**What is DNS?**

DNS (Domain Name System) provides a simple way for us to communicate with devices on the internet without remembering complex numbers. Much like every house has a unique address for sending mail directly to it, every computer on the internet has its own unique address to communicate with it called an IP address.

**Domain Hierarchy**

The **domain hierarchy** is a system for organizing and categorizing domain names in the **Domain Name System (DNS)**, which is responsible for translating human-readable domain names (like www.example.com) into machine-readable IP addresses (like 192.0.2.1). The domain hierarchy is structured in a tree-like format with multiple levels, where each level represents a different segment of the domain name.

**Overview of the DNS Domain Hierarchy:**

At its core, the DNS hierarchy is designed to make domain name resolution (i.e., finding the IP address associated with a domain name) efficient and scalable. The hierarchy is divided into several **levels** from top to bottom:

1. **Root Domain (Top-Level)**:
   * This is the highest level of the domain hierarchy.
   * It is represented by a dot (.) and is essentially the "root" of the entire DNS system.
   * The root is managed by a group of organizations, such as the Internet Corporation for Assigned Names and Numbers (**ICANN**), which oversees the global DNS infrastructure.
2. **Top-Level Domains (TLDs)**:
   * The **Top-Level Domains (TLDs)** are the first part of the domain name after the root, and they fall directly below the root in the DNS hierarchy.
   * TLDs are categorized into several groups:
     + **Generic TLDs (gTLDs)**: These are common domain extensions like .com, .org, .net, .edu, .gov, and newer ones like .tech, .app, and .io.
     + **Country Code TLDs (ccTLDs)**: These are country-specific domain extensions, typically consisting of two letters, such as .us (United States), .uk (United Kingdom), .de (Germany), and .jp (Japan).
     + **Sponsored TLDs**: These are specialized TLDs sponsored by specific organizations or entities, such as .aero for the aviation industry or .museum for museums.
3. **Second-Level Domains (SLDs)**:
   * The **Second-Level Domain (SLD)** is the portion of the domain name directly to the left of the TLD. It typically represents the **unique name** of the website, organization, or service.
   * For example, in example.com, example is the second-level domain.
   * Organizations or individuals can **register** second-level domains under available TLDs. The combination of the SLD and the TLD forms a fully qualified domain name (FQDN).
   * In some cases, the SLD can reflect the name of a business, service, or geographic location.
4. **Subdomains (Third-Level Domains and Lower)**:
   * **Subdomains** are any additional levels that exist to the left of the second-level domain.
   * They allow for further structuring of a website or organization, making it easier to manage different sections of a site or different services.
   * For example, in blog.example.com, blog is a subdomain of example.com.
   * Subdomains can be created by the domain owner (e.g., store.example.com, support.example.com) and are often used for distinct services, departments, or regions.
   * Subdomains can be created at any level, so a domain like news.blog.example.com is a **third-level domain** (or subdomain of a subdomain).
5. **Hostnames**:
   * Hostnames are used to identify individual devices or servers within a domain.
   * For example, www.example.com is a **fully qualified domain name (FQDN)** where www is the hostname (which typically refers to a web server), and example.com is the domain name.
   * Other examples of hostnames might include mail.example.com (for a mail server) or ftp.example.com (for an FTP server).

**DNS Record Types**

**A Record**

These records resolve to IPv4 addresses, for example 104.26.10.229

**AAAA Record**

These records resolve to IPv6 addresses, for example 2606:4700:20::681a:be5

**CNAME Record**

These records resolve to another domain name, for example, TryHackMe's online shop has the subdomain name store.tryhackme.com which returns a CNAME record shops.shopify.com. Another DNS request would then be made to shops.shopify.com to work out the IP address.

**MX Record**

These records resolve to the address of the servers that handle the email for the domain you are querying, for example an MX record response for tryhackme.com would look something like alt1.aspmx.l.google.com.

**TXT Record**

TXT records are free text fields where any text-based data can be stored. TXT records have multiple uses, but some common ones can be to list servers that have the authority to send an email on behalf of the domain (this can help in the battle against spam and spoofed email). They can also be used to verify ownership of the domain name when signing up for third party services.

**What happens when you make a DNS request**

1. Local Cache Check: When you request a website, your computer first checks its own cache to see if it has recently looked up the address. If not, it asks your Recursive DNS Server.
2. Recursive DNS Server: This server, often provided by your ISP, checks its cache for the address. If it's not found, the server starts a search by contacting root DNS servers.
3. Root DNS Servers: These servers guide the request to the appropriate Top Level Domain (TLD) server based on the domain extension (like .com).
4. TLD Servers: They direct the request to the specific authoritative DNS server that knows the exact address of the domain.
5. Authoritative DNS Servers: These servers store and provide the final DNS records for the domain. The information is cached locally with a TTL (Time To Live) value to speed up future requests.