COMPUTER COMMUNICATION NETWORKS

LAB EXPERIMENT 11

NAME: RAHIL SHARMA
PRN: 18070123062
BATCH: 2018-2022
DIVISION: G2; EA 3

Aim: Port Scan. Write a client program which finds out which ports are responded by a remote server. (program to list host names from command line, attempt to open socket to each one and print the remote host, the remote port, the local address and the local port.)

Theory: A port scanner is an application designed to probe a server or host for open ports. Such an application may be used by administrators to verify security policies of their networks and by attackers to identify network services running on a host and exploit vulnerabilities. A port scan or port scan is a process that sends client requests to a range of server port addresses on a host, with the goal of finding an active port; this is not a nefarious process in and of itself. The majority of uses of a port scan are not attacks, but rather simple probes to determine services available on a remote machine. The port scanner function does all the work. It first converts the port numbers from text string to numerical using function atoi (ascii to integer). It then goes into a loop, checking each port one by one. First it creates a socket, opens a connection on the IP address and

the port and reports if the connection on that port is successful or failed. Finally it closes the socket and connection.

Procedure:

1. Run your program. It will ask IP address.
2. Enter the above found IP address (3.6.152.226).
3. It will ask lower port. Enter 20 .
4. It will ask higher port. Enter 25 .
5. The program will check ports 20 to 25 and report which ports are open on this web site. You will get only 21 open and rest all failed.
6. Run the program again and check ports 80 to 85. You will fine port 80 open (HTTP) and rest all failed. Similarly you can try other web sites (google.com, yahoo.com etc). Do not give too big a range of ports. Just 4 or 5 ports in the range.
7. The program may go into waiting loop trying to check ports which are non-existent on the server.
8. Normally all web sites will have port 80 open. Upload program editor screen shot and program output screen shot.

Code of the two programs:

Code 1:

```
import socket

def port_scan():
   link = input("Enter link to perform scan ports on:")
   host = socket.gethostbyname(link)
   res = "a"

   while (res != "bye"):
      min = input("Enter the lowest limit of range")
      max = input("Enter the highest limit of range")
      for port in range (int(min), int(max)):
         try:
            client_socket = socket.socket()
            print("Trying connection to", host,"on port",port,".....")
            if client_socket.connect_ex((host, port))==0:
```

```
            print("Connection to", host, "on port", port, "was Successful");
          else:
            print("Connection to",host,"on port",port,"was A Failure");
          port = port + 1;
          client_socket.close()
        except socket.error:
          print("Connection to",host,"on port",port,"was A Failure");
          port = port + 1;
          client_socket.close()

    res = input("Scan ended, enter 'bye' to exit, or any other input to search in different
range:")
  print("Scanner Exited")

if _name_ == '_main_':
  port_scan()
```

Code 2:

```
import socket
t_host=str(input("Enterthehosttobescanned:"))
t_ip=socket.gethostbyname(t_host)
print(t_ip)
i = 0
for i in range(20,86):
  t_port = i
  try:
    sock=socket.socket()
    res=sock.connect((t_ip,t_port))
    print("Port{}:OPEN!!".format(t_port))
    sock.close()
  except:
    print("Port{}:Closed".format(t_port))

print("Port Scanning complete")
```

Outputs and Screenshots:

For Code 1:

In [*]:

```
import socket

def port_scan():
    link = input("Enter link to perform scan ports on:")
    host = socket.gethostbyname(link)
    res = "a"

    while (res != "bye"):
        min = input("Enter the lowest limit of range")
        max = input("Enter the highest limit of range")
        for port in range (int(min), int(max)):
            try:
                client_socket = socket.socket()
                print("Trying connection to", host,"on port",port,".......")
                if client_socket.connect_ex((host, port))==0:
                    print("Connection to", host, "on port", port, "was Successful");
                else:
                    print("Connection to",host,"on port",port,"was A Failure");
                port = port + 1;
                client_socket.close()
            except socket.error:
                print("Connection to",host,"on port",port,"was A Failure");
                port = port + 1;
                client_socket.close()

        res = input("Scan ended, enter 'bye' to exit, or any other input to search in different range:")
    print("Scanner Exited")

port_scan()
```

```
Enter link to perform scan ports on:www.google.com
Enter the lowest limit of range20
Enter the highest limit of range25
Trying connection to 172.217.167.36 on port 20 .......
Connection to 172.217.167.36 on port 20 was A Failure
Trying connection to 172.217.167.36 on port 21 .......
Connection to 172.217.167.36 on port 21 was A Failure
Trying connection to 172.217.167.36 on port 22 .......
Connection to 172.217.167.36 on port 22 was A Failure
Trying connection to 172.217.167.36 on port 23 .......
Connection to 172.217.167.36 on port 23 was A Failure
Trying connection to 172.217.167.36 on port 24 .......
Connection to 172.217.167.36 on port 24 was A Failure
Scan ended, enter 'bye' to exit, or any other input to search in different range:y
Enter the lowest limit of range80
Enter the highest limit of range85
Trying connection to 172.217.167.36 on port 80 .......
Connection to 172.217.167.36 on port 80 was Successful
Trying connection to 172.217.167.36 on port 81 .......
Connection to 172.217.167.36 on port 81 was A Failure
Trying connection to 172.217.167.36 on port 82 .......
Connection to 172.217.167.36 on port 82 was A Failure
Trying connection to 172.217.167.36 on port 83 .......
Connection to 172.217.167.36 on port 83 was A Failure
Trying connection to 172.217.167.36 on port 84 .......
Connection to 172.217.167.36 on port 84 was A Failure

Scan ended, enter 'bye' to exit, or any other input to search in different range:
```

In [ ]:

For Code 2:

In [*]:

```
import socket
t_host=str(input("Enterthehosttobescanned:"))
t_ip=socket.gethostbyname(t_host)
print(t_ip)
i = 0
for i in range(20,86):
    t_port = i
    try:
        sock=socket.socket()
        res=sock.connect((t_ip,t_port))
        print("Port{}:OPEN!!".format(t_port))
        sock.close()
    except:
        print("Port{}:Closed".format(t_port))

print("Port Scanning complete")
```

```
Enterthehosttobescanned:www.google.com
142.250.77.132
Port20:Closed
Port21:Closed
Port22:Closed
Port23:Closed
Port24:Closed
Port25:Closed
Port26:Closed
Port27:Closed
Port28:Closed
Port29:Closed
Port30:Closed
Port31:Closed
Port32:Closed
```

```
Port30:Closed
Port31:Closed
Port32:Closed
Port33:Closed
Port34:Closed
Port35:Closed
Port36:Closed
Port37:Closed
Port38:Closed
Port39:Closed
Port40:Closed
Port41:Closed
Port42:Closed
Port43:Closed
Port44:Closed
Port45:Closed
Port46:Closed
Port47:Closed
Port48:Closed
Port49:Closed
Port50:Closed
Port51:Closed
Port52:Closed
Port53:Closed
Port54:Closed
Port55:Closed
Port56:Closed
Port57:Closed
Port58:Closed
Port59:Closed
Port60:Closed
```

```
Port59:Closed
Port60:Closed
Port61:Closed
Port62:Closed
Port63:Closed
Port64:Closed
Port65:Closed
Port66:Closed
Port67:Closed
Port68:Closed
Port69:Closed
Port70:Closed
Port71:Closed
Port72:Closed
Port73:Closed
Port74:Closed
Port75:Closed
Port76:Closed
Port77:Closed
Port78:Closed
Port79:Closed
Port80:OPEN!!
Port81:Closed
Port82:Closed
Port83:Closed
Port84:Closed
Port85:Closed
Port Scanning complete
```

Conclusion: From this experiment we have learnt how to implement port scan in Python and get to know the working different ports in Computer Networks.