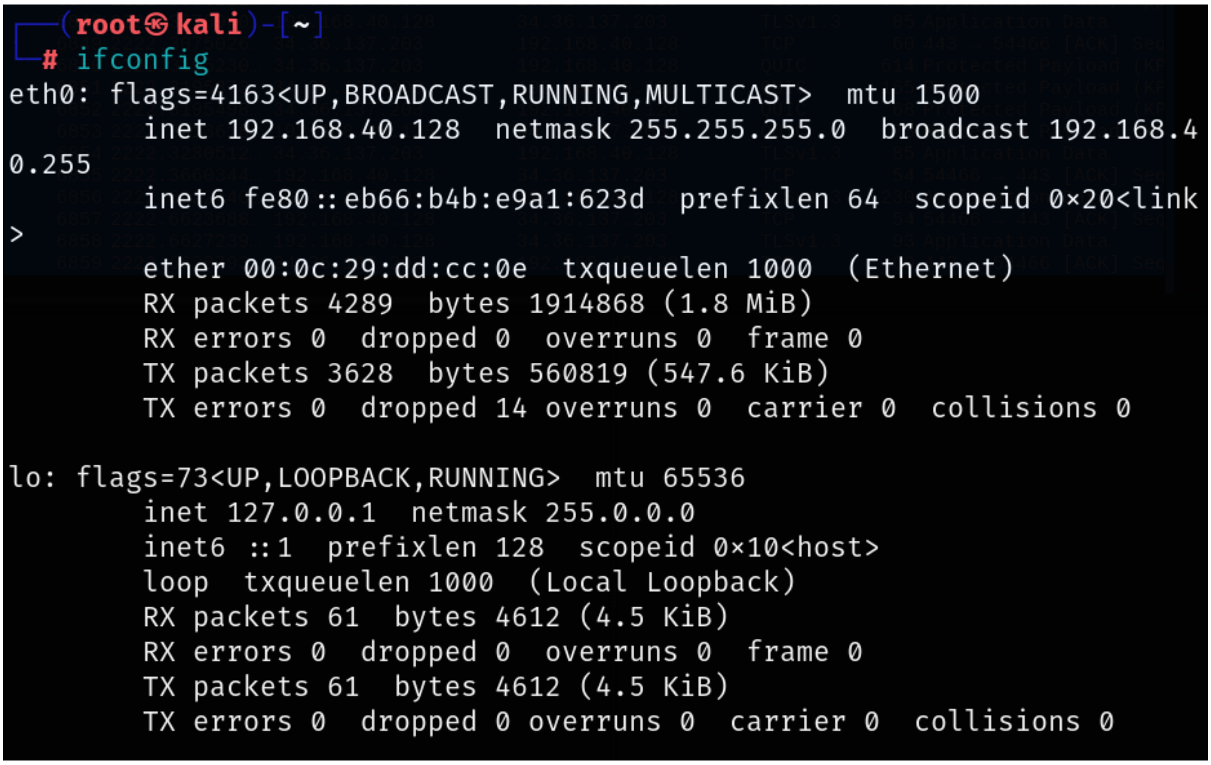


```
┌──(root㉿kali)-[~]
└─# dig @192.0.2.1 google.com +short
;; communications error to 192.0.2.1#53: timed out
;; communications error to 192.0.2.1#53: timed out
;; communications error to 192.0.2.1#53: timed out

; <<>> DiG 9.20.4-4-Debian <<>> @192.0.2.1 google.com +short
; (1 server found)
;; global options: +cmd
;; no servers could be reached
```

Capturing DNS Traffic on Client interface

1. Finding Interface

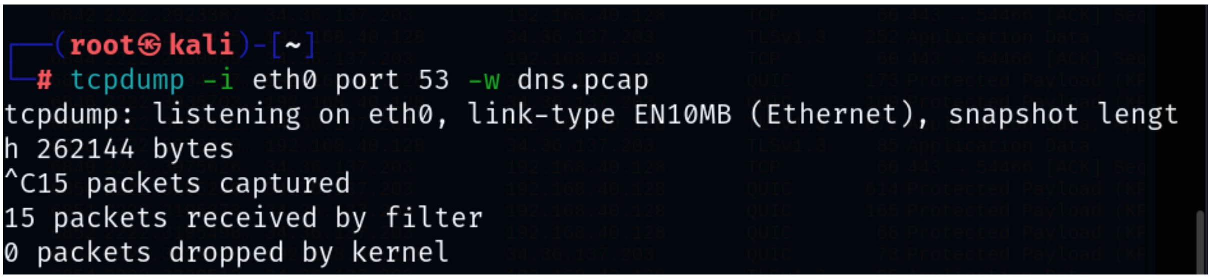
Command Used: “**ifconfig**”



```
┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.40.128  netmask 255.255.255.0  broadcast 192.168.4
0.255
        inet6 fe80::eb66:b4b:e9a1:623d  prefixlen 64  scopeid 0x20<link
>
        ether 00:0c:29:dd:cc:0e  txqueuelen 1000  (Ethernet)
        RX packets 4289  bytes 1914868 (1.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3628  bytes 560819 (547.6 KiB)
        TX errors 0  dropped 14 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 61  bytes 4612 (4.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 61  bytes 4612 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

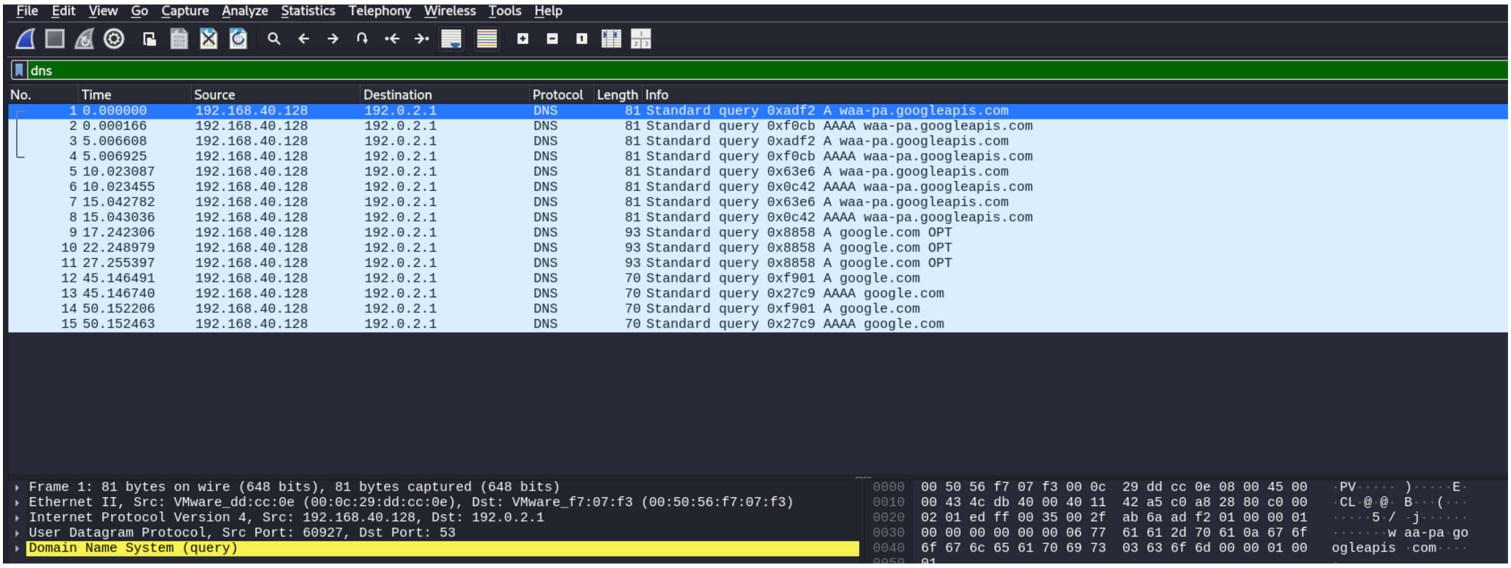
2. Capturing Traffic

Command Used: "**tcpdump -i eth0 port 53 -w dns.pcap**”



