Wireshark Network Traffic Analysis

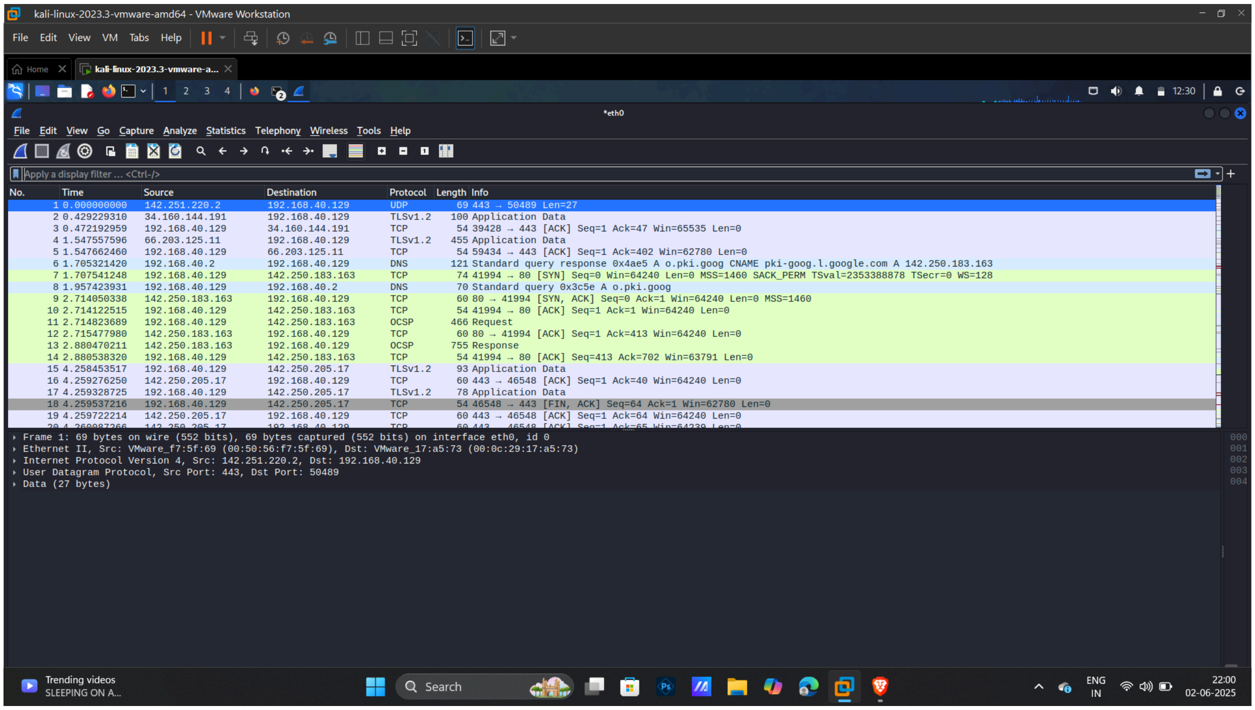
Tool Used: Wireshark
System Used: Kali Linux
Interface Used: `wlan0` (or `eth0`, based on your system)

Objective

To capture live network packets using Wireshark and identify basic protocols and traffic types.

Steps Performed

1. Launched Wireshark using `sudo wireshark`.
2. Selected the active interface (`wlan0`) for packet capture.
3. Started capturing live network traffic.
4. Generated network traffic by:
   - Pinging `google.com`
   - Browsing websites like `https://kali.org` and `https://wikipedia.org`
5. Stopped the capture after approximately 1 minute.
6. Applied protocol filters (DNS, TCP, OSCP, TLSv1.2).


