

## Task 5: Network Traffic Capture & Analysis with Wireshark

### Objective

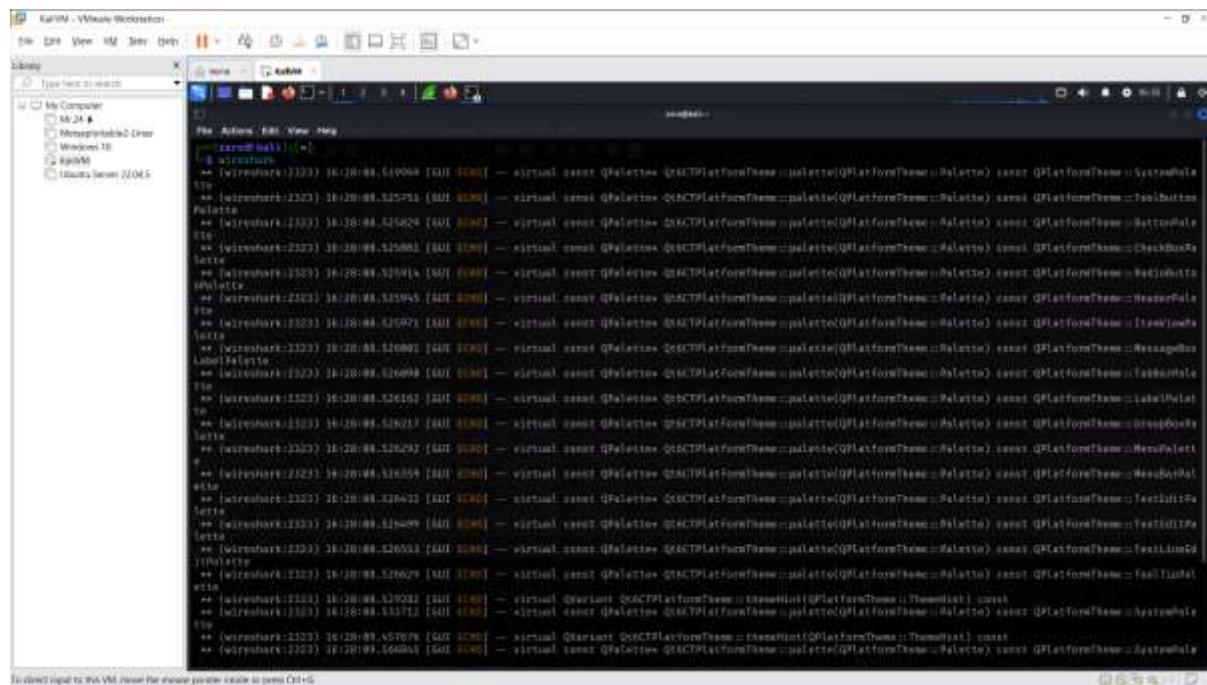
Capture live network traffic, apply filters to isolate specific protocols or hosts, and analyze packets to understand communication patterns and detect anomalies.

### Environment

- **Operating System:** Kali Linux
- **Tool Used:** Wireshark

### Steps Performed

#### 1. Opened Wireshark



#### 2. sudo wireshark

Selected the active interface (eth0 for wired / wlan0 for wireless).

### Choose the Right Network Interface

- In Wireshark's home screen, select your **active network interface**:

- **eth0** → wired
- **wlan0** → Wi-Fi
- You can confirm your active interface with:

`ip a`

```

KaliVM - VMware Workstation
File Edit View VM Help
File Actions Edit View Help
[certbot] (root)
$ ip a
1: lo:   mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00 state UNKNOWN group default qlen 1000
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
        valid_lft forever preferred_lft forever
2: eth0:   mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:1b:8e:0b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10 brd 192.168.1.255 scope global dynamic preferedlifetm
        valid_lft 1000sec preferred_lft 1000sec
        inet6 fe80::20c:29ff:fe1b:8e0b/64 scope link noPreferred
            valid_lft forever preferred_lft forever
3: wlan0:   mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:1b:8e:0b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11 brd 192.168.1.255 scope global dynamic preferedlifetm
        valid_lft 1000sec preferred_lft 1000sec
        inet6 fe80::20c:29ff:fe1b:8e0b/64 scope link noPreferred
            valid_lft forever preferred_lft forever
4: docker0:   mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether 02:42:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
[certbot] (root)
$ 