```
┌──(root㉿kali)-[~]
└─# tcpdump -i eth0 port 53 -w dns.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot lengt
h 262144 bytes
^C15 packets captured
15 packets received by filter
0 packets dropped by kernel
```

3. Analysing **pcap** file in Wireshark



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

```
No.  Time         Source           Destination   Protocol  Length Info
   1 0.000000     192.168.40.128   192.0.2.1     DNS       81 Standard query 0xadf2 A waa-pa.googleapis.com
   2 0.000166     192.168.40.128   192.0.2.1     DNS       81 Standard query 0xf0cb AAAA waa-pa.googleapis.com
   3 5.006608     192.168.40.128   192.0.2.1     DNS       81 Standard query 0xadf2 A waa-pa.googleapis.com
   4 5.006925     192.168.40.128   192.0.2.1     DNS       81 Standard query 0xf0cb AAAA waa-pa.googleapis.com
   5 10.023087    192.168.40.128   192.0.2.1     DNS       81 Standard query 0x63e6 A waa-pa.googleapis.com
   6 10.023455    192.168.40.128   192.0.2.1     DNS       81 Standard query 0x0c42 AAAA waa-pa.googleapis.com
   7 15.042782    192.168.40.128   192.0.2.1     DNS       81 Standard query 0x63e6 A waa-pa.googleapis.com
   8 15.043036    192.168.40.128   192.0.2.1     DNS       81 Standard query 0x0c42 AAAA waa-pa.googleapis.com
   9 17.242306    192.168.40.128   192.0.2.1     DNS       93 Standard query 0x8858 A google.com OPT
  10 22.248979    192.168.40.128   192.0.2.1     DNS       93 Standard query 0x8858 A google.com OPT
  11 27.255397    192.168.40.128   192.0.2.1     DNS       93 Standard query 0x8858 A google.com OPT
  12 45.146491    192.168.40.128   192.0.2.1     DNS       70 Standard query 0xf901 A google.com
  13 45.146740    192.168.40.128   192.0.2.1     DNS       70 Standard query 0x27c9 AAAA google.com
  14 50.152206    192.168.40.128   192.0.2.1     DNS       70 Standard query 0xf901 A google.com
  15 50.152463    192.168.40.128   192.0.2.1     DNS       70 Standard query 0x27c9 AAAA google.com
```

```
› Frame 1: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)
› Ethernet II, Src: VMware_dd:cc:0e (00:0c:29:dd:cc:0e), Dst: VMware_f7:07:f3 (00:50:56:f7:07:f3)
› Internet Protocol Version 4, Src: 192.168.40.128, Dst: 192.0.2.1
› User Datagram Protocol, Src Port: 60927, Dst Port: 53
› Domain Name System (query)
```

```
0000  00 50 56 f7 07 f3 00 0c  29 dd cc 0e 08 00 45 00   ·PV·····)·····E·
0010  00 43 4c db 40 00 40 11  42 a5 c0 a8 28 80 c0 00   ·CL·@·@·B···(···
0020  02 01 ed ff 00 35 00 2f  ab 6a ad f2 01 00 00 01   ·····5·/·j······
0030  00 00 00 00 00 00 06 77  61 61 2d 70 61 0a 67 6f   ·······w aa-pa·go
0040  6f 67 6c 65 61 70 69 73  03 63 6f 6d 00 00 01 00   ogleapis·com····
0050  01
```

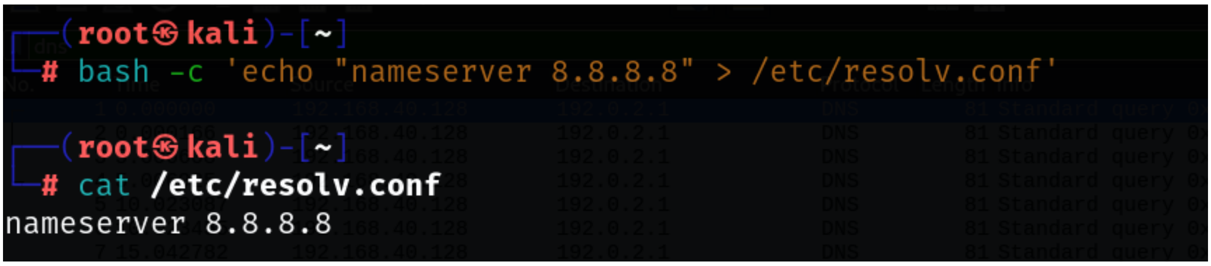
Diagnosis

1. If DNS queries go to a non-responsive server or to the wrong IP → DNS misconfiguration.

2. If queries go out and replies exist but client still fails → check client firewall or caching.

Resolving Problem

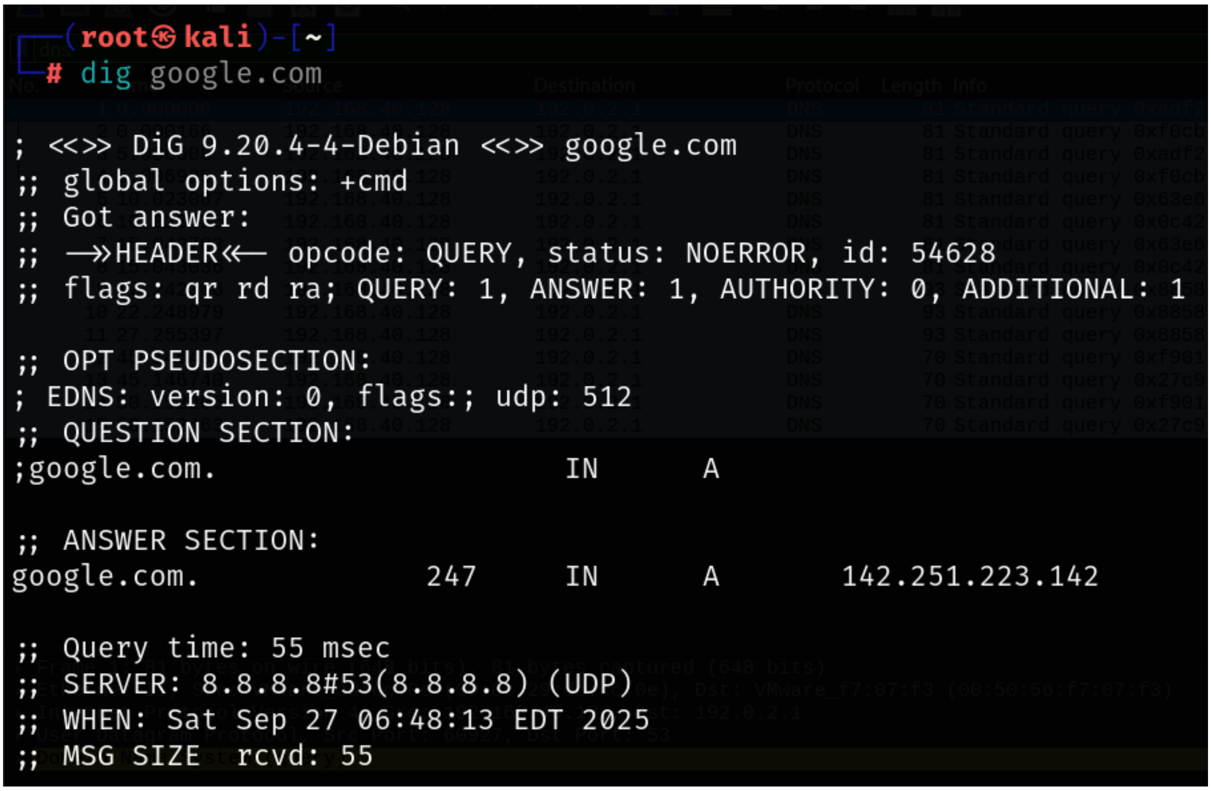
1. Restoring **/etc/resolv.conf** to a valid resolver



