

Q) Consider a bank server to maintain user credentials:-

AIM:-

To create a bank server and a client and encrypting the user credential by the bank server and sending it to the client.

ALGORITHM:-

- On the server side :-
 - Create A socket (with a port no.)
 - Bind the server to the socket
 - ~~Connect the server to socket and wait for connecting~~
 - After binding, change the state of the server in listen mode.
 - And allow the server the ~~server~~ accept queries.
 - Once the receives the query from the client the server should encrypt user data and sent it back to the user/client (The encrypted data)

- Once received close the socket
- On the client side:-
 - Create a Socket
 - Connect the Socket to the server's port no. port number
 - After connecting, send the client's credentials acc. to the server.
 - Later, receive the encrypted user credentials.
 - Once data received, close the socket.

CODE ON THE CLIENT 'S SIDE

```
#include<stdio.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
// Client Side
int main()
{
    int soc, port, numsend, len, len1;
    char username1[100], password1[100], username[100], password[100];
    struct sockaddr_in server, client;
    printf("Enter the length of your username and password\n");
    scanf("%d%d", &len, &len1);
    printf("Enter the username and password\n");
    scanf("%s%s", username, password);
    printf("Enter the port no.\n");
    scanf("%d", &port);
    soc=socket(AF_INET, SOCK_STREAM, 0);
    if(soc<0)
        printf("Socket not created\n");
    else
        printf("Socket Created\n");
    server.sin_family=AF_INET;
    server.sin_addr.s_addr=htonl(INADDR_ANY);
    server.sin_port=htons(port);

    if(connect(soc,(struct sockaddr*)&server,sizeof(server))<0)
        printf("Can't connect\n");
    else
        printf("Connected\n");

    send(soc,&username,sizeof(username),0);
```

```

        send(soc,&len,sizeof(len),0);
        send(soc,&password,sizeof(password),0);
        send(soc,&len1,sizeof(len1),0);
        recv(soc,username1,sizeof(username1),0);
        recv(soc,password1,sizeof(password1),0);
        printf("Encrypted username and password\n%s\
n%s",username1,password1);
        return 0;
}

```

CODE ON THE SERVER'S SIDE

```

#include<stdio.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<string.h>
// Server Side
void reverse(char arr[], int start, int end)
{
    int temp;
    if (start >= end)
        return;
    temp = arr[start];
    arr[start] = arr[end];
    arr[end] = temp;
    reverse(arr, start+1, end-1);
}

void encyrpt(char username[],char password[],int len,int len1){
    char t;
    reverse(username,0,len-1);

```

```

        username[len]='1';username[len+1]='2';username[len+2]='3';
        printf("\n%s\n",username);
        for(int j=len1;j>=3;--j){
            password[j]=password[j-1];
        }password[3]='@';
        for(int j=len1+1;j>=1;--j){
            password[j]=password[j-1];
        }password[1]='!';
        for(int i=0;i<len1+2;++i) printf("%c",password[i]);
    }

int main()
{
    int s,b,sport,key,len,len1=0;
    char username[100],password[100];
    struct sockaddr_in saddr,caddr;
    printf("Enter any Desired port number\n");
    scanf("%d",&sport);
    int clen=sizeof(caddr);
    s=socket(AF_INET,SOCK_STREAM,0);
    if(s<0)
    {
        printf("Error while socket creation\n");
    }

    else
        printf("Socket created successfully\n");
    saddr.sin_family=AF_INET;
    saddr.sin_addr.s_addr=htonl(INADDR_ANY);
    saddr.sin_port=htons(sport);
    b=bind(s,(struct sockaddr*)&saddr,sizeof(saddr));
    if(b==0)
        printf("Interface binded to the socket\n");
    else

```

```
        printf("Interface not binded to the socket\n");
        listen(s,5);
        key=accept(s,(struct sockaddr*)&caddr,&clen);
        if(key<0)
            printf("Error\n");
        else
            printf("Well Connected\n");
        recv(key,&username,sizeof(username),0);
        recv(key,&len,sizeof(len),0);
        recv(key,&password,sizeof(password),0);
        recv(key,&len1,sizeof(len1),0);
        printf("The received username and password\n
%s\n%s\n",username,password);
        encyrpt(username,password,len,len1);
        send(key,username,sizeof(username),0);
        send(key,password,sizeof(password),0);
    }
```

OUTPUTS

ON THE SERVER SIDE



```
ghost@kali:~/Desktop/C-C++  
File Actions Edit View Help Second.c x C Rough.c  
C Second.c Command  
└─(ghost㉿kali)-[~]ude<stdio.h>  
$ cd "/home/ghost/Desktop/C-C++/" && gcc Second.c -o S  
econd && "/home/ghost/Desktop/C-C++/"Second  
Enter any Desired port number  
10000  
Socket created successfully  
Interface binded to the socket  
Well Connected  
The received username and password  
ashutosh  
ashutosh  
8 {  
9 int temp;  
hsotuhsa123  
a!sh@utosh  
11 }--> return;  
└─(ghost㉿kali)-[~/Desktop/C-C++]  
$ ┌ 13 arr[start] = arr[end];  
14 arr[end] = temp;  
15 reverse(arr, start+1, end-1);  
16 }  
17  
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
```

ON THE CLIENT SIDE

```
ghost@kali:~/Desktop/C-C++  
File Actions Edit View Help  
└─(ghost㉿kali)-[~]  
$ cd "/home/ghost/Desktop/C-C++/" && gcc Rough.c -o R  
ough && "/home/ghost/Desktop/C-C++/"Rough  
Enter the length of your username and password  
8  
8  
Enter the username and password  
ashutosh  
ashutosh  
Enter the port no.  
10000  
Socket Created  
Connected  
Encrypted username and password  
hsotuhsa123  
a!sh@utosh  
└─(ghost㉿kali)-[~/Desktop/C-C++]  
$ ┌
```

PACKET TRACER PART













