

“Network Traffic Analysis using Wireshark: Identifying Legitimate vs Malicious Packets.”

Editor – Arya Mokashi

Project Overview – Packet Analysis with Wireshark

1. Objective

To explore and understand how network traffic works, how packets can be analyzed in Wireshark, and how different types of records (A vs HTTPS) behave in DNS queries.

2. Key Learning Activities

1. Packet Analysis Basics

- Captured live traffic using Wireshark.
- Learned how to filter by protocol (DNS, ICMP, HTTP/HTTPS).
- Observed packet headers (source/destination IP, port, protocol).

2. Reachability Testing with Ping (ICMP)

- Sent ICMP Echo Requests (ping) to websites.
- Understood that:
 - **Reply (Echo Reply)** → host is reachable.
 - **No reply/timeout** → ICMP may be blocked (by firewall/security policy), not always that the host is down.

3. DNS Query Analysis

- Discovered why two DNS packets appear:
 - **A Record** → resolves domain to IPv4 address.
 - **HTTPS Record** → newer DNS type, used by modern browsers to check if a site supports HTTPS and extra security features (like QUIC, HTTP/3).
- Example:
 - Query for A netacad.com → gives IPv4.

- Query for HTTPS netacad.com → asks if HTTPS endpoints exist and what parameters they support.

7325	374.931226	192.168.0.106	192.168.0.1	DNS	83	Standard query 0x4dfe A idbroker-b-us.webex.com
7326	374.932543	192.168.0.106	192.168.0.1	DNS	83	Standard query 0x8233 HTTPS idbroker-b-us.webex.com

3. Observations

1. Ping / ICMP Analysis

- When pinging a website, Wireshark captured ICMP Echo Request and Echo Reply packets.
- If the reply was received → server is reachable.
- Some servers still responded to DNS/HTTP requests even when ICMP replies were blocked.

2. DNS Record Analysis

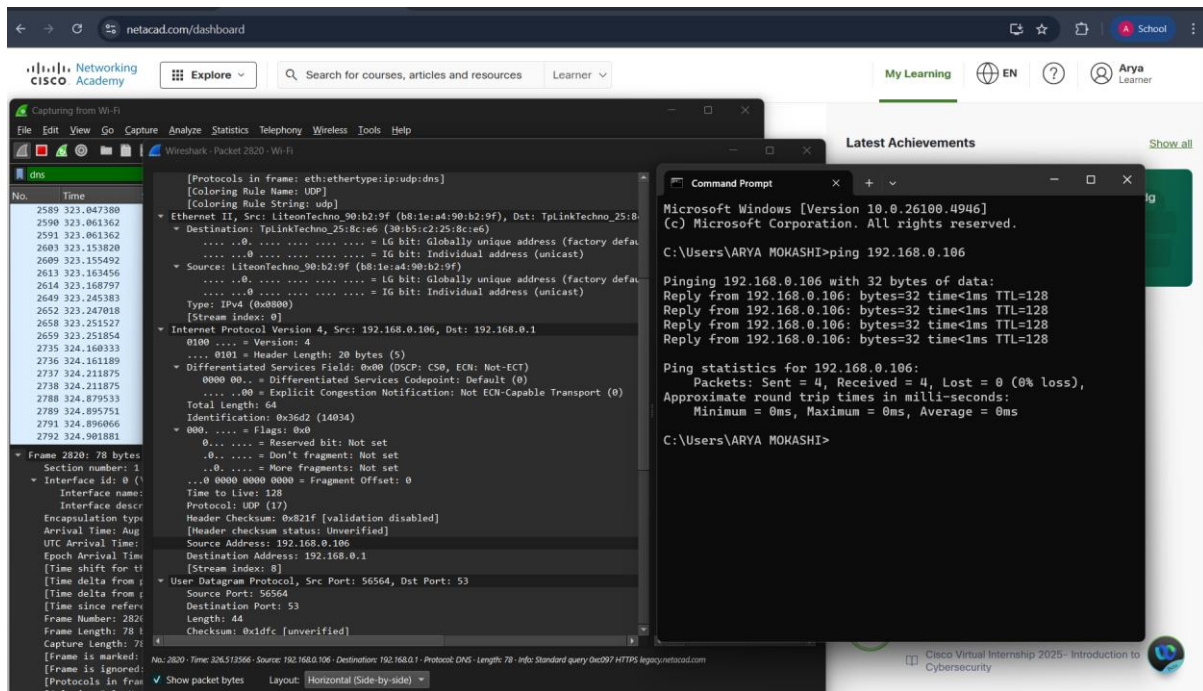
- For a single domain query, Wireshark showed two DNS queries:
 - One for A Record → returned the IPv4 address of the domain.
 - One for HTTPS Record → checked HTTPS support and extra security info.
- This explained why we saw two packets with similar information but at slightly different times.

3. Packet Details in Wireshark

- We could view the source and destination IP addresses, port numbers, and protocol information.
- For DNS responses, the exact IP address assigned to the domain was visible.
- For ICMP, we saw the request–reply sequence clearly in Wireshark.

4. Skills Gained

- Capturing and filtering packets using Wireshark.
- Identifying reachability using ICMP.
- Understanding DNS record types and why multiple queries appear.
- Beginning awareness of normal vs. abnormal traffic.



- **Traffic analysis exercise: It's a trap!**

Analysed a PCAP(Packet Capture) from <https://www.malware-traffic-analysis.net/training-exercises.html>

Conclusions n

1. Legitimate vs Malicious Data in Payload

- **Legitimate Packets**

- DNS responses contain domain names (e.g., netacad.com).
- HTTP/HTTPS traffic contains request headers (Host:, User-Agent:) and sometimes readable text (if not encrypted).
- Application protocols (FTP, SMTP, etc.) have structured, human-readable commands (USER, PASS, MAIL FROM).