```

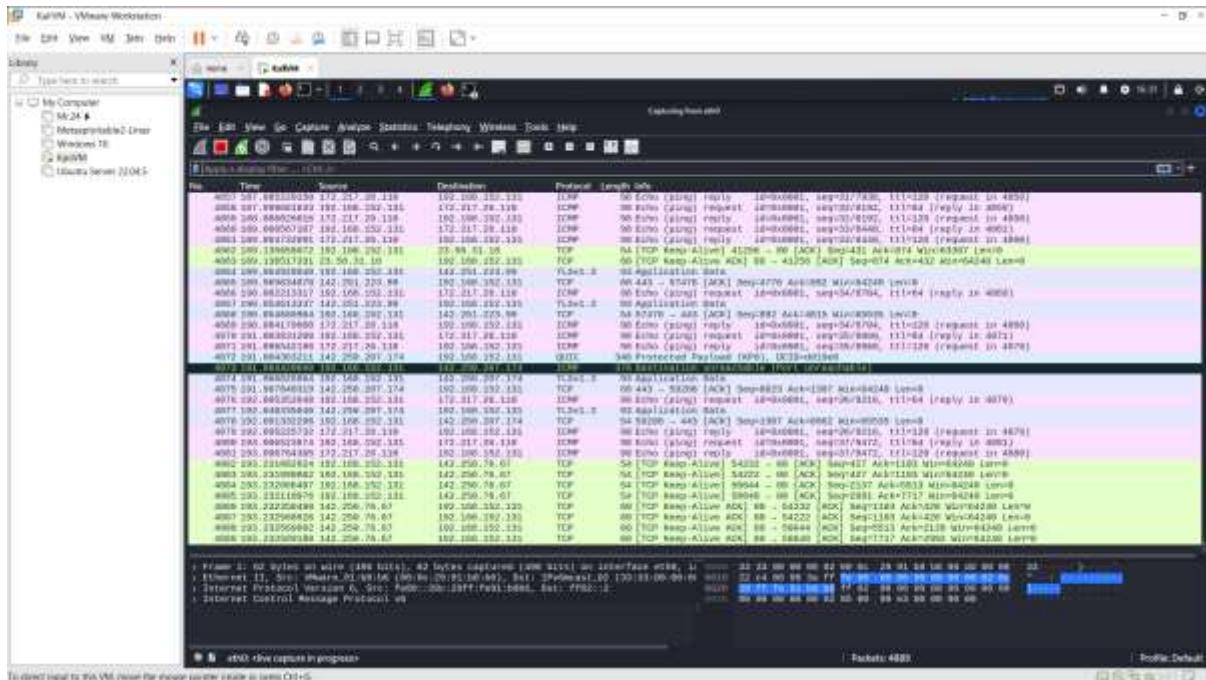
### 3. Started Packet Capture

Began capturing live network packets.

### 4. Generated Network Traffic

- Visited multiple websites
- Ran:

`ping google.com`



## 5. Stopped and Saved Capture

- Saved the file as task5.pcapng for analysis.

