RAW SOCKET PROGRAMMING

APARNAA MAHALAXMI ARULLJOTHI (A20560995)

How your client and server communicate over raw sockets

The client and server communicate over raw sockets using manually crafted TCP packets

Client side:

- The client creates a raw socket using **socket(AF_INET, SOCK_RAW, IPPROTO_TCP).** This enables sending TCP packets directly at the transport layer, bypassing higher-level abstractions.
- The client constructs custom TCP packets using the **iphdr (IP header)** and **tcphdr (TCP header)** structures. The packet contains essential fields such as source IP, destination IP, source port, destination port, sequence number, and TCP flags like SYN.
- The client sends these crafted packets to the server's IP and port using **sendto().** This is used for both normal traffic simulation and attack simulation (e.g., SYN flood).

Server side:

- The server creates a raw socket using **socket(AF_INET, SOCK_RAW, IPPROTO_TCP)** to receive raw TCP packets.
- The server uses **recvfrom()** to capture incoming packets from the network.
- The captured packets are parsed to extract the **iphdr** and **tcphdr** headers for analysis.
- The server checks the TCP flags in the received packet.
- If the packet has the SYN flag set and the ACK flag unset, it indicates a connection initiation request.
- The server tracks such packets for SYN flood detection.
- The server also monitors the destination port and source IP to detect patterns indicative of port scanning.

CHALLENGES FACED:

I ran into two permission issues while working on the project.

- First, raw sockets required administrative privileges, so I had to run the client and server programs with **sudo** to get them working.
- Second, I faced problems using **gcc** for compiling the code, which I fixed by adding the **--privileged flag** in the Docker setup to give the container the required permissions.