Del johnver C. Pascual

BSIT 3WMAD

1. **Think of an analogy of a client/server system. Illustrate it and justify.**

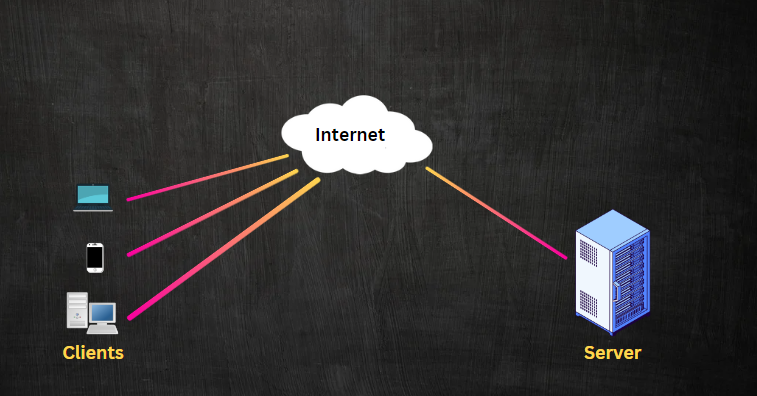
An example of a client/server system is a restaurant where the customer (client) puts an order (request) with the server (server), who processes it in the kitchen (database) and delivers the meal (response). Clients rely on servers to manage inquiries and deliver information, just as diners rely on staff to understand their preferences and meet their needs. Similar to how servers store and retrieve information for numerous clients at once, the kitchen effectively handles ingredients and meals to ensure timely service, guaranteeing a seamless and enjoyable experience for all.

**2.**

A web server responds to requests from clients, which are usually web browsers, and sends the resources—such as files or web pages—back to the client. When a user inputs a URL or clicks a link, the browser sends an HTTP request to the web server, starting this interaction. After that, the server finds the requested resource in its storage to fulfill the request. The resource is sent back to the browser immediately by the server if it is a static file (such as an HTML page). However, if the request involves dynamic content (e.g., data from a database), the server might interact with backend services, such as databases or server-side scripts, to generate the content before sending it.

Following a successful processing attempt, the server replies to the client via HTTP with the requested data and a status code (200 for success, 404 for "not found"). In order to provide a secure connection between the client and server, web servers frequently encrypt data using Secure HTTP (HTTPS). Web servers oversee the sharing of web content in this way, guaranteeing that consumers may get the information they require online.

A web server interacts with clients, like web browsers. When a user enters a URL, the browser sends an HTTP request to the server. The server processes the request by finding the webpage or file. After processing, the server sends an HTTP response back to the browser. This response contains the requested webpage. The process happens quickly, allowing users to view websites.



**3. Three Services that a Web Server Can Provide**

1.**Hosting Websites** - A web server hosts websites by storing all necessary files like HTML, CSS, and JavaScript. When users request a website, the server delivers the content to their browser. This allows users to view and interact with the website.

2.**Handling APIs** - Web servers manage APIs that allow different applications to exchange data. When an application sends a request for data, the server processes it and sends back the requested information in formats like JSON or XML.

3.**Serving Dynamic Content** - A web server can serve dynamic content by generating web pages based on user inputs or interactions. For example, it retrieves data from a database and customizes the response, such as showing a user-specific dashboard.

The web server uses server-side programs like PHP, Python, or Node.js and data collecting from databases to create content dynamically when a user requests a page (such as a product page or personalized dashboard).

**3.**

**A. How does DDOS work?**

The goal of a distributed denial of service (DDoS) assault is to overload a target network, server, or service by sending an overwhelming quantity of traffic from several different sources at once. Due to the excessive traffic volume, the server's bandwidth, CPU, and memory are depleted, which causes the service to function slowly or not at all for authorized users.

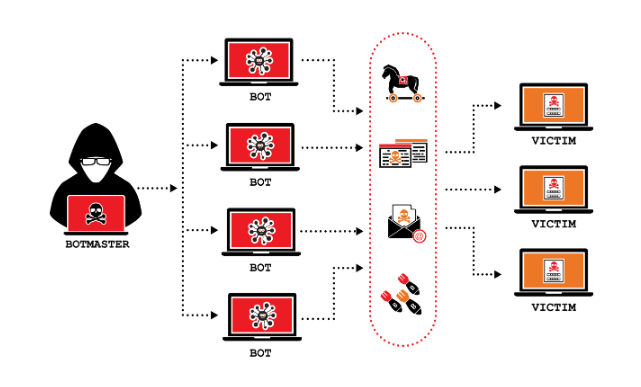
Usually, the assault consists of a number of compromised computers (also known as botnets) cooperating to send a large number of requests to the target server at once. A DDoS attack aims to interfere with the service, either temporarily or permanently, rather than steal data.

**B. DDoS in relation to the client/server architecture**

In the client/server architecture, clients (such as web browsers or apps) send requests to a server, and the server processes these requests and sends back the appropriate response. In a typical scenario, the server handles requests at a manageable rate, allowing it to serve clients efficiently.

In a DDoS attack, malicious clients (botnets) overwhelm the server by sending an abnormal number of requests. Since the server can only process a finite number of requests at a time, the excess traffic causes the server to slow down or crash. As a result, legitimate clients are unable to get their requests processed, and the service becomes unavailable or unreliable.

**C. Illustrate the concept.**



**D. Provide at least two real-life scenarios of DDOS attacks to web servers.**

**GitHub DDoS Attack (2018)** In February 2018, GitHub, one of the largest software development platforms, experienced the largest DDoS attack at the time, peaking at 1.35 Tbps (terabits per second). The attack utilized a method called memcached reflection, where attackers exploited a vulnerability in the memcached servers to amplify the attack traffic. The overwhelming volume of requests caused GitHub to go offline for about 10 minutes before they mitigated the attack using automated protection services.

**Dyn DNS Attack (2016)** In October 2016, a DDoS attack on Dyn, a major DNS provider, disrupted several popular websites, including Twitter, Netflix, Reddit, and Spotify. The attack involved Mirai botnets, which were made up of IoT (Internet of Things) devices like webcams and smart appliances that were compromised and used to send massive amounts of traffic to Dyn’s DNS servers. The attack affected internet services across the U.S. and parts of Europe for several hours.