```
┌──(root㉿kali)-[~]
└─# bash -c 'echo "nameserver 8.8.8.8" > /etc/resolv.conf'

┌──(root㉿kali)-[~]
└─# cat /etc/resolv.conf
nameserver 8.8.8.8
```

2. Re-running the commands

Command 1: “**dig google.com**”



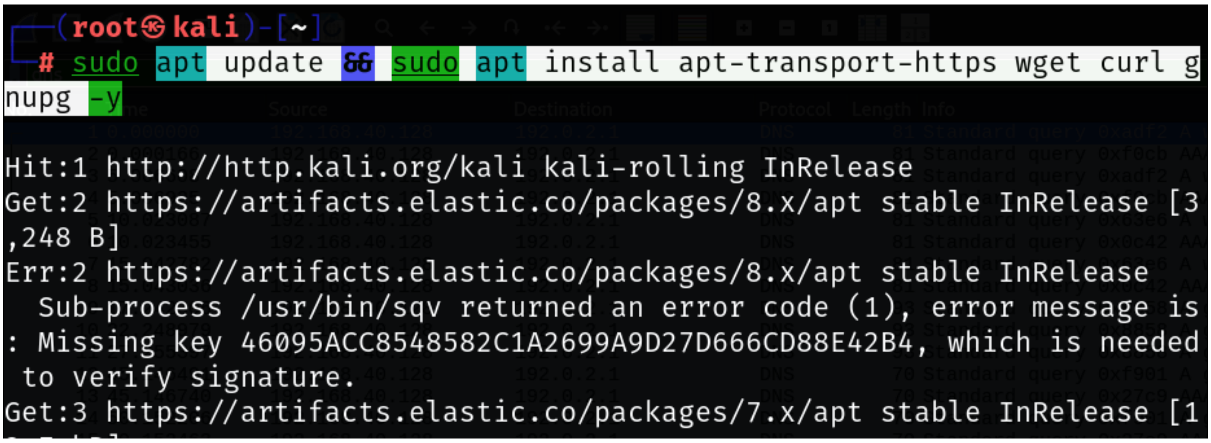
```
┌──(root㉿kali)-[~]
└─# dig google.com

; <<>> DiG 9.20.4-4-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54628
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             247     IN      A       142.251.223.142

;; Query time: 55 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Sat Sep 27 06:48:13 EDT 2025
;; MSG SIZE  rcvd: 55
```

SOC and Networking Integration

Installing Elasticsearch

1. Installing Dependencies

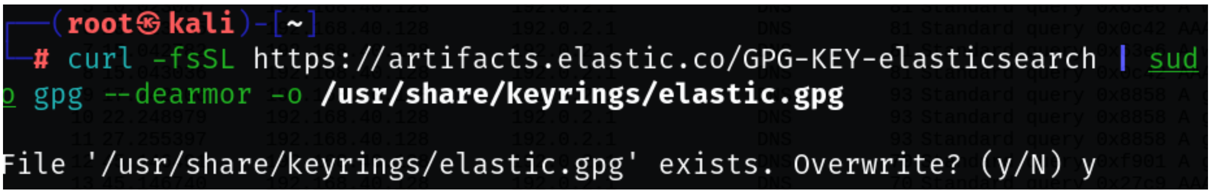
Command Used: “**apt update && sudo apt install apt-transport-https wget curl gnupg -y**”



```
┌──(root㉿kali)-[~]
└─# sudo apt update && sudo apt install apt-transport-https wget curl g
nupg -y

Hit:1 http://http.kali.org/kali kali-rolling InRelease
Get:2 https://artifacts.elastic.co/packages/8.x/apt stable InRelease [3
,248 B]
Err:2 https://artifacts.elastic.co/packages/8.x/apt stable InRelease
  Sub-process /usr/bin/sqv returned an error code (1), error message is
: Missing key 46095ACC8548582C1A2699A9D27D666CD88E42B4, which is needed
 to verify signature.
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [1
```

2. Making a Key Location to save key of elasticsearch

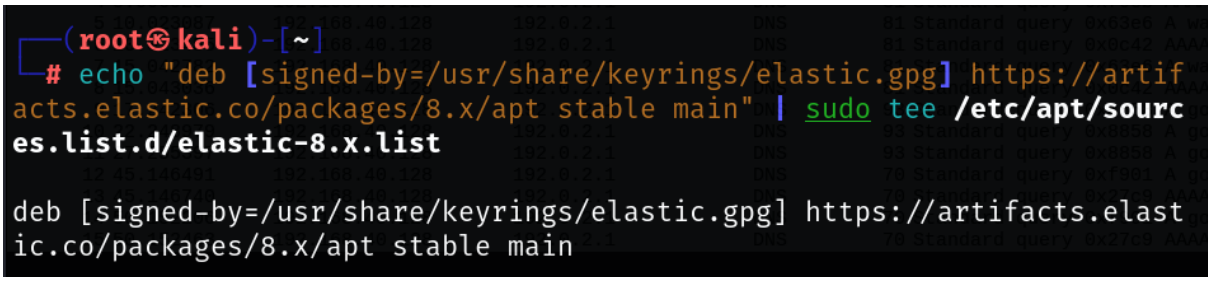
Command Used: “**curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elastic.gpg**”



```
┌──(root㉿kali)-[~]
└─# curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sud
o gpg --dearmor -o /usr/share/keyrings/elastic.gpg

