

Task 3: Networking Basics for Cyber Security

1. Learn basic networking concepts (IP, MAC, DNS, TCP/UDP):

IP Address (Internet Protocol)

What it is?

An IP address is a unique numerical identifier for a device on a network.

Just like a home address identifies where you live.

Types:

- IPv4 → 192.168.1.10
- IPv6 → 2001:0db8:85a3::8a2e:0370:7334

Public vs Private:

- **Private IP: Used inside local networks (home, office)**
 - 192.168.x.x
 - 10.x.x.x
- **Public IP: Used on the internet (assigned by ISP)**

Security relevance:

- **Attackers scan IPs to find open ports**
- **Firewalls filter traffic based on Ips**

MAC Address (Media Access Control)

What it is?

A MAC address is a hardware address permanently assigned to a network interface.

Example:

00:1A:2B:3C:4D:5E

Key points

- Works at Layer 2 (Data Link) of the OSI model
- Used inside local networks (LAN)

Difference between IP & MAC

IP Address	MAC Address
Logical	Physical
Can change	Usually fixed
Layer 3	Layer 2

Security relevance

- MAC spoofing can bypass weak network controls
- Used in ARP attacks

DNS (Domain Name System)

What it is

DNS translates **domain names into IP addresses**.

Example:

google.com → 142.250.192.14

Why it's needed

Humans remember names, computers use IPs.

DNS resolution flow

1. You type example.com
2. DNS server looks it up
3. IP address is returned

4. Browser connects to the server

Security relevance

- DNS spoofing / poisoning attacks
- Malicious domains used in phishing

TCP (Transmission Control Protocol)

What it is

TCP is a **reliable, connection-oriented** protocol.

Characteristics

- Guarantees delivery
- Error checking
- Packet ordering
- Uses **3-way handshake**

TCP Handshake

SYN → SYN-ACK → ACK

Common TCP services

Service Port

HTTP 80

HTTPS 443

FTP 21

SSH 22

Security relevance

- TCP SYN flood attacks
- Session hijacking risks

UDP (User Datagram Protocol)

What it is

UDP is a **fast, connectionless** protocol.

Characteristics

- No guarantee of delivery
- No handshake
- Faster than TCP

Common UDP services

Service Port

DNS 53

DHCP 67/68

VoIP Various

Security relevance

- UDP amplification attacks
- Used in DDoS attacks

2. Filter packets by protocol (DNS, TCP, HTTP) using Wireshark:

DNS (Domain Name System)

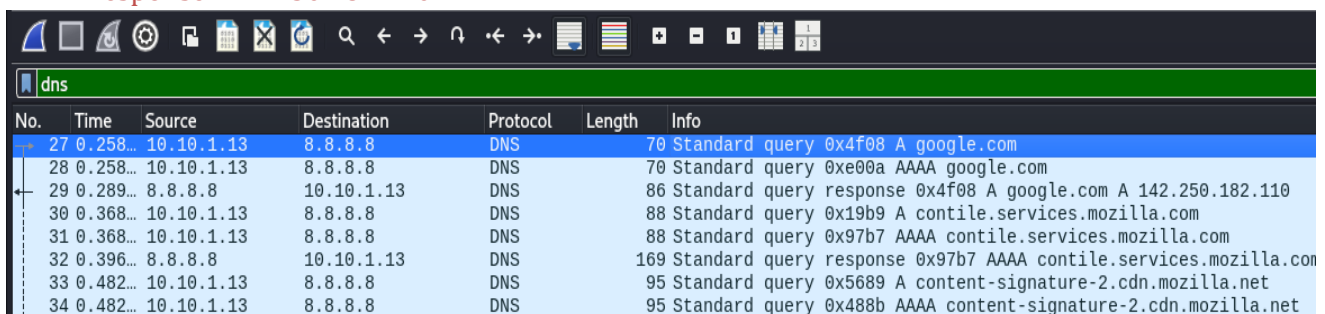
Used for resolving domain names (e.g., www.google.com) into IP addresses.