## 6. Applied Filters for Analysis

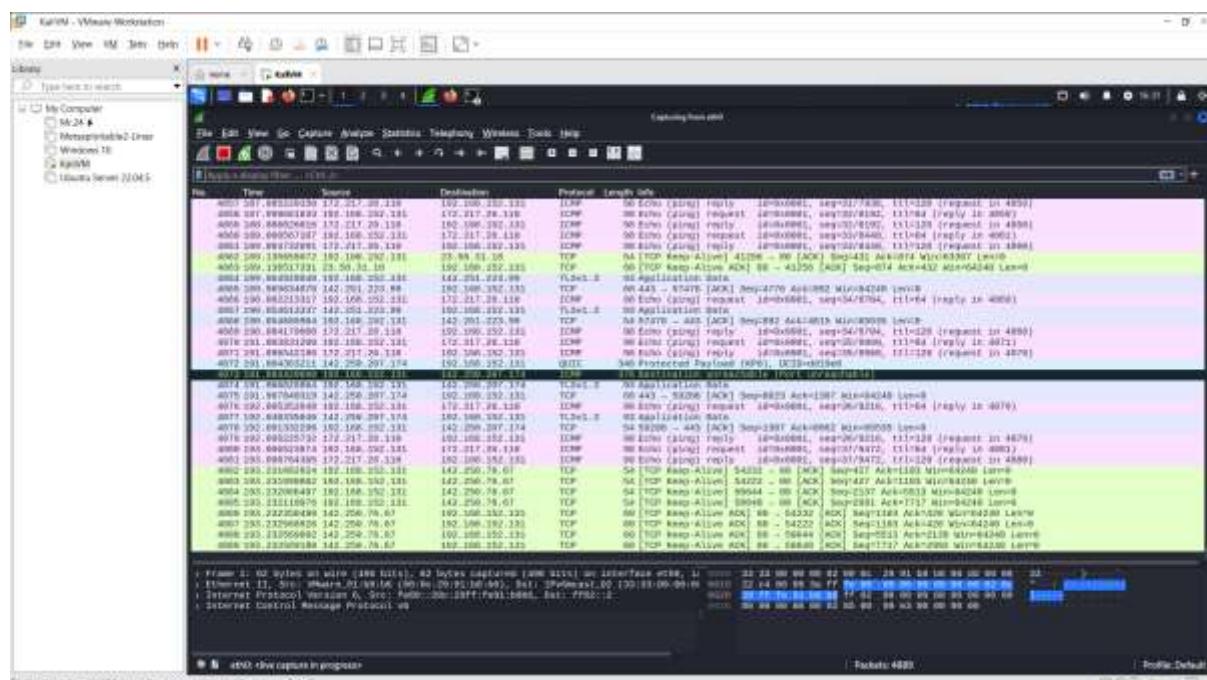
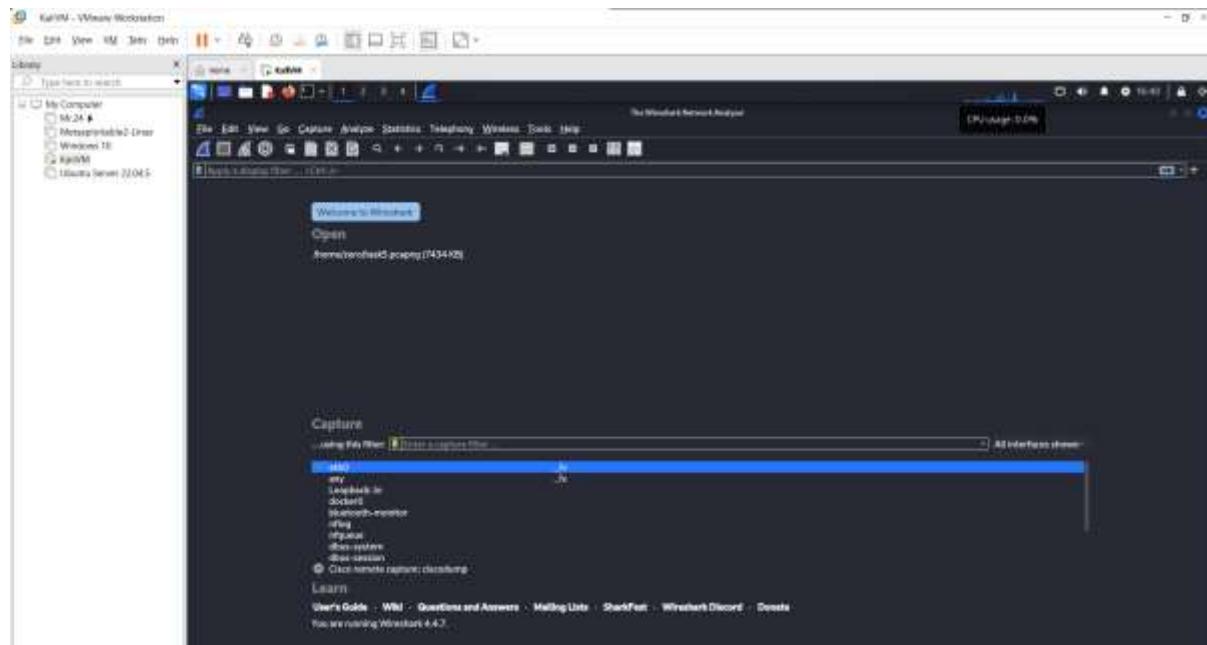
- HTTP traffic:**
- http**
- TCP traffic:**
- tcp**
- Filter by IP:**
- ip.addr == <target-ip>**

