Tryhackme week02

IP Address (Internet Protocol)

- An IP address identifies a device on a network.
- IPv4 addresses consist of **four octets** (8 bits each), totaling **32 bits**.

Private vs Public IP

- Each **device** on a local network has a **private IP**.
- The **router** has **one public IP** shared by all devices using **NAT**.

Private IPs:

1. Only work inside the same local network
2. Are not routable on the internet
3. Require devices to be on the same network to communicate

IPv4 Exhaustion & IPv6

- IPv4 addresses are limited and **exhausted**.
- The internet still works due to:
  - NAT
  - Address reuse
- **IPv6** was introduced as the long-term solution:
  - 128-bit address space
  - Vastly more available addresses
- Most systems use **dual stack** (IPv4 + IPv6).

MAC Address

- A MAC address uniquely identifies a network interface.
- 12-character hexadecimal value (e.g. `a4:c3:f0:85:ac:2d`)
- First half = manufacturer
- Second half = unique identifier
- MAC addresses can be **spoofed**.

ICMP & Ping

- Ping uses **ICMP** to test connectivity.
- Uses **echo request** and **echo reply** packets.
- Measures latency and packet loss between devices.

DNS (Domain Name System)

DNS resolves domain names to IP addresses and is not limited to websites.

DNS Record Types

- **A** → IPv4 address
- **AAAA** → IPv6 address
- **CNAME** → Alias to another domain
- **MX** → Mail server for a domain
- **TXT** → Verification and email security (SPF/DKIM/DMARC)

DNS Resolution Flow

1. Client checks **local cache**
2. Query sent to **recursive DNS server** (ISP)
3. If not cached:

- Recursive server queries **root DNS server**

4. Root server directs to correct **TLD server**
5. TLD server points to **authoritative DNS server**
6. Authoritative server returns the final answer
7. Response is cached based on **TTL**

> “Standard query response, No error” in Wireshark indicates an authoritative answer.

HTTP vs HTTPS

HTTP

- Transfers web data (HTML, images, video).
- Data is sent in **plaintext**.

HTTPS

- Encrypted version of HTTP.
- Protects confidentiality and integrity.
- Verifies server identity using certificates.

URL Structure

A URL tells the browser how to access a resource.

Components:

- **Scheme** (HTTP, HTTPS)
- **Host** (domain or IP)
- **Port** (80 HTTP, 443 HTTPS)
- **Path**
- **Query string** (e.g. `?id=1`)

- **Fragment** (page location)

HTTP Requests

HTTP Methods

- **GET** – retrieve data
- **POST** – submit data
- **PUT** – update data
- **DELETE** – remove data

HTTP Headers & Status Codes

- Headers provide metadata for requests and responses.

Common Request Headers

- Host
- User-Agent
- Content-Length
- Accept-Encoding
- Cookie

Common Response Headers

- Set-Cookie
- Cache-Control
- Content-Type
- Content-Encoding

Cookies

- Cookies store small pieces of data on the client.

- Used because HTTP is **stateless**.
- Common uses:
  - Authentication
  - Session tracking
  - Personalization
- Cookies store **tokens**, not passwords.

How Websites Work (High Level)

1. Browser makes a request
2. Server processes it
3. Server sends a response
4. Browser renders the page

Core Web Infrastructure Concepts

- **Frontend**: What the user sees in the browser
- **Backend**: Server-side logic and data processing

Load Balancers

- Distribute traffic across servers
- Provide redundancy and availability

CDN (Content Delivery Network)

- Serves static content from geographically close servers
- Reduces latency and server load

WAF (Web Application Firewall)

- Filters and blocks malicious web traffic before it reaches the server

---

Static vs Dynamic Content

- **Static**: Content that does not change
- **Dynamic**: Content generated per request

Key Takeaway (Week 2 Summary)

> When a URL is entered, DNS resolves the domain to an IP address, a TCP connection is established, and HTTP or HTTPS is used to request and deliver web content, with security, caching, and infrastructure components supporting the process.