Example:

Source: 10.10.1.13 → Destination: 8.8.8.8

Query: A www.google.com

Response: 142.250.182.110



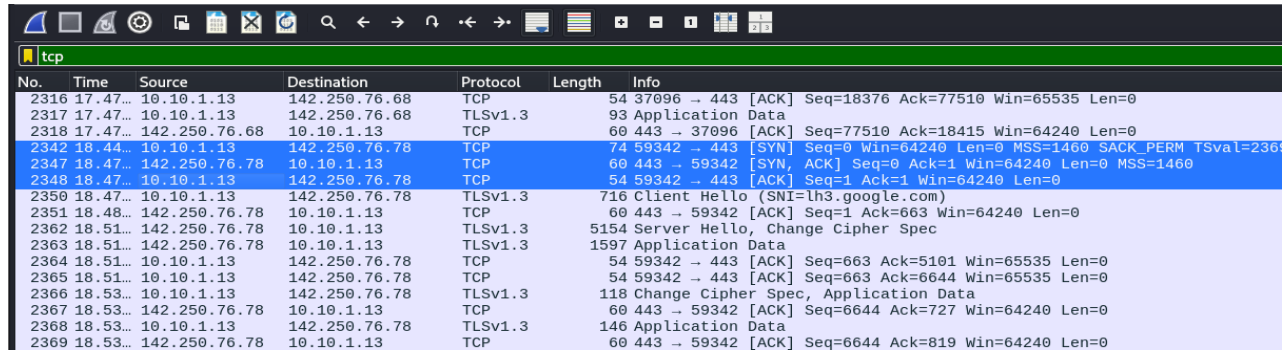
The image shows a Wireshark packet capture window with the filter 'dns' applied. The packet list shows several DNS queries and responses. The selected packet (No. 27) is a standard query for 'A google.com' from source 10.10.1.13 to destination 8.8.8.8.

No.	Time	Source	Destination	Protocol	Length	Info
27	0.258...	10.10.1.13	8.8.8.8	DNS	70	Standard query 0x4f08 A google.com
28	0.258...	10.10.1.13	8.8.8.8	DNS	70	Standard query 0xe00a AAAA google.com
29	0.289...	8.8.8.8	10.10.1.13	DNS	86	Standard query response 0x4f08 A google.com A 142.250.182.110
30	0.368...	10.10.1.13	8.8.8.8	DNS	88	Standard query 0x19b9 A contile.services.mozilla.com
31	0.368...	10.10.1.13	8.8.8.8	DNS	88	Standard query 0x97b7 AAAA contile.services.mozilla.com
32	0.396...	8.8.8.8	10.10.1.13	DNS	169	Standard query response 0x97b7 AAAA contile.services.mozilla.com
33	0.482...	10.10.1.13	8.8.8.8	DNS	95	Standard query 0x5689 A content-signature-2.cdn.mozilla.net
34	0.482...	10.10.1.13	8.8.8.8	DNS	95	Standard query 0x488b AAAA content-signature-2.cdn.mozilla.net

TCP (Transmission Control Protocol)

Used to establish reliable connections between client and server.

Example:



No.	Time	Source	Destination	Protocol	Length	Info
2316	17.47...	10.10.1.13	142.250.76.68	TCP	54	37096 → 443 [ACK] Seq=18376 Ack=77510 Win=65535 Len=0
2317	17.47...	10.10.1.13	142.250.76.68	TLSv1.3	93	Application Data
2318	17.47...	142.250.76.68	10.10.1.13	TCP	60	443 → 37096 [ACK] Seq=77510 Ack=18415 Win=64240 Len=0
2342	18.44...	10.10.1.13	142.250.76.78	TCP	74	59342 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2360
2347	18.47...	142.250.76.78	10.10.1.13	TCP	60	443 → 59342 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2348	18.47...	10.10.1.13	142.250.76.78	TCP	54	59342 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2350	18.47...	10.10.1.13	142.250.76.78	TLSv1.3	716	Client Hello (SNI=lh3.google.com)
2351	18.48...	142.250.76.78	10.10.1.13	TCP	60	443 → 59342 [ACK] Seq=1 Ack=663 Win=64240 Len=0
2362	18.51...	142.250.76.78	10.10.1.13	TLSv1.3	5154	Server Hello, Change Cipher Spec
2363	18.51...	142.250.76.78	10.10.1.13	TLSv1.3	1597	Application Data
2364	18.51...	10.10.1.13	142.250.76.78	TCP	54	59342 → 443 [ACK] Seq=663 Ack=5101 Win=65535 Len=0
2365	18.51...	10.10.1.13	142.250.76.78	TCP	54	59342 → 443 [ACK] Seq=663 Ack=6644 Win=65535 Len=0
2366	18.53...	10.10.1.13	142.250.76.78	TLSv1.3	118	Change Cipher Spec, Application Data
2367	18.53...	142.250.76.78	10.10.1.13	TCP	60	443 → 59342 [ACK] Seq=6644 Ack=727 Win=64240 Len=0
2368	18.53...	10.10.1.13	142.250.76.78	TLSv1.3	146	Application Data
2369	18.53...	142.250.76.78	10.10.1.13	TCP	60	443 → 59342 [ACK] Seq=6644 Ack=819 Win=64240 Len=0

Source: 10.10.1.13 → Destination: 142.250.76.78

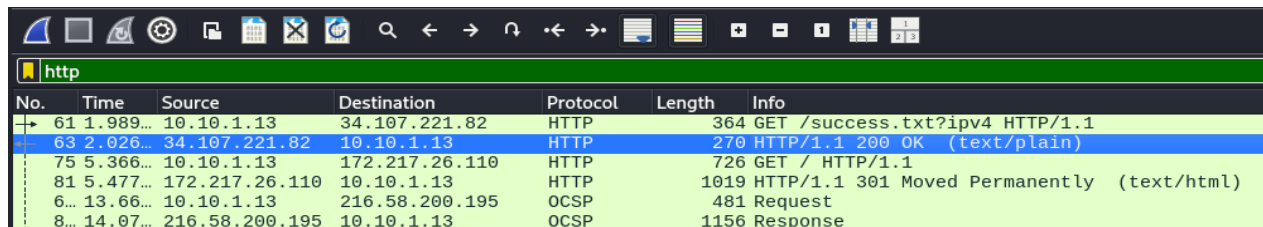
Flags: SYN → SYN/ACK → ACK (3-way handshake)

HTTP (Hypertext Transfer Protocol)

Application layer protocol used for web requests.

Example:

GET / HTTP/1.1



No.	Time	Source	Destination	Protocol	Length	Info
61	1.989...	10.10.1.13	34.107.221.82	HTTP	364	GET /success.txt?ip=4 HTTP/1.1
63	2.026...	34.107.221.82	10.10.1.13	HTTP	270	HTTP/1.1 200 OK (text/plain)
75	5.366...	10.10.1.13	172.217.26.110	HTTP	726	GET / HTTP/1.1
81	5.477...	172.217.26.110	10.10.1.13	HTTP	1019	HTTP/1.1 301 Moved Permanently (text/html)
6...	13.66...	10.10.1.13	216.58.200.195	OCSP	481	Request
8...	14.07...	216.58.200.195	10.10.1.13	OCSP	1156	Response

