4. Networking Protocols and Security 30 Marks
IPv6

1. Define IPv6 and explain why it was developed to replace IPv4. Highlight any key differences between IPv6 and IPv4. (2 Marks)

Ans: IPv6 is the latest version of the internet Protocol, designed to address the limitation of IPv4.

Reason for development: IPv6, with its 128-bit address scheme, provides approximately 3.4 * 10^38 addresses which is way larger than IPv4 addresses. IPv4 uses a 32-bit address scheme which allows for approximately 4.3 billion addresses.

IPv6 is designed with security in mind and includes support for IPsec.

Difference between IPv4 and IPv6:

1. IPv4 addresses are 32 bits, while IPv6 addresses are 128 bits.
2. IPv4 addresses are written in dotted-decimal format (e.g., 192.168.1.1) while IPv6 addresses are written in hexadecimal separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
3. IPv6 headers do not have a checksum, whereas IPv4 headers do.

2. Discuss the security benefits and challenges associated with the adoption of IPv6, particularly in comparison to IPv4. (3 Marks)

Ans: Benefits:
IPsec is a standard feature in IPv6, whereas it is optional in IPv4.
In IPv6, routers don't perform packet fragmentation. This is managed by the source device, reducing the chance of attacks.

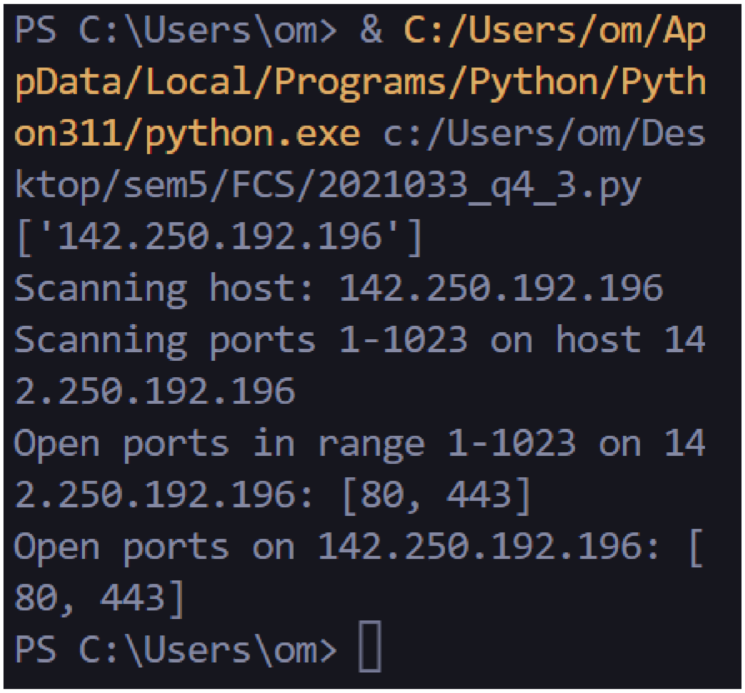
Challenges:
New protocols and features in IPv6 can be targets for attackers if not correctly understood and secured.
Some DNS zones are only served by IPv4-only authoritative servers.
It is not compatible with IPv4, meaning that direct communication between them requires translation or tunneling.

3. You are a network administrator responsible for ensuring IPv6 security. Write a script that tscans a network for IPv6-enabled devices and checks if they have any known securit vulnerabilities (e.g., open ports or misconfigured settings). Describe the tools and libraries you used in your script and provide a code snippet illustrating how your script detects potential vulnerabilities. (10 Marks)

Ans.
Tools and Libraries:

Nmap: A security scanner is used to discover devices running on a network and find open ports and various network attributes.