## 7. Analyzed Captured Data

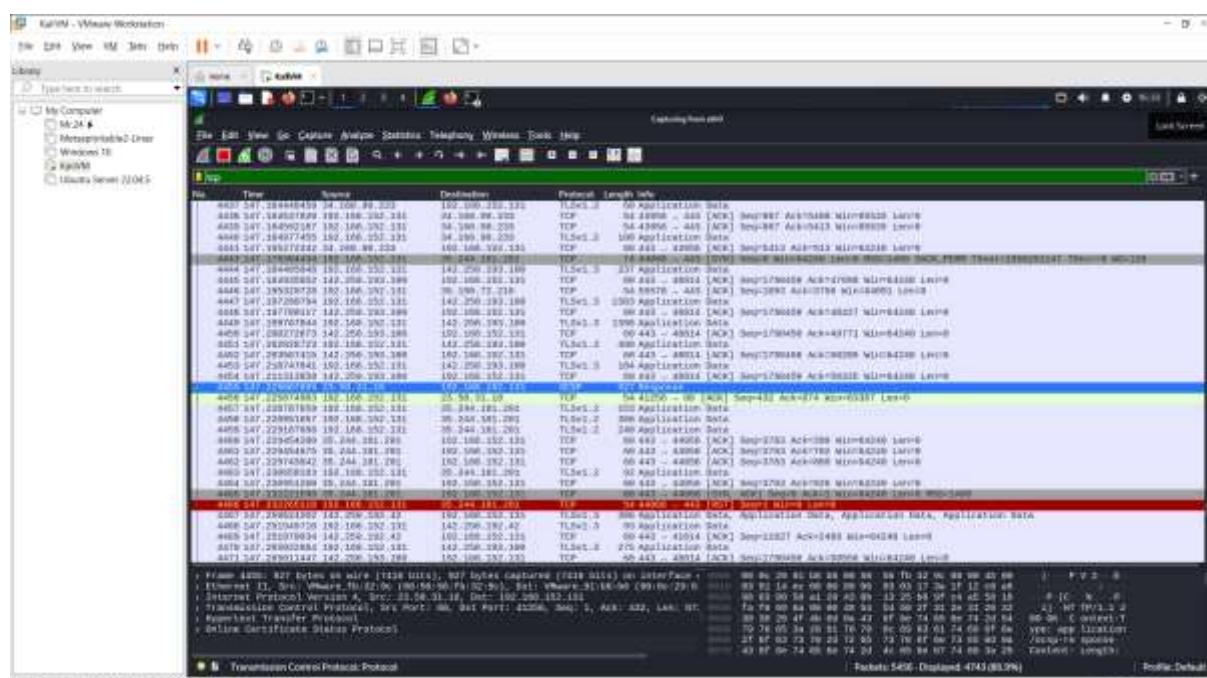
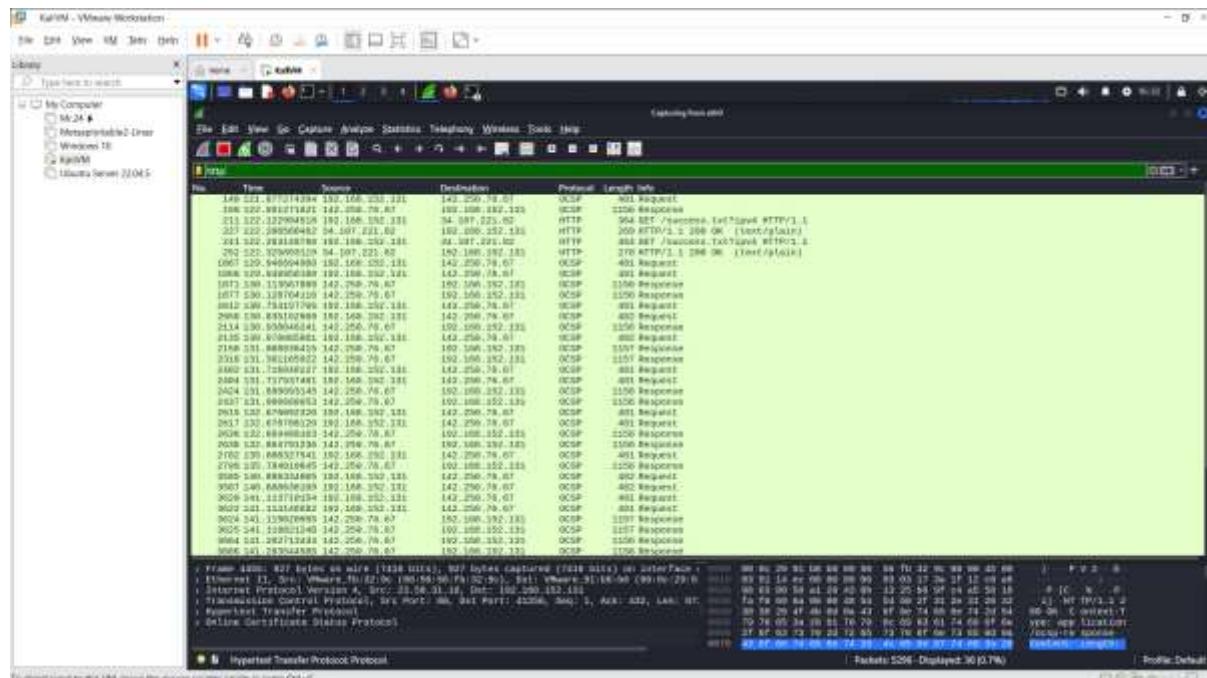
- Checked **Source** and **Destination** IPs
- Identified protocols used: HTTP, HTTPS, DNS, ICMP
- Used **Statistics → Protocol Hierarchy** for protocol breakdown
- Viewed **Conversations** and **Endpoints** to track communication flows