- **Malicious Packets**

- Often have **empty payloads** or **garbled data** in the “Raw Bytes” section.
- Data may appear as **random characters**, **encoded strings (Base64, hex dumps)**, or **no clear protocol structure**.
- For example:
 - Instead of GET /index.html HTTP/1.1, you might see sdf9sd89f8sd...

2. Color Coding in Wireshark

```
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
127 4.807774 52.156.123.84 10.6.13.133 TCP 92 Application Data
128 4.807775 52.156.123.84 10.6.13.133 TCP 60 443 > 52430 [ACK] Seq=2591 Ack=673 Win=524800 Len=0
129 4.807776 52.156.123.84 10.6.13.133 TLSv1.2 631 Application Data
130 4.807913 10.6.13.133 52.156.123.84 TCP 60 52430 > 443 [FIN, ACK] Seq=673 Ack=3168 Win=64768 Len=0
131 4.808181 10.6.13.133 52.156.123.84 TCP 60 52430 > 443 [FIN, ACK] Seq=673 Ack=3168 Win=64768 Len=0
132 4.825188 23.192.223.206 10.6.13.133 TCP 60 80 > 52431 [ACK] Seq=1 Acl=112 Win=64256 Len=0
133 4.825189 23.192.223.206 10.6.13.133 HTTP 241 HTTP/1.1 200 OK (text/plain)
134 4.825190 23.192.223.206 10.6.13.133 TCP 60 80 > 52431 [FIN, ACK] Seq=108 Ack=112 Win=64256 Len=0
135 4.825191 23.192.223.206 10.6.13.133 TCP 60 52431 > 80 [ACK] Seq=108 Ack=189 Win=65280 Len=0
136 4.825192 10.6.13.133 23.192.223.206 TCP 60 52431 > 80 [FIN, ACK] Seq=112 Ack=189 Win=65280 Len=0
137 4.854990 23.192.223.206 10.6.13.133 TCP 60 80 > 52431 [ACK] Seq=189 Ack=113 Win=64256 Len=0
138 4.889190 52.156.123.84 10.6.13.133 TCP 60 443 > 52430 [FIN, ACK] Seq=3168 Ack=674 Win=524800 Len=0
139 4.889192 10.6.13.133 52.156.123.84 TCP 60 52430 > 443 [ACK] Seq=674 Ack=3169 Win=64768 Len=0
140 4.901808 104.208.203.97 Broadcast ARP 60 ARP Announcement for 10.6.13.133
141 5.380326 10.6.13.133 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
142 5.427783 10.6.13.133 10.6.13.255 NBNS 110 Registration NB MASSFRICITIONc1e>
143 5.468827 10.6.13.133 10.6.13.3 DNS 82 Standard query response A client.ums.windows.com
144 5.514226 10.6.13.133 DNS 141 Standard query response DnsDns A client.ums.windows.com CNAME ums.notify.trafficmanager.net A 104.208.203.90
145 5.589187 104.208.203.90 10.6.13.133 TCP 66 52432 > 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
146 5.589189 104.208.203.90 10.6.13.133 TCP 66 443 > 52432 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1396 WS=1 SACK_PERM
147 5.589191 10.6.13.133 104.208.203.90 TCP 60 52432 > 443 [ACK] Seq=1 Acl=1 Win=65280 Len=0
148 5.590470 10.6.13.133 104.208.203.90 TLSv1.2 232 Client Hello (SHA1client.ums.windows.com)

▼ Frame 144: 141 bytes on wire (1128 bits), 141 bytes captured (1128 bits) on eth0
Encapsulation type: Ethernet (1)
Arrival Time: Jun 13, 2025 21:04:01.200777000 India Standard Time
Epoch Arrival Time: Jun 13, 2025 15:34:01.200777000 UTC
Epoch Arrival Time: 1749628841.200777000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.045399000 seconds]
[Time delta from previous displayed frame: 0.045399000 seconds]
[Time since reference or first frame: 5.514226000 seconds]
Frame Number: 144
Frame Length: 141 bytes (1128 bits)
Capture Length: 141 bytes (1128 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: ethIIethertype:ip:udp:dns]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]

▼ Ethernet II, Src: Dell 71:ad:36 (00:26:b9:71:ad:36), Dst: Intel ac:97:df (24:7f:03:ac:97:df)
  Destination: Intel ac:97:df (24:7f:03:ac:97:df)
  Source: Dell 71:ad:36 (00:26:b9:71:ad:36)
  Type: IPv4 (0x0800)
  [Stream index: 2]

▼ Internet Protocol Version 4, Src: 10.6.13.3, Dst: 10.6.13.133
```

- **Conclusion**

In this traffic analysis exercise, we learned how to capture and inspect packets using Wireshark and differentiate between legitimate and suspicious network traffic. Legitimate packets followed proper protocol structures (e.g., DNS queries showing domain names, HTTP headers with readable information), while malicious or suspicious packets often lacked meaningful data in the raw field or displayed irregular/unexpected patterns. Color coding in Wireshark (e.g., red highlights for TCP problems) further helped in identifying abnormal packets that require closer inspection.

• What Was Learned

- How to use **Wireshark** to capture and analyze packets.
- The difference between **legitimate traffic** (structured, protocol-compliant, human-readable) and **malicious traffic** (unstructured, encoded, or abnormal behavior).
- Understanding of **DNS record types** (A vs HTTPS record) and how websites may respond differently.
- Use of **packet payload inspection** (raw data view) to infer intent behind traffic.