Here is a sampel code I ran:

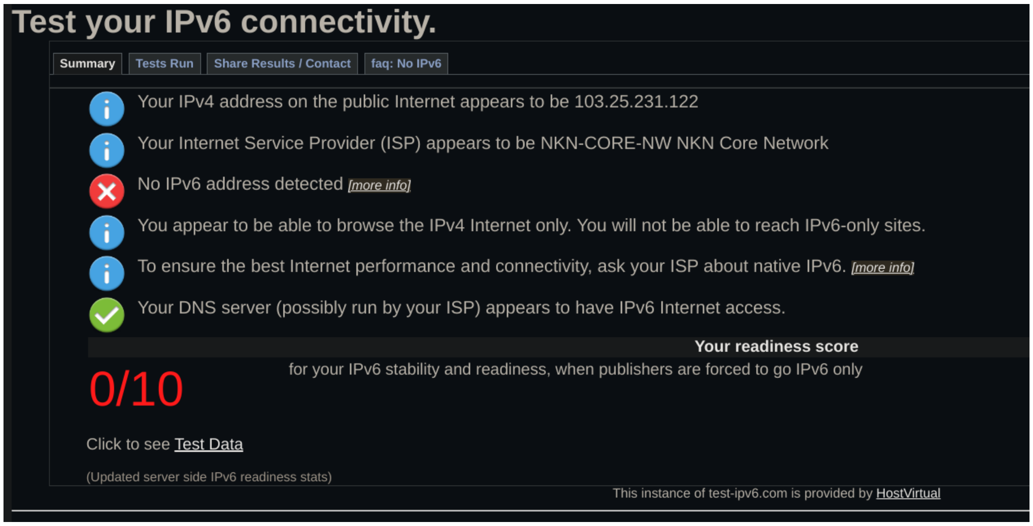


```
PS C:\Users\om> & C:/Users/om/Ap
pData/Local/Programs/Python/Pyth
on311/python.exe c:/Users/om/Des
ktop/sem5/FCS/2021033_q4_3.py
['142.250.192.196']
Scanning host: 142.250.192.196
Scanning ports 1-1023 on host 14
2.250.192.196
Open ports in range 1-1023 on 14
2.250.192.196: [80, 443]
Open ports on 142.250.192.196: [
80, 443]
PS C:\Users\om> 
```

PF code attached

Note for this question that i am not using -6 flag while generating the output and I am not scanning an IPv6 network because neither my machine nor my cryptography lab machine has the IPv6 enabled and below are some test that I ran to know about it mopre.

In order to perform the same commands for the IPv6 network you have to use the -6 command while scanning and you wil be good to go.



Test your IPv6 connectivity.

ⓘ Your IPv4 address on the public Internet appears to be 103.25.231.122

ⓘ Your Internet Service Provider (ISP) appears to be NKN-CORE-NW NKN Core Network

✗ No IPv6 address detected [more info]

ⓘ You appear to be able to browse the IPv4 Internet only. You will not be able to reach IPv6-only sites.

ⓘ To ensure the best Internet performance and connectivity, ask your ISP about native IPv6. [more info]

✓ Your DNS server (possibly run by your ISP) appears to have IPv6 Internet access.

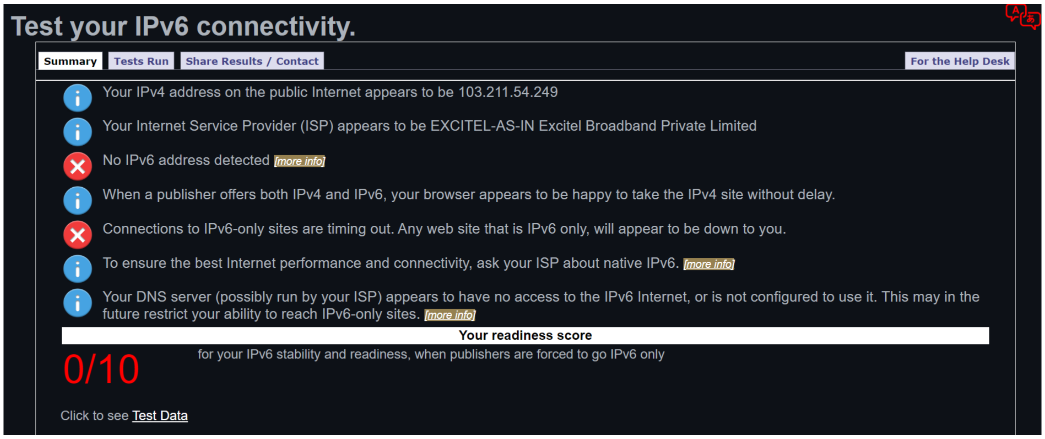
Your readiness score

0/10 for your IPv6 stability and readiness, when publishers are forced to go IPv6 only

Click to see Test Data

(Updated server side IPv6 readiness stats)

This instance of test-ipv6.com is provided by HostVirtual



Test your IPv6 connectivity.

