

Q2) Use Wireshark to capture and analyze DNS, TCP, UDP traffic and packet header, packet flow, options and flags

DNS

The image shows a Wireshark capture of DNS traffic. The packet list on the left shows a series of DNS queries and responses. The selected packet is a standard query response from 192.168.0.107 to 192.168.0.107, containing the IP address for google.com (142.250.182.14). The packet details pane on the right shows the structure of the DNS message, including the header, questions, and answers. The packet bytes pane on the right shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
2983	59.862083	192.168.0.107	192.168.0.107	DNS	86	Standard query response 0x7f8a A google.com A 216.58.196.174
2996	60.868530	192.168.0.107	192.168.0.107	DNS	82	Standard query 0x333a A t-ring-fdv2.msedge.net
2997	60.871973	192.168.0.107	192.168.0.107	DNS	124	Standard query response 0x333a Server failure A t-ring-fdv2.msedge.net CNAME t-ring-t-9999.fdv2-t-msedge.net
2998	60.872848	192.168.0.107	192.168.0.107	DNS	82	Standard query 0x333a A t-ring-fdv2.msedge.net
3013	61.873196	192.168.0.107	192.168.0.107	DNS	82	Standard query 0x333a A t-ring-fdv2.msedge.net
3040	63.873270	192.168.0.107	192.168.0.107	DNS	82	Standard query 0x333a A t-ring-fdv2.msedge.net
3050	64.918256	49.205.72.130	192.168.0.107	DNS	124	Standard query response 0x333a Server failure A t-ring-fdv2.msedge.net CNAME t-ring-t-9999.fdv2-t-msedge.net
3051	64.927793	192.168.0.107	192.168.0.107	DNS	86	Standard query 0x6280 A dc1.kns.kaspersky-labs.com
3052	64.929958	192.168.0.107	192.168.0.107	DNS	221	Standard query response 0x6280 A dc1.kns.kaspersky-labs.com CNAME kns-dc1.geokns.kaspersky.com A 202.163.7.44 A 202.163.7.91 A ...
3053	64.933831	192.168.0.107	192.168.0.107	DNS	78	Standard query 0x0cdd A bx-ring.msedge.net
3054	64.937316	192.168.0.107	192.168.0.107	DNS	164	Standard query response 0x0cdd A bx-ring.msedge.net CNAME bx-ring.bx-9999.bx-msedge.net CNAME bx-9999.bx-msedge.net A 150.171.7...
3093	64.979181	192.168.0.107	192.168.0.107	DNS	86	Standard query 0x7b57 A a-ring-fallback.msedge.net
3098	64.981899	192.168.0.107	192.168.0.107	DNS	135	Standard query response 0x7b57 A a-ring-fallback.msedge.net CNAME a-9999.a-dc-msedge.net A 131.253.33.254
3269	74.318393	192.168.0.107	8.8.8.8	DNS	80	Standard query 0x0001 PTR 8.8.8.8.in-addr.arpa
3270	74.321666	8.8.8.8	192.168.0.107	DNS	104	Standard query response 0x0001 PTR 8.8.8.8.in-addr.arpa PTR dns.google
3271	74.323315	192.168.0.107	8.8.8.8	DNS	70	Standard query 0x0002 A google.com
3272	74.324061	8.8.8.8	192.168.0.107	DNS	86	Standard query response 0x0002 A google.com A 142.250.182.14
3273	74.330155	192.168.0.107	8.8.8.8	DNS	70	Standard query 0x0003 AAAA google.com
3274	74.412380	8.8.8.8	192.168.0.107	DNS	98	Standard query response 0x0003 AAAA google.com AAAA 2404:6800:4007:819::200e
3522	85.603796	192.168.0.107	192.168.0.107	DNS	89	Standard query 0xf054 A word-edit.officeapps.live.com
3529	85.608031	192.168.0.107	192.168.0.107	DNS	279	Standard query response 0xf054 A word-edit.officeapps.live.com CNAME word-edit-geo.wac.trafficmanager.net CNAME word-edit.wac.t...

