

Exp-09

Remote Command Execution

Aaryan Singhal (RA1911003010257)

Server Code:-

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <stdlib.h>
#include <netdb.h>
#include <netinet/in.h>
#include <string.h>
#include <sys/stat.h>
#include <arpa/inet.h>
#include <unistd.h>
#define MAX 1000
int main()
{
    int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
    int size;
    char buffer[MAX], message[] = "Command Successfully executed !";
    struct sockaddr_in clientAddress, serverAddress;
    socklen_t clientLength = sizeof(clientAddress);
    bzero(&serverAddress, sizeof(serverAddress));
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
    serverAddress.sin_port = htons(9976);
    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
    while (1)
    {
        bzero(buffer, sizeof(buffer));
        recvfrom(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr
        *)&clientAddress,
        &clientLength);
        system(buffer);
        printf("Command Executed ... %s ", buffer);
        sendto(serverDescriptor, message, sizeof(message), 0, (struct sockaddr
        *)&clientAddress,
        clientLength);
    }
    close(serverDescriptor);
    return 0;
}
```

Client Code:-

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <unistd.h>
#include <netdb.h>
#include <netinet/in.h>
#include <string.h>
#include <arpa/inet.h>
#define MAX 1000
int main()
{
    int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
    char buffer[MAX], message[MAX];
    struct sockaddr_in cliaddr, serverAddress;
    socklen_t serverLength = sizeof(serverAddress);
    bzero(&serverAddress, sizeof(serverAddress));
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = inet_addr("127.0.0.1");
    serverAddress.sin_port = htons(9976);
    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
    while (1)
    {
        printf("\nCOMMAND FOR EXECUTION ... ");
        fgets(buffer, sizeof(buffer), stdin);
        sendto(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&serverAddress,
serverLength);
        printf("\nData Sent !");
        recvfrom(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&serverAddress,
&serverLength);
        printf("UDP SERVER : %s", message);
    }
    return 0;
```

}

Screenshots:-

```
Server.c
1 #include <arpa/inet.h>
2 #include <unistd.h>
3 #define MAX 1000
4
5 int main()
6 {
7     int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
8     int size;
9     char buffer[MAX], message[] = "Command Successfully executed !";
10    struct sockaddr_in clientAddress, serverAddress;
11    socklen_t clientLength = sizeof(clientAddress);
12
13    bzero(serverAddress, sizeof(serverAddress));
14    serverAddress.sin_family = AF_INET;
15    serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
16    serverAddress.sin_port = htons(8080);
17
18    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
19    while (1)
20    {
21        bzero(buffer, sizeof(buffer));
22        recvfrom(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&clientAddress, &clientLength);
23        system(buffer);
24        printf("Command Executed ... %s", buffer);
25        sendto(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&clientAddress, clientLength);
26    }
27    close(serverDescriptor);
28    return 0;
29 }
```

```
Client.c
1 #include <arpa/inet.h>
2 #include <unistd.h>
3 #define MAX 1000
4
5 int main()
6 {
7     int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
8     int size;
9     char buffer[MAX], message[] = "Command Successfully executed !";
10    struct sockaddr_in clientAddress, serverAddress;
11    socklen_t clientLength = sizeof(clientAddress);
12
13    bzero(serverAddress, sizeof(serverAddress));
14    serverAddress.sin_family = AF_INET;
15    serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
16    serverAddress.sin_port = htons(8080);
17
18    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
19    while (1)
20    {
21        bzero(buffer, sizeof(buffer));
22        recvfrom(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&clientAddress, &clientLength);
23        system(buffer);
24        printf("Command Executed ... %s", buffer);
25        sendto(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&clientAddress, clientLength);
26    }
27    close(serverDescriptor);
28    return 0;
29 }
```

```
RA1911003010265:~/environment $ cd RA257_REU
RA1911003010265:~/environment/RA257_REU $ gcc Client.c
RA1911003010265:~/environment/RA257_REU $ ./a.out
COMMAND FOR EXECUTION ... aaryan
Data Sent !UDP SERVER : aaryan
COMMAND FOR EXECUTION ...
```

4

```
Server.c
1 #include <arpa/inet.h>
2 #include <unistd.h>
3 #define MAX 1000
4
5 int main()
6 {
7     int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
8     int size;
9     char buffer[MAX], message[] = "Command Successfully executed !";
10    struct sockaddr_in clientAddress, serverAddress;
11    socklen_t clientLength = sizeof(clientAddress);
12
13    bzero(serverAddress, sizeof(serverAddress));
14    serverAddress.sin_family = AF_INET;
15    serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
16    serverAddress.sin_port = htons(8080);
17
18    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
19    while (1)
20    {
21        bzero(buffer, sizeof(buffer));
22        recvfrom(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&clientAddress, &clientLength);
23        system(buffer);
24        printf("Command Executed ... %s", buffer);
25        sendto(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&clientAddress, clientLength);
26    }
27    close(serverDescriptor);
28    return 0;
29 }
```

```
Client.c
1 #include <arpa/inet.h>
2 #include <unistd.h>
3 #define MAX 1000
4
5 int main()
6 {
7     int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
8     int size;
9     char buffer[MAX], message[] = "Command Successfully executed !";
10    struct sockaddr_in clientAddress, serverAddress;
11    socklen_t clientLength = sizeof(clientAddress);
12
13    bzero(serverAddress, sizeof(serverAddress));
14    serverAddress.sin_family = AF_INET;
15    serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
16    serverAddress.sin_port = htons(8080);
17
18    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
19    while (1)
20    {
21        bzero(buffer, sizeof(buffer));
22        recvfrom(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&clientAddress, &clientLength);
23        system(buffer);
24        printf("Command Executed ... %s", buffer);
25        sendto(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&clientAddress, clientLength);
26    }
27    close(serverDescriptor);
28    return 0;
29 }
```

```
RA1911003010265:~/environment $ cd RA257_REU
RA1911003010265:~/environment/RA257_REU $ gcc Client.c
RA1911003010265:~/environment/RA257_REU $ ./a.out
COMMAND FOR EXECUTION ... aaryan
Data Sent !UDP SERVER : Command Successfully executed !
COMMAND FOR EXECUTION ...
```

```
. /a.out - "ip-172-31-2-69" x  . /a.out - "ip-172-31-2-69" x +
RA1911003010265:~/environment $ cd RA257_REU
RA1911003010265:~/environment/RA257_REU $ gcc Client.c
RA1911003010265:~/environment/RA257_REU $ ./a.out

COMMAND FOR EXECUTION ... aaryan

Data Sent !UDP SERVER : Command Successfully executed !
COMMAND FOR EXECUTION ... █
```

Task View File Explorer Visual Studio Code Firefox Google Chrome Microsoft Edge ? 31°C Mostly