ⓘ Your IPv4 address on the public Internet appears to be 103.211.54.249

ⓘ Your Internet Service Provider (ISP) appears to be EXCITEL-AS-IN Excitel Broadband Private Limited

✗ No IPv6 address detected [more info]

ⓘ When a publisher offers both IPv4 and IPv6, your browser appears to be happy to take the IPv4 site without delay.

✗ Connections to IPv6-only sites are timing out. Any web site that is IPv6 only, will appear to be down to you.

ⓘ To ensure the best Internet performance and connectivity, ask your ISP about native IPv6. [more info]

ⓘ Your DNS server (possibly run by your ISP) appears to have no access to the IPv6 Internet, or is not configured to use it. This may in the future restrict your ability to reach IPv6-only sites. [more info]

Your readiness score

0/10 for your IPv6 stability and readiness, when publishers are forced to go IPv6 only

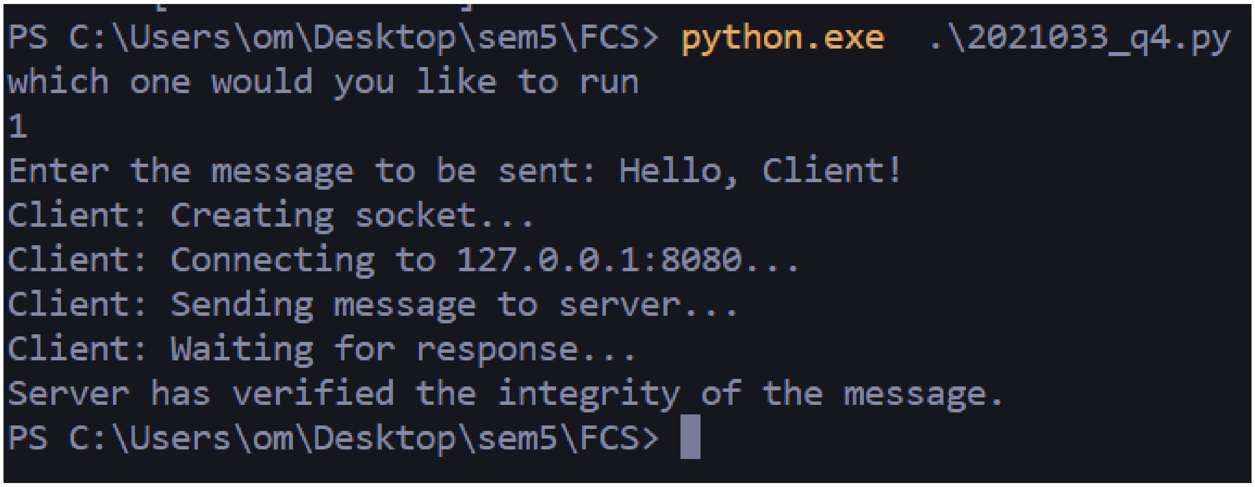
Click to see Test Data

Also for the commands and flags that i have used in the code I have used thefollwoing source: https://www.digitalocean.com/community/tutorials/how-to-use-nmap-to-scan-for-open-ports

TCP
The main script starts with:

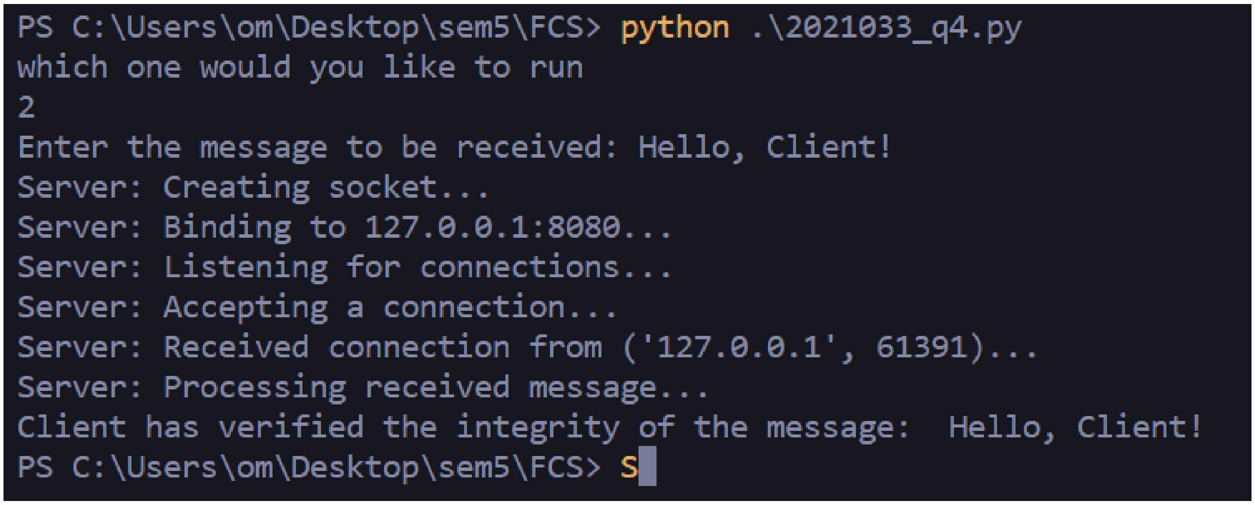
```
def answer1():
    send_message = input("Enter the message to be sent: ")
    client_send_1(send_message,"127.0.0.1", 8080)

def answer2():
    receive_message = input("Enter the message to be received: ")
    server_receive_1(receive_message,"127.0.0.1", 8080)

if __name__ == "__main__":
    print("which one would you like to run")
    program = int(input())
    if program == 1:
        answer1()
    elif program == 2:
        answer2()
    elif program == 3:
        print("Client or Server: ")
        mode = str(input())
        answer3(mode)
```

1. Write a Python program that establishes a TCP connection with a remote server and sends a simple message "Hello, Server!" to it. Include error handling to deal with potential connection issues and exceptions. (5 Marks)



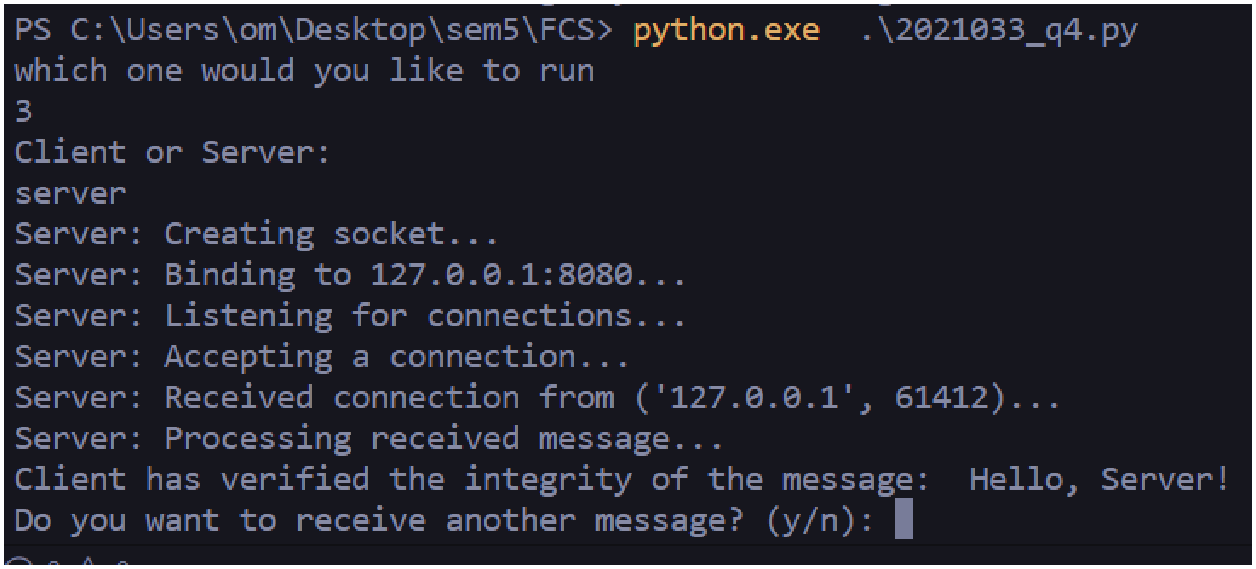
```
PS C:\Users\om\Desktop\sem5\FCS> python.exe  .\2021033_q4.py
which one would you like to run
1
Enter the message to be sent: Hello, Client!
Client: Creating socket...
Client: Connecting to 127.0.0.1:8080...
Client: Sending message to server...
Client: Waiting for response...
Server has verified the integrity of the message.
PS C:\Users\om\Desktop\sem5\FCS> 
```