000. = Flags: 0x0
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: UDP (17)
Header Checksum: 0x9c8b [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.0.107
Destination Address: 8.8.8.8
[Stream Index: 79]
User Datagram Protocol, Src Port: 63599, Dst Port: 53
Domain Name System (query)
Transaction ID: 0x0003
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
[Response In: 3274]

Domain Name System: Protocol

Packets: 3652 - Displayed: 138 (3.8%) - Dropped: 0 (0.0%)

Profile: Default

TCP

The image shows a Wireshark capture of TCP traffic. The packet list on the left shows a series of TCP segments. The selected packet is a TCP segment from 192.168.0.107 to 192.168.0.107, containing the IP address for google.com (142.250.182.14). The packet details pane on the right shows the structure of the TCP segment, including the header, options, and data. The packet bytes pane on the right shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
3628	87.775948	13.107.42.254	192.168.0.107	TCP	54	443 → 56673 [FIN, ACK] Seq=7521 Ack=1209 Win=4193280 Len=0
3629	87.775948	13.107.213.254	192.168.0.107	TCP	54	443 → 56675 [FIN, ACK] Seq=8231 Ack=1370 Win=42496 Len=0
3630	87.775948	52.123.128.254	192.168.0.107	TCP	54	443 → 56664 [ACK] Seq=1 Ack=2 Win=16382 Len=0
3631	87.775948	52.123.128.254	192.168.0.107	TCP	54	443 → 56664 [FIN, ACK] Seq=1 Ack=2 Win=16382 Len=0
3632	87.775948	150.171.69.254	192.168.0.107	TCP	54	443 → 56662 [ACK] Seq=1 Ack=2 Win=16382 Len=0
3633	87.775948	150.171.69.254	192.168.0.107	TCP	54	443 → 56662 [FIN, ACK] Seq=1 Ack=2 Win=16382 Len=0
3634	87.776035	192.168.0.107	104.86.188.195	TCP	54	56663 → 443 [RST, ACK] Seq=1755 Ack=18544 Win=0 Len=0
3635	87.776107	192.168.0.107	104.86.188.195	TCP	54	56663 → 443 [RST] Seq=1755 Win=0 Len=0
3636	87.776174	192.168.0.107	13.107.42.254	TCP	54	56673 → 443 [ACK] Seq=1209 Ack=7522 Win=260864 Len=0
3637	87.776232	192.168.0.107	13.107.213.254	TCP	54	56675 → 443 [ACK] Seq=1370 Ack=8232 Win=260864 Len=0
3638	87.776300	192.168.0.107	52.123.128.254	TCP	54	56664 → 443 [ACK] Seq=2 Ack=2 Win=1020 Len=0
3639	87.776378	192.168.0.107	150.171.69.254	TCP	54	[TCP Dup ACK 3622#1] 56662 → 443 [ACK] Seq=2 Ack=1 Win=1022 Len=0
3640	87.776486	192.168.0.107	150.171.69.254	TCP	54	56662 → 443 [ACK] Seq=2 Ack=2 Win=1022 Len=0
3641	87.891158	192.168.0.107	34.237.73.95	TLSv1.2	294	Application Data
3642	88.111132	192.168.0.107	192.168.0.107	TLSv1.2	310	Application Data
3643	88.152438	192.168.0.107	34.237.73.95	TCP	54	55616 → 443 [ACK] Seq=1619 Ack=1611 Win=512 Len=0
3645	90.698702	192.168.0.107	192.168.0.100	TCP	164	56498 → 8009 [PSH, ACK] Seq=2091 Ack=2209 Win=513 Len=110
3646	90.705689	192.168.0.100	192.168.0.107	TCP	164	8009 → 56498 [PSH, ACK] Seq=2209 Ack=2201 Win=1514 Len=110
3647	90.746217	192.168.0.107	192.168.0.100	TCP	54	56498 → 8009 [ACK] Seq=2201 Ack=2319 Win=512 Len=0
3648	90.991022	192.168.0.107	208.115.231.82	TLSv1.2	444	Ignored Unknown Record
3649	91.155438	208.115.231.82	192.168.0.107	TCP	54	443 → 55605 [ACK] Seq=1 Ack=392 Win=126 Len=0