File '/usr/share/keyrings/elastic.gpg' exists. Overwrite? (y/N) y
```

3. Adding Elasticsearch APT repository

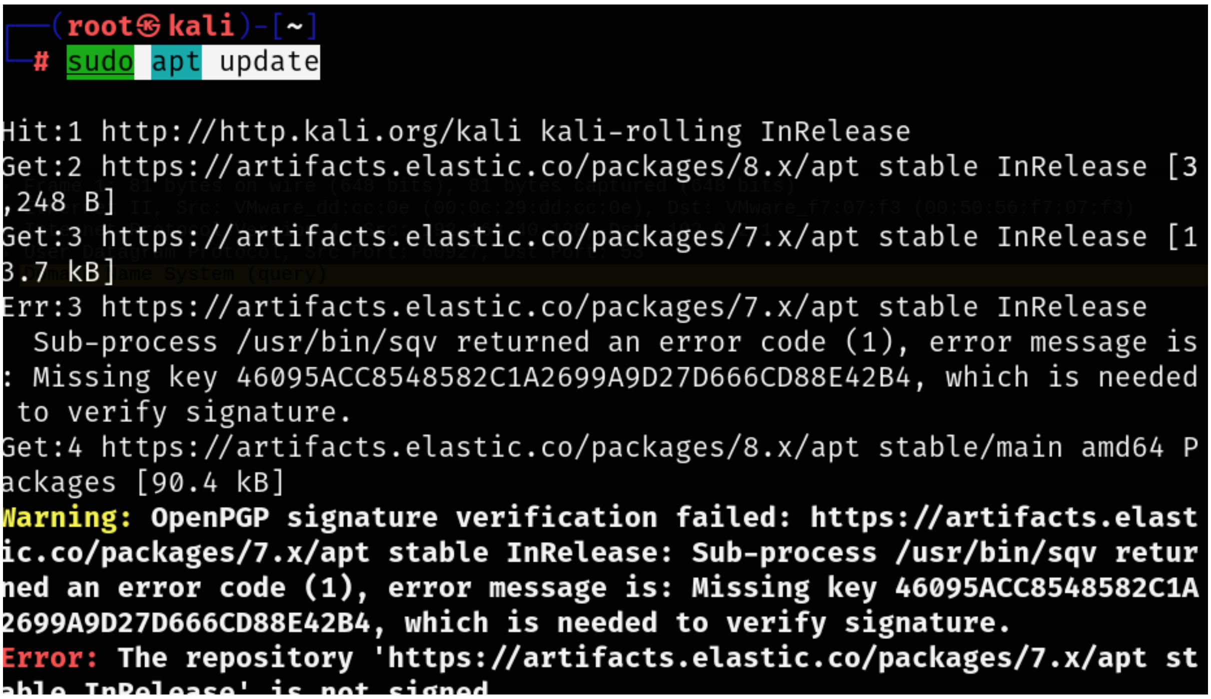
Command Used: “**echo "deb [signed-by=/usr/share/keyrings/elastic.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list**”



```
┌──(root㉿kali)-[~]
└─# echo "deb [signed-by=/usr/share/keyrings/elastic.gpg] https://artif
acts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sourc
es.list.d/elastic-8.x.list

deb [signed-by=/usr/share/keyrings/elastic.gpg] https://artifacts.elast
ic.co/packages/8.x/apt stable main
```

4. Updating Kali Packages

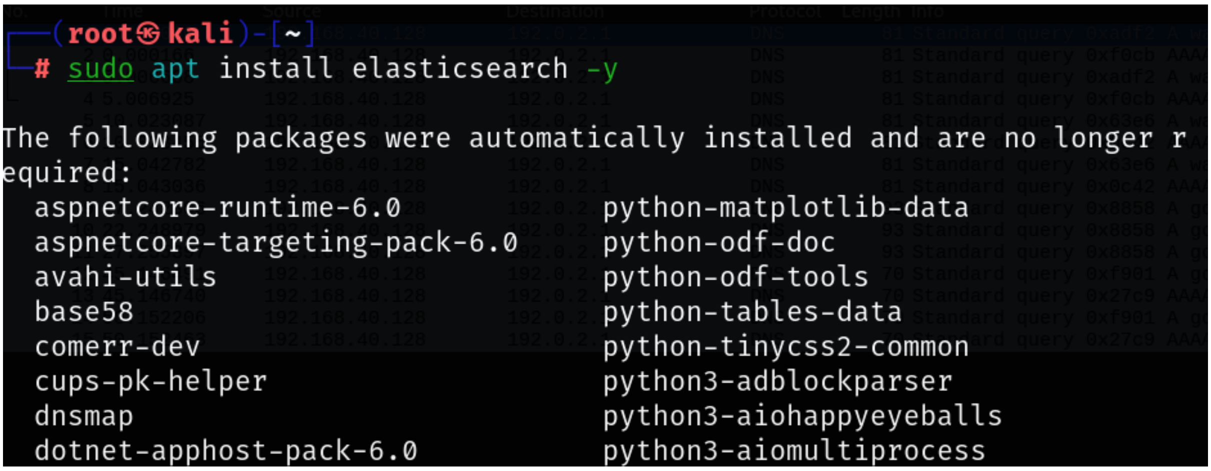
Command Used: “**sudo apt update**”



```
┌──(root㉿kali)-[~]
└─# sudo apt update

Hit:1 http://http.kali.org/kali kali-rolling InRelease
Get:2 https://artifacts.elastic.co/packages/8.x/apt stable InRelease [3
,248 B]
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [1
3.7 kB]
Err:3 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
  Sub-process /usr/bin/sqv returned an error code (1), error message is
: Missing key 46095ACC8548582C1A2699A9D27D666CD88E42B4, which is needed
 to verify signature.
Get:4 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 P
ackages [90.4 kB]
Warning: OpenPGP signature verification failed: https://artifacts.elast
ic.co/packages/7.x/apt stable InRelease: Sub-process /usr/bin/sqv retur
ned an error code (1), error message is: Missing key 46095ACC8548582C1A
2699A9D27D666CD88E42B4, which is needed to verify signature.
Error: The repository 'https://artifacts.elastic.co/packages/7.x/apt st
able InRelease' is not signed.
```

5. Installing elasticsearch

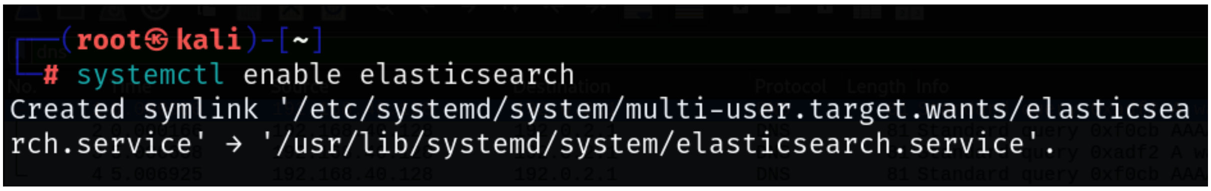
Command Used : “**sudo apt install elasticsearch -y**”



