CCP

Project

Presentation

PROJECT TITLE :

"Port Inspector"

Course Title : Programming Fundamentals

Team Members :

- Rushda Abbasi CT-25074
- Ramla Kamran CT-25051

Section : B

Course Instructor : Sir Abdullah

Institute : NED University

Introduction

- A port inspector/scanner is a tool that checks which ports on a target system are open or closed.
- Used by network administrators for security testing.
- Helps identify which services are running on a host.

Goal of our Project

- To develop a simple port scanner using the C programming language.
- To understand socket programming and network communication basics.
- To analyze which ports on a host are open or closed.

Working Principle

This program determines if a port is OPEN by attempting to complete a TCP three-way handshake using the connect() function.
Core Mechanism:

1. Setup: A TCP socket is created and configured with the target IP and Port.
2. Test: The connect() function tries to initiate a connection.
3. Result:
    - Success (Returns 0): The handshake completes.
    - Failure (Returns -1):
        - The target sends an RST (Reset). The port is CLOSED (no service is listening).
        - The attempt times out (no response). The port is FILTERED (a firewall is blocking it).

Tools and Technologies

- Include:
  - Programming Language: C
  - Libraries Used:
    - <stdio.h> for input/output
    - <string.h> for string handling
    - <arpa/inet.h> for internet functions
    - <unistd.h> for close()
  - IDE : VS Code
  - Platform: Linux

Algorithm

Start
↓
Input IP and port range
↓
For each port:
  Try to connect
  ↓
  If success → Port open
  Else → Port closed
↓
Display results
↓
End

Code Explanation

Key Steps and Functions:

- Input: Gathers the target IP address and port number from the user.
- socket(): Creates a network socket (AF_INET for IPv4, SOCK_STREAM for TCP) to serve as the communication endpoint.
- struct sockaddr_in: Configures the target's address, using htons() to ensure the port number is in correct network byte order, and inet_addr() to convert the IP string.
- connect(): Attempts to initiate a TCP connection to the target.
  - Return value 0: Connection successful. The Port is OPEN and listening.
  - Return value -1: Connection failed. The Port is CLOSED or FILTERED (blocked by a firewall).
- close(): Closes the socket to release system resources.

In summary, it's a minimal client program that uses the success or failure of a connect() call to determine a port's status.

Conclusion

This simple port scanner demonstrates how basic network communication works using socket programming in C. It allows the user to input an IP address and a port number, then attempts to connect to that port to determine whether it is open or closed. Through this project, we learn the practical use of sockets, IP addressing, and the TCP connection process. It also highlights the importance of network security and how open ports can expose vulnerabilities in a system. Overall, the project strengthens understanding of low-level networking concepts, C programming, and system-level communication between devices.

Thank You