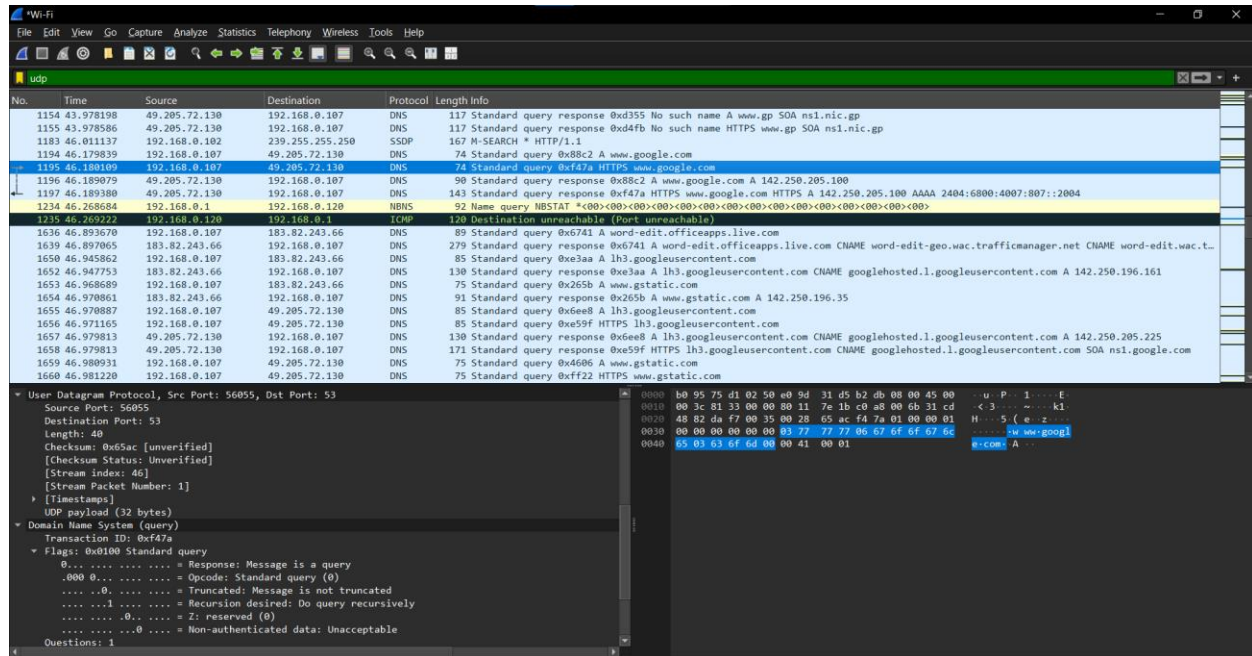
Transmission Control Protocol, Src Port: 443, Dst Port: 55616, Seq: 1349, Ack: 1619, Len: 262
Source Port: 443
Destination Port: 55616
[Stream Index: 17]
[Conversation completeness: Incomplete (12)]
[TCP Segment Len: 262]
Sequence Number: 1349 (relative sequence number)
Sequence Number (raw): 2382009801
[Next Sequence Number: 1611 (relative sequence number)]
Acknowledgment Number: 1619 (relative ack number)
Acknowledgment Number (raw): 306551294
0101 = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window: 506
[Calculated window size: 506]
[Window size scaling factor: -1 (unknown)]
Checksum: 0x8fd1 [Unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0

Transmission Control Protocol: Protocol

Packets: 3652 - Displayed: 3322 (91.0%) - Dropped: 0 (0.0%)

Profile: Default

UDP



What is TCP?

- **Connection-oriented** protocol.
- Ensures **reliable** and **ordered** data transmission.
- Uses **error checking** and **retransmission** if packets are lost.
- Establishes a connection **before** sending data (3-way handshake).

How TCP Works?

1. **Three-Way Handshake** (Before data transfer)
 - a. **Step 1:** Client sends SYN (synchronize) to the server.
 - b. **Step 2:** Server responds with SYN-ACK (synchronize + acknowledge).
 - c. **Step 3:** Client replies with ACK (acknowledge).
 - d. Now, the connection is **established**, and data transfer begins.
2. **Data Transfer**
 - a. Data is sent in **segments**.
 - b. Each segment has a **sequence number**.
 - c. The receiver **acknowledges** received segments.
 - d. Lost packets are **retransmitted**.
3. **Connection Termination**
 - a. Either side sends a FIN (finish) flag to close the connection.

What is UDP?

- **Connectionless** protocol (No handshake).
- **Faster** than TCP but **less reliable**.
- No retransmission of lost packets.

How UDP Works?

1. Sender **transmits** data without checking if the receiver is ready.
2. Receiver **processes** the data but does not send an acknowledgment.
3. If a packet is lost, **no retransmission** happens.

What is DNS?

- Converts **domain names** (e.g., google.com) into **IP addresses** (142.250.190.78).
- Uses **UDP (Port 53)** for quick responses.
- Uses **TCP** for large responses (e.g., zone transfers).

How DNS Works?

1. User types www.google.com in the browser.
2. The system sends a **DNS Query** to a DNS server.
3. DNS server replies with the **IP address** of google.com.
4. The system uses this IP address to **connect** to the website.