kali-linux-2023.3-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
kali-linux-2023.3-vmware-a...
12:30
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Trending videos
SLEEPING ON A...
Search
ENG
IN
22:00
02-06-2025

```
No.    Time          Source            Destination       Protocol  Length Info
     1 0.000000000   142.251.220.2     192.168.40.129    UDP         69 443 → 50489 Len=27
     2 0.429229310   34.160.144.191    192.168.40.129    TLSv1.2    100 Application Data
     3 0.472192959   192.168.40.129    34.160.144.191    TCP         54 39428 → 443 [ACK] Seq=1 Ack=47 Win=65535 Len=0
     4 1.547557596   66.203.125.11     192.168.40.129    TLSv1.2    455 Application Data
     5 1.547662460   192.168.40.129    66.203.125.11     TCP         54 59434 → 443 [ACK] Seq=1 Ack=402 Win=62780 Len=0
     6 1.705321420   192.168.40.2      192.168.40.129    DNS        121 Standard query response 0x4ae5 A o.pki.goog CNAME pki-goog.l.google.com A 142.250.183.163
     7 1.707541248   192.168.40.129    142.250.183.163   TCP         74 41994 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2353388878 TSecr=0 WS=128
     8 1.957423931   192.168.40.129    192.168.40.2      DNS         70 Standard query 0x3c5e A o.pki.goog
     9 2.714050338   142.250.183.163   192.168.40.129    TCP         60 80 → 41994 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    10 2.714122515   192.168.40.129    142.250.183.163   TCP         54 41994 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
    11 2.714823689   192.168.40.129    142.250.183.163   OCSP       466 Request
    12 2.715477980   142.250.183.163   192.168.40.129    TCP         60 80 → 41994 [ACK] Seq=1 Ack=413 Win=64240 Len=0
    13 2.880470211   142.250.183.163   192.168.40.129    OCSP       755 Response
    14 2.880538320   192.168.40.129    142.250.183.163   TCP         54 41994 → 80 [ACK] Seq=413 Ack=702 Win=63791 Len=0
    15 4.258453517   192.168.40.129    142.250.205.17    TLSv1.2     93 Application Data
    16 4.259276250   142.250.205.17    192.168.40.129    TCP         60 443 → 46548 [ACK] Seq=1 Ack=40 Win=64240 Len=0
    17 4.259328725   192.168.40.129    142.250.205.17    TLSv1.2     78 Application Data
    18 4.259537216   192.168.40.129    142.250.205.17    TCP         54 46548 → 443 [FIN, ACK] Seq=64 Ack=1 Win=62780 Len=0
    19 4.259722214   142.250.205.17    192.168.40.129    TCP         60 443 → 46548 [ACK] Seq=1 Ack=64 Win=64240 Len=0
    20 4.260087266   142.250.205.17    192.168.40.129    TCP         60 443 → 46548 [ACK] Seq=1 Ack=65 Win=64239 Len=0
```

```
› Frame 1: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface eth0, id 0
› Ethernet II, Src: VMware_f7:5f:69 (00:50:56:f7:5f:69), Dst: VMware_17:a5:73 (00:0c:29:17:a5:73)
› Internet Protocol Version 4, Src: 142.251.220.2, Dst: 192.168.40.129
› User Datagram Protocol, Src Port: 443, Dst Port: 50489
› Data (27 bytes)
```

Protocols Identified

1. DNS (Domain Name System)

- **Protocol Filter:** `dns`
- **Port:** 53 (UDP)
- **Description:** Resolves domain names to IP addresses.
- **Example:**
  - Query: `www.google.com`
  - Response IP: `142.250.77.206`

2. OCSP (Online Certificate Status Protocol)

- **Protocol Filter:** `ocsp`
- **Port:** 80 or 443 (commonly used over HTTP/HTTPS)

- **Description:** OCSP is used to check the revocation status of digital certificates in real time. Web browsers or clients use OCSP to query a certificate authority (CA) server to verify whether an SSL/TLS certificate is still valid.
- **Example:**
    - **Visited Site:** `https://wikipedia.org`
    - **Observed Packet:**
        - An **OCSP request** was sent from the client to the OCSP responder URL embedded in the server's certificate.
        - The **OCSP response** indicated the certificate was in "good" standing.
    - **Additional headers observed:** `Content-Type`, `Host`, `User-Agent`, `Accept`, etc.

3. TLSv1.2 (Transport Layer Security version 1.2)

- **Protocol Filter:** `tls` (or use `ssl` in older Wireshark versions)
- **Port:** 443 (TCP)
- **Description:** TLSv1.2 is a cryptographic protocol used to secure communication over the internet, typically for HTTPS. It ensures confidentiality, integrity, and authentication between client and server.
- **Example:**
    - **Visited Site:** `https://wikipedia.org`
    - **Observed Packet:**
        - **Client Hello** sent by the browser to initiate the secure handshake.
        - **Server Hello** received, containing certificate and cipher suite info.
        - **Encrypted handshake messages** followed, establishing a secure connection.

    - **Additional details observed:** TLS version, cipher suites offered, server certificate details.

4. TCP (Transmission Control Protocol)

- **Protocol Filter:** `tcp`
- **Port:** Various
- **Description:** Provides reliable communication for applications.
- **Example:**
    - TCP handshakes for website connections.
    - Data packets for HTTP and HTTPS traffic.

Conclusion

This exercise provided hands-on experience in packet capturing and protocol analysis. Successfully identified key internet protocols such as DNS, HTTP, TCP, and ICMP, gaining a deeper understanding of how data is transmitted over networks.