

# Secure Remote Command Executor Project

## Slide 1: Project Introduction

Project Title: Secure Remote Command Executor over TCP

Language: C++ on Linux

Goal: Allow remote clients to connect to a server, authenticate, and execute commands securely.

Core Concepts: Socket programming, threading, process execution, and basic security.

## Slide 2: What is Socket Programming?

A method for two systems to communicate over a network.

Types: TCP (reliable), UDP (faster but less reliable)

Used to send and receive data between client and server.

## Slide 3: Why TCP Sockets?

Reliable and connection-based.

Guarantees that all data arrives in order.

Essential for command execution and response handling.

## Slide 4: What is the Role of Client and Server?

Client: Connects to server, authenticates, sends commands, receives results.

Server: Listens for clients, authenticates them, executes commands, and sends outputs.

## Slide 5: What is popen()?

Used to execute Linux shell commands from C++.

Allows capturing the output of the executed command.

Alternative to `system()` with output capture.

## Slide 6: Why Authentication?

Ensures only authorized users can run commands.

In our case, done using simple password.

Can be upgraded to certificate-based SSL.

# Secure Remote Command Executor Project

## Slide 7: What is Multi-threading?

Allows server to handle multiple clients at the same time.

Each client runs in a separate thread.

Implemented using pthread library in C++.

## Slide 8: What is IPC (Inter-process Communication)?

Enables communication between different processes.

TCP sockets act as IPC channels between client and server.

popen() also acts as IPC between server and shell.

## Slide 9: Linux OS Concepts Demonstrated

Memory Management: buffers and dynamic memory

Processes: popen creates child processes

Scheduling: OS schedules threads and processes

Virtual Memory: all programs use it by default

Multi-threading: pthreads

IPC: sockets

File System: accessed via command execution

## Slide 10: Common Viva Questions & Answers

Q: Why not UDP?

A: Its unreliable; TCP ensures complete command execution.

Q: What happens on wrong password?

A: Connection is closed with an error message.

Q: How can this be extended?

A: Add SSL, GUI, or file transfer support.

# Secure Remote Command Executor Project