```
┌──(root㉿kali)-[~]
└─# sudo apt install elasticsearch -y

The following packages were automatically installed and are no longer r
equired:
  aspnetcore-runtime-6.0           python-matplotlib-data
  aspnetcore-targeting-pack-6.0    python-odf-doc
  avahi-utils                      python-odf-tools
  base58                           python-tables-data
  comerr-dev                       python-tinycss2-common
  cups-pk-helper                   python3-adblockparser
  dnsmap                           python3-aiohappyeyeballs
  dotnet-apphost-pack-6.0          python3-aiomultiprocess
```

6. Enabling services and starting elasticsearch

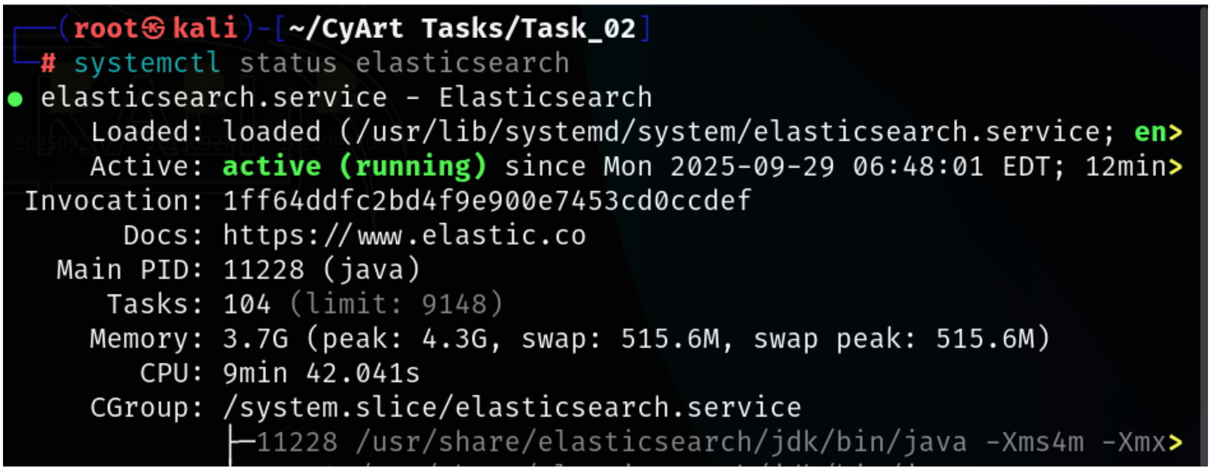
Command 1: “**systemctl enable elasticsearch**”
Command 2: “**systemctl start elasticsearch**”



```
┌──(root㉿kali)-[~]
└─# systemctl enable elasticsearch
Created symlink '/etc/systemd/system/multi-user.target.wants/elasticsea
rch.service' → '/usr/lib/systemd/system/elasticsearch.service'.
```

7. Checking Status of elastic search

Command Used: “**systemctl status elasticsearch**”

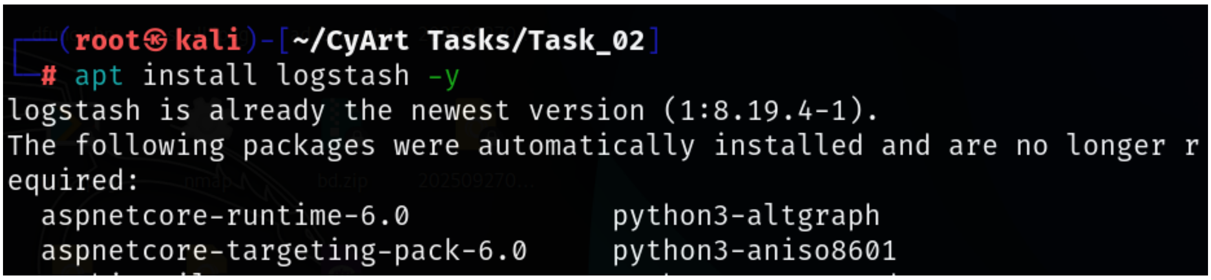


```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; en>
     Active: active (running) since Mon 2025-09-29 06:48:01 EDT; 12min>
 Invocation: 1ff64ddfc2bd4f9e900e7453cd0ccdef
       Docs: https://www.elastic.co
   Main PID: 11228 (java)
      Tasks: 104 (limit: 9148)
     Memory: 3.7G (peak: 4.3G, swap: 515.6M, swap peak: 515.6M)
        CPU: 9min 42.041s
     CGroup: /system.slice/elasticsearch.service
             ├─11228 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx>
```

Installing logstash

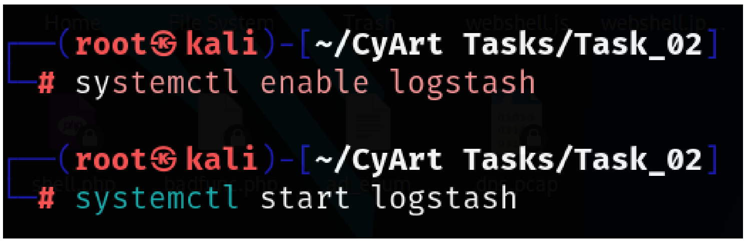
1. Installing logstash

Command Used: “**apt install logstash -y**”



```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# apt install logstash -y
logstash is already the newest version (1:8.19.4-1).
The following packages were automatically installed and are no longer r
equired:
  aspnetcore-runtime-6.0            python3-altgraph
  aspnetcore-targeting-pack-6.0     python3-aniso8601
```

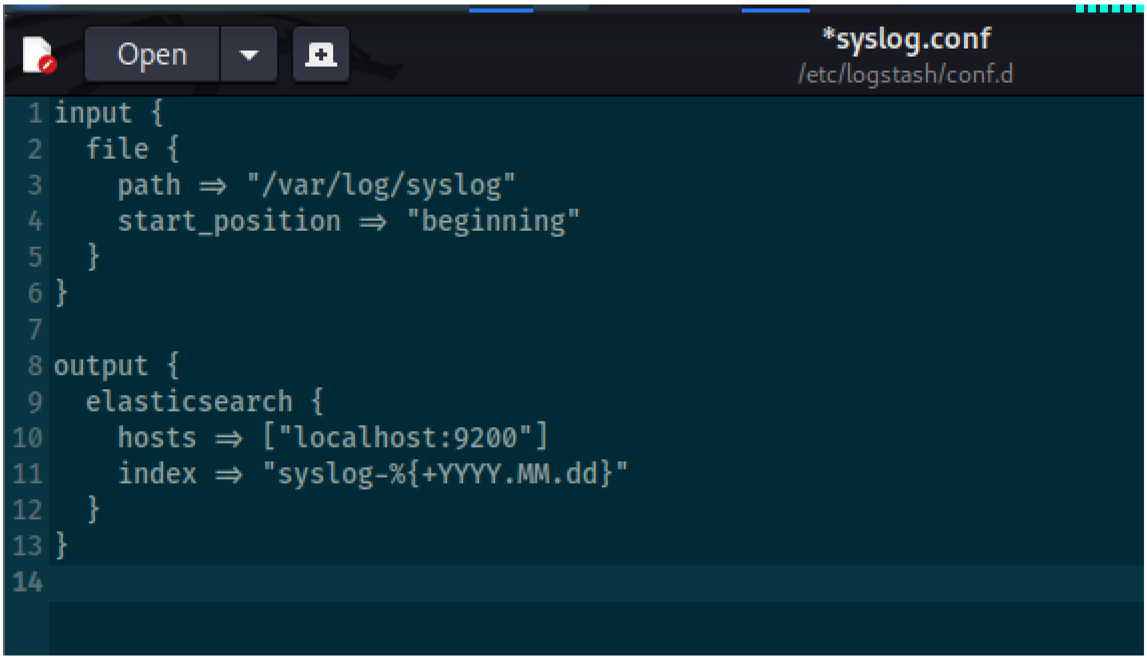
2. Enabling and starting logstash services
Command 1: “**systemctl enable logstash**”
Command 2: “**systemctl start logstash**”



