21CY681- Internet Protocol lab

Name: MEERA T

Register Number: CB.EN.P2CYS22002

Title: Establish a Client-Client Secure communication

protocol using socket programming.

Date: 26-12-2022

Aim:

To establish client-client secure communication protocol using socket programming.

Client and Server Connection:

This is the python program for server for establishing a connection between client and server.

```
GNU nano 5.9

Deport socket

1 Create a 180/80 reader

Sock = Socket.Socket(Socket.AF_INET, Socket.SOCK_STREAM)

3 Bind Yid socket to the port

Sock = Socket.Socket(Socket.AF_INET, Socket.SOCK_STREAM)

3 Bind Yid socket to the port

Sock.Dind(Sers) = ('localhost', 10000)

print('starting up on {} port {}'.format(*server_address))

Sock.Dind(Server_address)

2 Linear for incoming commercions

Sock.Dind(Server_address)

2 Linear for incoming commercions

Sock.Dind(Server_address)

2 Linear for incoming commercions

print('sint for a connection

print('sint for a connection')

connection, client_address - sock.accept()

try:

print('connection from', client_address)

**Precise sussages from the client and print then

while True:

nessage:
    print('received message', message)

else:
    print('received message', message)
```

This is the python program for client for establishing a connection between client and server.

This is the scapy captured pcap which is being shown in wireshark.

Client and Server connection using RSA Encryption key:

This shows a python program for the server where RSA encryption key is being used to make the message encrypted.

This shows a python program for the client where RSA encryption key is being used to make the message encrypted.

Now we are using Scapy which is a tool used here to capture the traffic between the client and server.

```
>>> capture = sniff(iface="lo", count= )

^C>>> capture.summary()

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin S

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin S

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 SA

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 SA

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 SA

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin A

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin A

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin PA / Raw

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin PA / Raw

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 A

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 A

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 PA / Raw

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 PA / Raw

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 PA / Raw

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 PA / Raw

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 PA / Raw

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin PA / Raw

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin PA / Raw

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin PA / Raw

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 A

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin PA / Raw

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:webmin PA / Raw

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:39354 A

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:39354 A

Ether / IP / TCP 127.0.0.1:39354 > 127.0.0.1:39354 A

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 A

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 A

Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 A

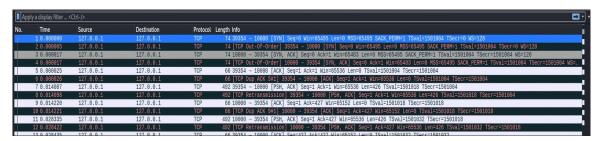
Ether / IP / TCP 127.0.0.1:webmin > 127.0.0.1:39354 A
```

When the client program and server program is running in parallel, the connection is being is made and the message is being sent.

```
(kali@ kali)-[~/Desktop/IP]
$ python3 client.py
connecting to localhost port 10000
Enter a message to send to the server (enter 'q' to quit): hi
Enter a message to send to the server (enter 'q' to quit): hello
Enter a message to send to the server (enter 'q' to quit):
```

When the server is connected, it receives messages from client.

The scapy captured file is being saved .The same pcap file is opened in Wireshark.



When we open a file we get the data.