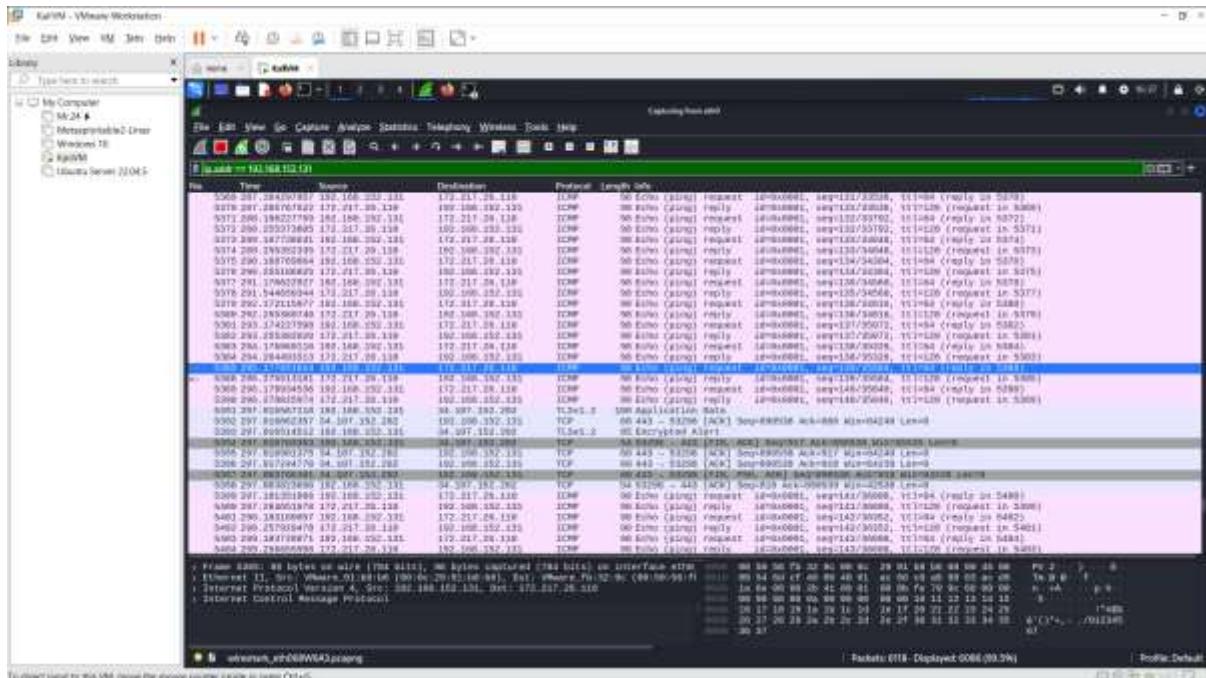
## Screenshots Included

- Full capture window



- Filtered view (HTTP/TCP)





- Protocol statistics
- Conversations & Endpoints summaries

## ✓ Outcome

- Successfully captured and analyzed live network traffic
- Applied protocol and IP filters for targeted inspection
- Observed different network protocols in action
- Gained insight into how data travels across the network