```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# systemctl enable logstash

┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# systemctl start logstash
```

3. Configuring logstash to collect logs

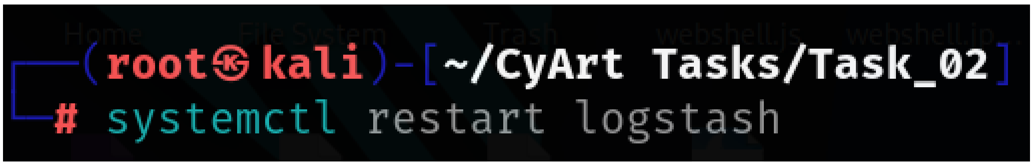
Command Used: “**nano /etc/logstash/conf.d/syslog.conf**”



```
input {
  file {
    path => "/var/log/syslog"
    start_position => "beginning"
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "syslog-%{+YYYY.MM.dd}"
  }
}
```

4. Restarting service

Command Used: “**systemctl restart logstash**”

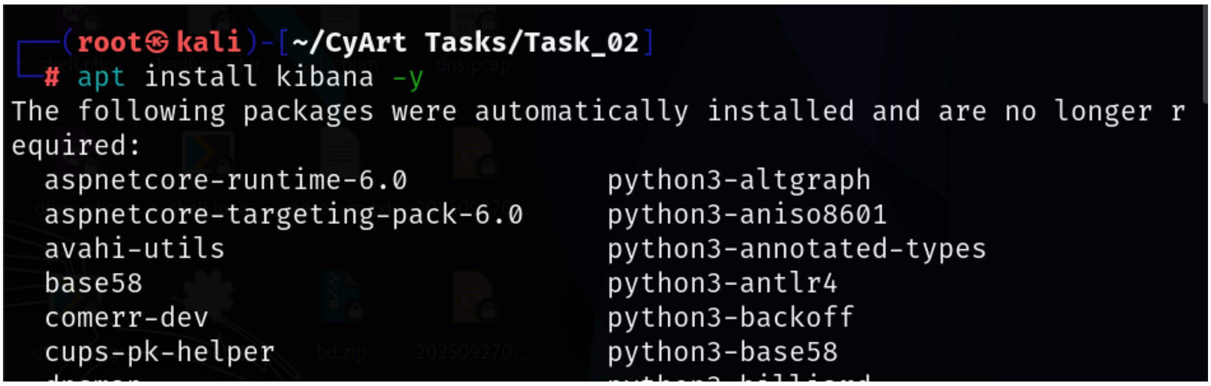


```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# systemctl restart logstash
```

Installing Kibana

1. Installing Kibana

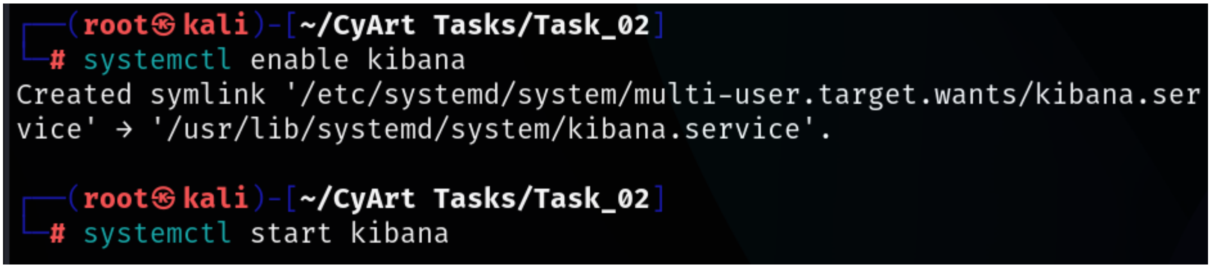
Command Used: “**apt install kibana -y**”



```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# apt install kibana -y
The following packages were automatically installed and are no longer r
equired:
  aspnetcore-runtime-6.0                python3-altgraph
  aspnetcore-targeting-pack-6.0         python3-aniso8601
  avahi-utils                           python3-annotated-types
  base58                                python3-antlr4
  comerr-dev                            python3-backoff
  cups-pk-helper                        python3-base58
```

2. Enabling and starting Kibana services

Command 1: “systemctl enable kibana”
Command 2: “systemctl start kibana”

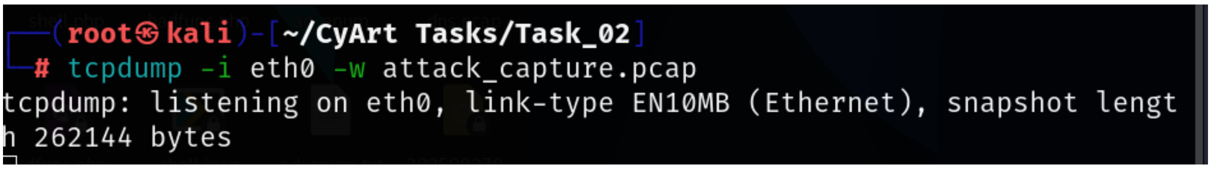


```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# systemctl enable kibana
Created symlink '/etc/systemd/system/multi-user.target.wants/kibana.ser
vice' → '/usr/lib/systemd/system/kibana.service'.

┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# systemctl start kibana
```

SIMULATING AN INCIDENT

1. Capturing traffic from tcpdump

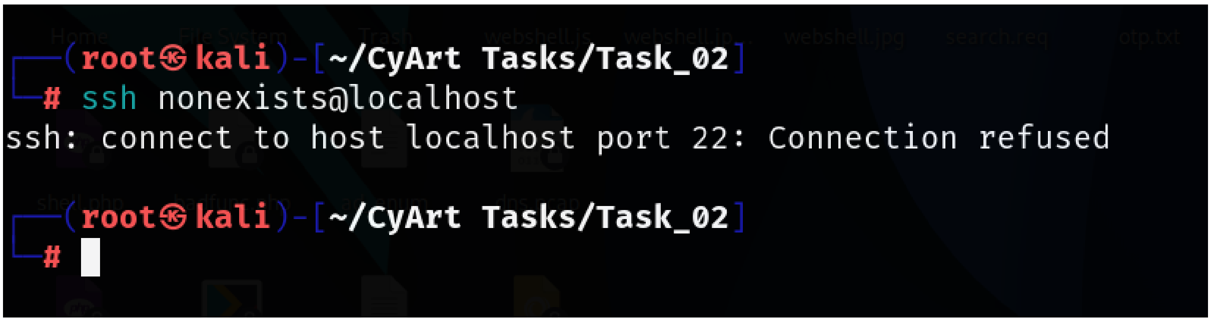
Command Used: “**tcpdump -i eth0 -w attack_capture.pcap**”



