DNS Interview Questions & Answers

1. Explain the DNS resolution process step by step.

DNS resolution is the process of converting a human-readable domain name (e.g., www.example.com) into an IP address. The step-by-step process is as follows:

1. **User Query:** The user types a domain name into a web browser.
2. **Browser Cache Check:** The browser first checks its local cache to see if the domain's IP address is already stored.
3. **Operating System Cache Check:** If the browser cache does not contain the IP address, the query is sent to the operating system's DNS resolver.
4. **Local DNS Resolver Query:** If the OS cache does not have the record, the query is sent to the configured DNS resolver, typically provided by the ISP.
5. **Root DNS Servers:** If the resolver does not have the IP cached, it queries one of the 13 root DNS servers.
6. **TLD (Top-Level Domain) Server Query:** The root server directs the query to the TLD server (e.g., .com, .org, .net servers).
7. **Authoritative DNS Server Query:** The TLD server provides the IP address of the authoritative DNS server for the domain.
8. **IP Address Retrieval:** The authoritative DNS server responds with the correct IP address.
9. **Response Caching:** The resolver caches the result to speed up future queries.
10. **Website Access:** The browser uses the IP address to establish a connection with the web server and loads the website.

---

2. What is the difference between authoritative and recursive DNS servers?

- **Recursive DNS Server:**
    - Acts as an intermediary between the user and other DNS servers.
    - Performs the full DNS lookup process on behalf of the client.
    - Typically provided by ISPs or third-party DNS providers (e.g., Google Public DNS, Cloudflare DNS).
- **Authoritative DNS Server:**
    - Stores actual DNS records (A, CNAME, MX, etc.) for a domain.
    - Provides the final answer in the DNS resolution process.
    - Managed by domain owners or hosting providers.

Recursive servers fetch data, while authoritative servers store and provide the definitive DNS records.

---

3. How does DNS caching improve performance? Where does it occur?

DNS caching reduces the time required to resolve domain names by storing previously retrieved DNS query results. It improves performance by:

- **Reducing latency:** Eliminates the need to repeat DNS queries for frequently accessed websites.
- **Lowering DNS server load:** Fewer queries reduce traffic to upstream DNS servers.
- **Enhancing user experience:** Websites load faster as queries are resolved locally.

Where does DNS caching occur?

- **Browser Cache:** Stores DNS responses for recently visited websites.
- **Operating System Cache:** Maintains a local DNS cache for frequently accessed domains.
- **Recursive Resolver Cache:** DNS servers (e.g., ISP resolvers) cache responses to serve multiple users efficiently.

---

4. What is TTL in DNS, and why is it important?

- **TTL (Time-To-Live)** is a setting in DNS records that defines how long a record is valid before it must be refreshed.
- **Importance of TTL:**
    - A **short TTL (e.g., 60 seconds)** ensures frequent updates but increases query load.
    - A **long TTL (e.g., 24 hours)** reduces query frequency but delays propagation of changes.
    - TTL balances performance and accuracy in DNS resolution.

---

5. How does DNS-based load balancing work in large-scale systems?

DNS-based load balancing distributes traffic across multiple servers by returning different IP addresses for the same domain.

- **Round-Robin DNS:** Each DNS query gets a different server IP in a rotating order.
- **Geolocation-based Routing:** Directs users to the closest data center to minimize latency.
- **Failover DNS:** If a server fails, the DNS resolver removes it from the list and directs traffic to healthy servers.
- **Anycast DNS:** Uses the same IP address across multiple locations, routing requests to the nearest server.

This approach improves reliability, availability, and performance in large-scale applications.

---

6. What are common DNS-related security threats, and how can they be mitigated?

- **DNS Spoofing / Cache Poisoning:** Attackers inject false DNS records, redirecting users to malicious websites.
    - **Mitigation:** Use DNSSEC (Domain Name System Security Extensions) to verify DNS records.
- **DDoS Attacks on DNS Servers:** Attackers overload DNS servers with traffic, making domains unreachable.
    - **Mitigation:** Implement rate limiting, Anycast DNS, and load balancing.
- **Man-in-the-Middle Attacks:** Intercept DNS queries to manipulate responses.
    - **Mitigation:** Use encrypted DNS protocols like DNS-over-HTTPS (DoH) or DNS-over-TLS (DoT).
- **NXDOMAIN Attack:** Attackers flood DNS resolvers with queries for non-existent domains, exhausting server resources.
    - **Mitigation:** Deploy response rate limiting (RRL) and DNS firewalls.

Securing DNS is critical to ensuring safe and reliable internet communication.