UE23CS252B: COMPUTER NETWORKS

Mini Project 2

ABHISHEK P **PES2UG23AM002**
DEVAKI PRASAD **PES2UG23AM029**
CSE(AIML)-4A

File Transfer System using raw sockets

Client Program:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <openssl/ssl.h>
#include <openssl/err.h>

#define SERVER_IP "192.168.64.3"
#define PORT 9090
#define BUF_SIZE 1024
#define FILENAME "file.txt"

int main() {
    SSL_library_init();
    OpenSSL_add_all_algorithms();
    SSL_load_error_strings();

    const SSL_METHOD *method = TLS_client_method();
    SSL_CTX *ctx = SSL_CTX_new(method);

    SSL *ssl;
    int sock = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in addr = {
        .sin_family = AF_INET,
        .sin_port = htons(PORT)
    };
    inet_pton(AF_INET, SERVER_IP, &addr.sin_addr);
    connect(sock, (struct sockaddr*)&addr, sizeof(addr));

    ssl = SSL_new(ctx);
    SSL_set_fd(ssl, sock);
    if (SSL_connect(ssl) <= 0) {
        ERR_print_errors_fp(stderr);
        return 1;
    }

    FILE *fp = fopen(FILENAME, "rb");
```

```
    if (!fp) {
        perror("File open failed");
        return 1;
    }

    char buffer[BUF_SIZE];
    int bytes;
    while ((bytes = fread(buffer, 1, BUF_SIZE, fp)) > 0) {
        SSL_write(ssl, buffer, bytes);
    }

    fclose(fp);
    SSL_shutdown(ssl);
    SSL_free(ssl);
    close(sock);
    SSL_CTX_free(ctx);
    printf("📤 File sent successfully.\n");
    return 0;
}
```

Server Program:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <openssl/ssl.h>
#include <openssl/err.h>

#define PORT 9090
#define BUF_SIZE 1024

void *handle_client(void *arg) {
    SSL *ssl = (SSL *)arg;
    FILE *fp = fopen("received.txt", "wb");
    if (!fp) {
        perror("File open failed");
        SSL_shutdown(ssl);
        SSL_free(ssl);
        pthread_exit(NULL);
    }

    char buffer[BUF_SIZE];
    int bytes;
    while ((bytes = SSL_read(ssl, buffer, BUF_SIZE)) > 0) {
        fwrite(buffer, 1, bytes, fp);
    }

    printf("✅ File received from client\n");
    fclose(fp);
    SSL_shutdown(ssl);
    SSL_free(ssl);
    pthread_exit(NULL);
}
```

```
int main() {
    SSL_library_init();
    OpenSSL_add_all_algorithms();
    SSL_load_error_strings();

    const SSL_METHOD *method = TLS_server_method();
    SSL_CTX *ctx = SSL_CTX_new(method);
    SSL_CTX_use_certificate_file(ctx, "cert.pem", SSL_FILETYPE_PEM);
    SSL_CTX_use_PrivateKey_file(ctx, "key.pem", SSL_FILETYPE_PEM);

    int server_fd = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in addr = {
        .sin_family = AF_INET,
        .sin_port = htons(PORT),
        .sin_addr.s_addr = INADDR_ANY
    };

    bind(server_fd, (struct sockaddr*)&addr, sizeof(addr));
    listen(server_fd, 5);
    printf("📡 Server waiting on port %d...\n", PORT);

    while (1) {
        int client_fd = accept(server_fd, NULL, NULL);
        SSL *ssl = SSL_new(ctx);
        SSL_set_fd(ssl, client_fd);

        if (SSL_accept(ssl) <= 0) {
            ERR_print_errors_fp(stderr);
            close(client_fd);
            continue;
        }

        pthread_t tid;
        pthread_create(&tid, NULL, handle_client, ssl);
        pthread_detach(tid);
    }

    close(server_fd);
    SSL_CTX_free(ctx);
    return 0;
}
```

Screenshots:


```
Last login: Thu Apr 24 17:55:07 on ttys000
abhishekp@Abhisheks-MacBook-Air-3 ~ % cd downloads
abhishekp@Abhisheks-MacBook-Air-3 downloads % nano sender.c
abhishekp@Abhisheks-MacBook-Air-3 downloads % gcc sender.c -o sender -I/opt/homebrew/opt/openssl/include -L/opt/homebrew/opt/openssl/lib -lssl -lcrypto
abhishekp@Abhisheks-MacBook-Air-3 downloads % ./sender
📤 File sent successfully.
abhishekp@Abhisheks-MacBook-Air-3 downloads %
```