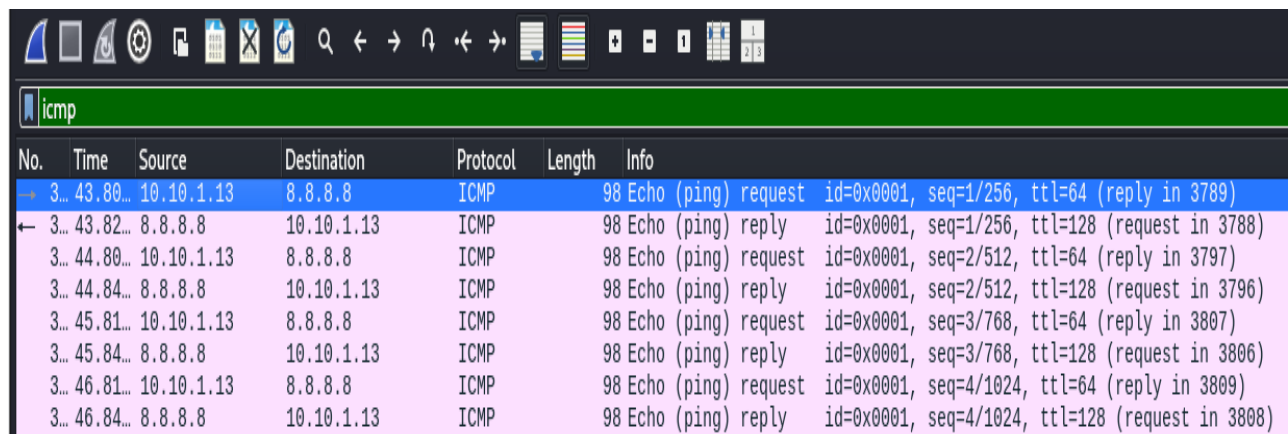
Host: www.google.com

Response: 200 OK

4. ICMP (Internet Control Message Protocol)

Used for ping and diagnostic messages.

Example:



No.	Time	Source	Destination	Protocol	Length	Info
3...	43.80...	10.10.1.13	8.8.8.8	ICMP	98	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 3789)
3...	43.82...	8.8.8.8	10.10.1.13	ICMP	98	Echo (ping) reply id=0x0001, seq=1/256, ttl=128 (request in 3788)
3...	44.80...	10.10.1.13	8.8.8.8	ICMP	98	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 3797)
3...	44.84...	8.8.8.8	10.10.1.13	ICMP	98	Echo (ping) reply id=0x0001, seq=2/512, ttl=128 (request in 3796)
3...	45.81...	10.10.1.13	8.8.8.8	ICMP	98	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 3807)
3...	45.84...	8.8.8.8	10.10.1.13	ICMP	98	Echo (ping) reply id=0x0001, seq=3/768, ttl=128 (request in 3806)
3...	46.81...	10.10.1.13	8.8.8.8	ICMP	98	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in 3809)
3...	46.84...	8.8.8.8	10.10.1.13	ICMP	98	Echo (ping) reply id=0x0001, seq=4/1024, ttl=128 (request in 3808)

Echo (ping) request → Echo reply from 8.8.8.8

3. Identify plain-text traffic vs encrypted traffic:

How to Identify Plain-Text in Wireshark

1. Use Display Filters

http, ftp, telnet.

2. Look at Packet Details

- Expand Application Layer
- You can read:
 - URLs
 - Usernames
 - Passwords
 - Form data

3. Follow Stream:

Right-click a packet → Follow → TCP Stream

If you can read the conversation clearly, it's plain-text.

Example (HTTP):

- GET /login.php HTTP/1.1
- Host: example.com
- username=admin&password=12345

Security risk:

Anyone sniffing the network can see this.

How to Identify Encrypted Traffic in Wireshark

Use Filters:

- Tls
- Ssl
- tcp.port == 443

Packet Appearance:

- Data appears as random bytes
- You cannot see usernames or content

TLS Handshake Indicators:

Look for--

- Client Hello
- Server Hello

- Certificate exchange

Example (Encrypted Payload):

- Data: 8f 3a b9 e2 91 c4 7a ...
- This is unreadable without decryption keys.

Observations:

- Wireshark was used to capture live network traffic.
- DNS packets showed domain names being translated into IP addresses.
- TCP packets displayed a three-way handshake using SYN, SYN-ACK, and ACK flags.
- HTTP traffic was readable in plain text, while HTTPS traffic was encrypted and unreadable.
- This shows how secure communication protects user data.