```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# tcpdump -i eth0 -w attack_capture.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot lengt
h 262144 bytes
```

2. Unauthorized SSH login attempt

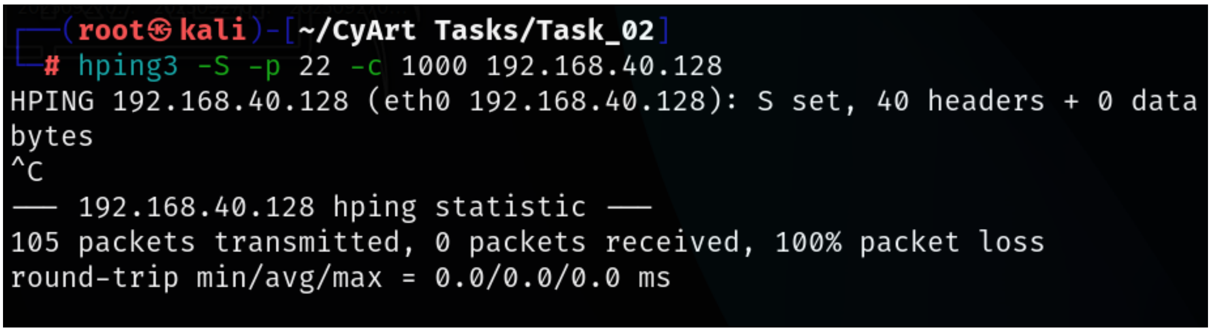
Command Used: “**ssh nonexists@localhost**”



```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# ssh nonexists@localhost
ssh: connect to host localhost port 22: Connection refused

┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# 
```

3. DoS Attack by hping3

Command Used: “**hping3 -S -p 22 -c 1000 192.168.40.128**”

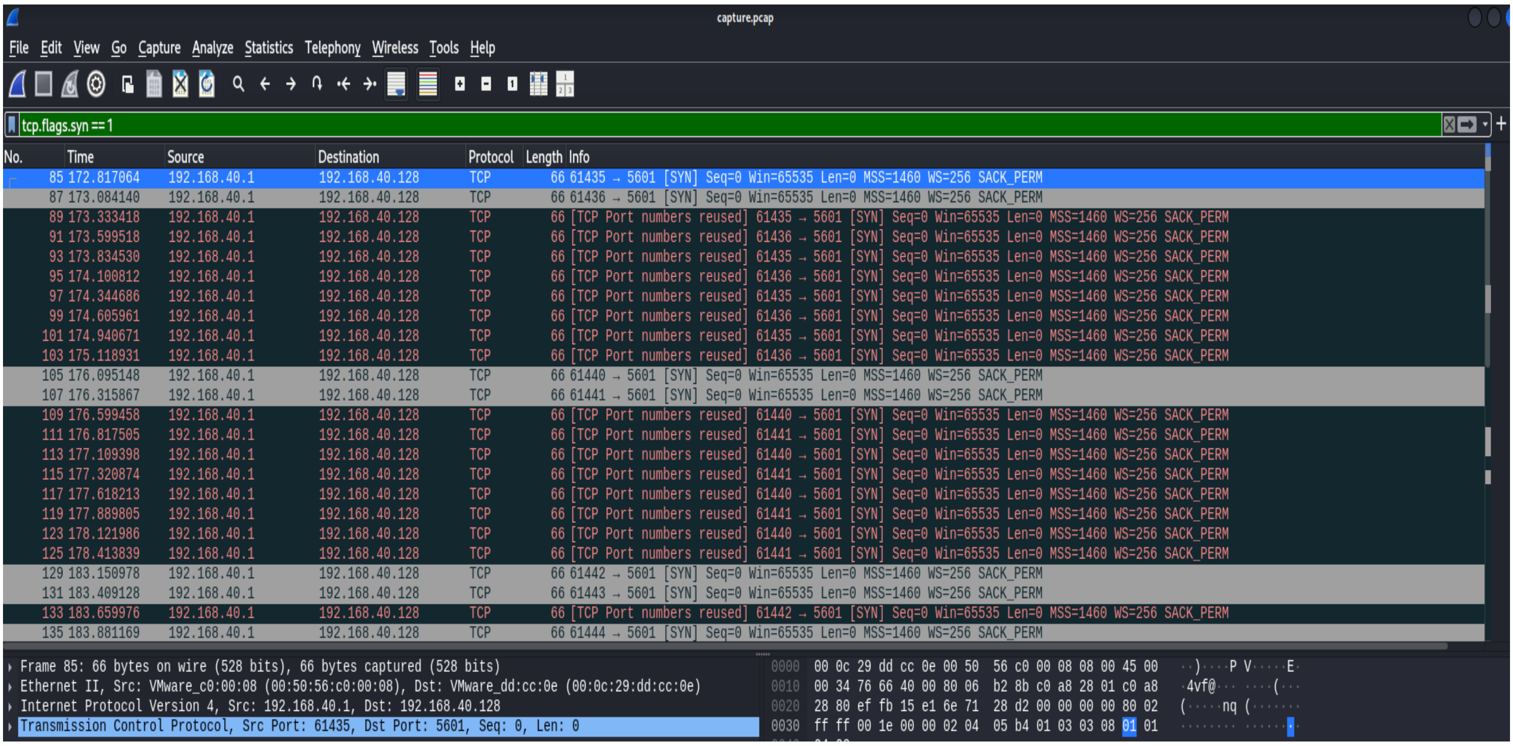


```
┌──(root㉿kali)-[~/CyArt Tasks/Task_02]
└─# hping3 -S -p 22 -c 1000 192.168.40.128
HPING 192.168.40.128 (eth0 192.168.40.128): S set, 40 headers + 0 data
bytes
^C
--- 192.168.40.128 hping statistic ---
105 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ANALYSING

1. DoS Patterns in Wireshark

Filter Used: “tcp.flags.syn == 1”


capture.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.flags.syn == 1
```

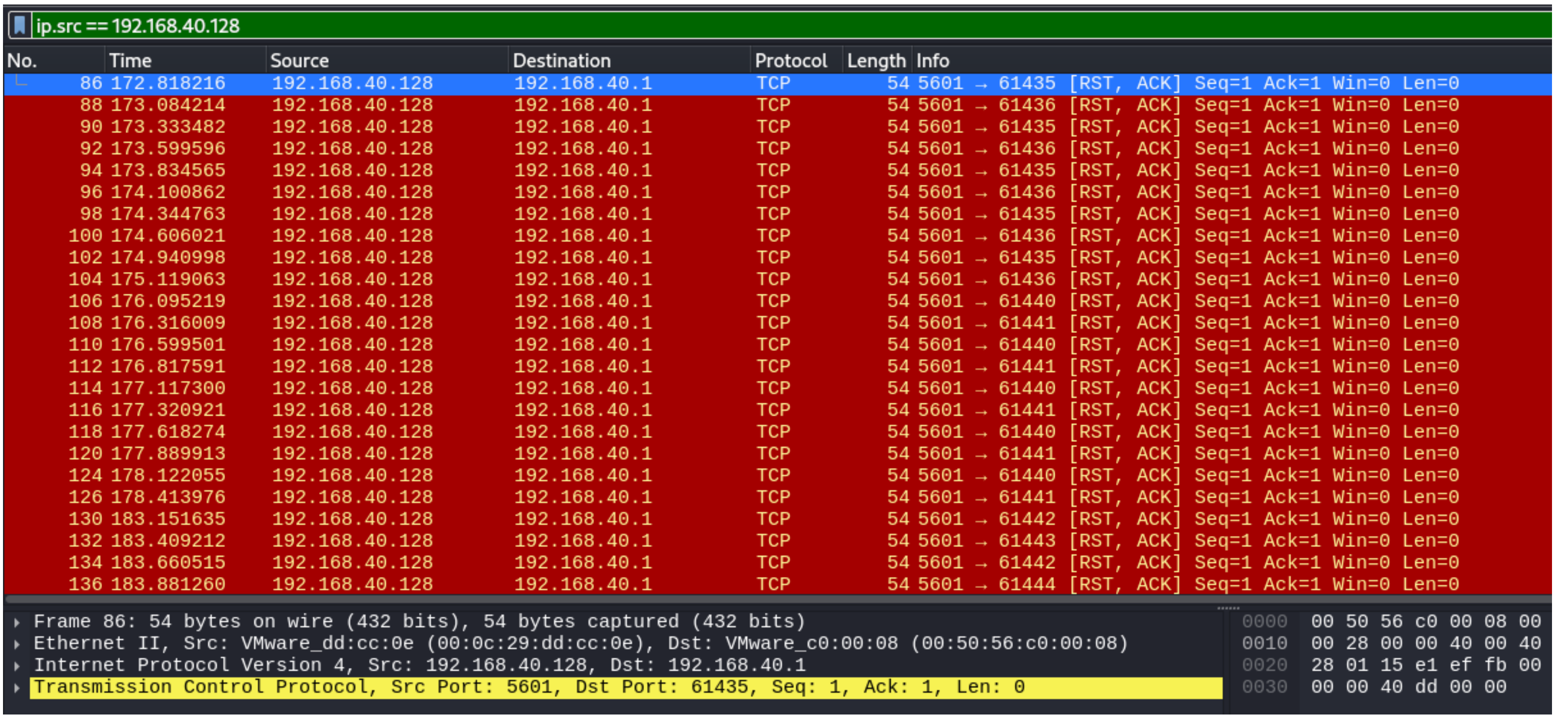
```
No.   Time         Source         Destination      Protocol Length Info
   85 172.817064   192.168.40.1   192.168.40.128   TCP      66 61435 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
   87 173.084140   192.168.40.1   192.168.40.128   TCP      66 61436 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
   89 173.333418   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61435 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
   91 173.599518   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61436 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
   93 173.834530   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61435 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
   95 174.100812   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61436 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
   97 174.344686   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61435 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
   99 174.605961   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61436 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  101 174.940671   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61435 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  103 175.118931   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61436 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  105 176.095148   192.168.40.1   192.168.40.128   TCP      66 61440 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  107 176.315867   192.168.40.1   192.168.40.128   TCP      66 61441 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  109 176.599458   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61440 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  111 176.817505   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61441 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  113 177.109398   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61440 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  115 177.320874   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61441 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  117 177.618213   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61440 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  119 177.889805   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61441 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  123 178.121986   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61440 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  125 178.413839   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61441 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  129 183.150978   192.168.40.1   192.168.40.128   TCP      66 61442 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  131 183.409128   192.168.40.1   192.168.40.128   TCP      66 61443 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  133 183.659976   192.168.40.1   192.168.40.128   TCP      66 [TCP Port numbers reused] 61442 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  135 183.881169   192.168.40.1   192.168.40.128   TCP      66 61444 → 5601 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
```

```
› Frame 85: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
› Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_dd:cc:0e (00:0c:29:dd:cc:0e)
› Internet Protocol Version 4, Src: 192.168.40.1, Dst: 192.168.40.128
› Transmission Control Protocol, Src Port: 61435, Dst Port: 5601, Seq: 0, Len: 0
```

```
0000  00 0c 29 dd cc 0e 00 50  56 c0 00 08 08 00 45 00   ··)····P V·····E·
0010  00 34 76 66 40 00 80 06  b2 8b c0 a8 28 01 c0 a8   ·4vf@··· ····(···
0020  28 80 ef fb 15 e1 6e 71  28 d2 00 00 00 00 80 02   (·····nq (·······
0030  ff ff 00 1e 00 00 02 04  05 b4 01 03 03 08 01 01   ········ ········
```

2. IP Filtration

Filter Used: “**ip.src == 192.168.40.128**”



```
ip.src == 192.168.40.128
No.  Time        Source          Destination   Protocol Length Info
  86 172.818216  192.168.40.128  192.168.40.1  TCP      54 5601 → 61435 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  88 173.084214  192.168.40.128  192.168.40.1  TCP      54 5601 → 61436 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  90 173.333482  192.168.40.128  192.168.40.1  TCP      54 5601 → 61435 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  92 173.599596  192.168.40.128  192.168.40.1  TCP      54 5601 → 61436 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  94 173.834565  192.168.40.128  192.168.40.1  TCP      54 5601 → 61435 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  96 174.100862  192.168.40.128  192.168.40.1  TCP      54 5601 → 61436 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  98 174.344763  192.168.40.128  192.168.40.1  TCP      54 5601 → 61435 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 100 174.606021  192.168.40.128  192.168.40.1  TCP      54 5601 → 61436 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 102 174.940998  192.168.40.128  192.168.40.1  TCP      54 5601 → 61435 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 104 175.119063  192.168.40.128  192.168.40.1  TCP      54 5601 → 61436 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 106 176.095219  192.168.40.128  192.168.40.1  TCP      54 5601 → 61440 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 108 176.316009  192.168.40.128  192.168.40.1  TCP      54 5601 → 61441 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 110 176.599501  192.168.40.128  192.168.40.1  TCP      54 5601 → 61440 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 112 176.817591  192.168.40.128  192.168.40.1  TCP      54 5601 → 61441 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 114 177.117300  192.168.40.128  192.168.40.1  TCP      54 5601 → 61440 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 116 177.320921  192.168.40.128  192.168.40.1  TCP      54 5601 → 61441 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 118 177.618274  192.168.40.128  192.168.40.1  TCP      54 5601 → 61440 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 120 177.889913  192.168.40.128  192.168.40.1  TCP      54 5601 → 61441 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 124 178.122055  192.168.40.128  192.168.40.1  TCP      54 5601 → 61440 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 126 178.413976  192.168.40.128  192.168.40.1  TCP      54 5601 → 61441 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 130 183.151635  192.168.40.128  192.168.40.1  TCP      54 5601 → 61442 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 132 183.409212  192.168.40.128  192.168.40.1  TCP      54 5601 → 61443 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 134 183.660515  192.168.40.128  192.168.40.1  TCP      54 5601 → 61442 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 136 183.881260  192.168.40.128  192.168.40.1  TCP      54 5601 → 61444 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

▸ Frame 86: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)                       0000  00 50 56 c0 00 08 00
▸ Ethernet II, Src: VMware_dd:cc:0e (00:0c:29:dd:cc:0e), Dst: VMware_c0:00:08 (00:50:56:c0:00:08) 0010  00 28 00 00 40 00 40
▸ Internet Protocol Version 4, Src: 192.168.40.128, Dst: 192.168.40.1                       0020  28 01 15 e1 ef fb 00
▸ Transmission Control Protocol, Src Port: 5601, Dst Port: 61435, Seq: 1, Ack: 1, Len: 0    0030  00 00 40 dd 00 00
```