2. Extend your Python program to handle a response from the server. If the server responds with "Hello, Client!", print the response. Otherwise, display an error message. (4 Marks)

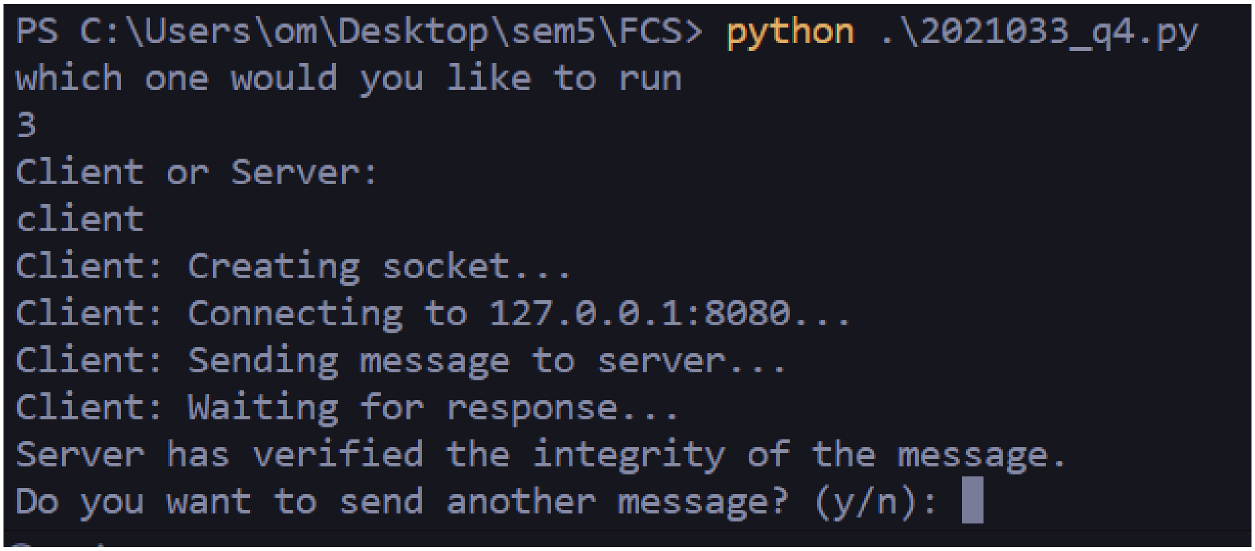


```
PS C:\Users\om\Desktop\sem5\FCS> python .\2021033_q4.py
which one would you like to run
2
Enter the message to be received: Hello, Client!
Server: Creating socket...
Server: Binding to 127.0.0.1:8080...
Server: Listening for connections...
Server: Accepting a connection...
Server: Received connection from ('127.0.0.1', 61391)...
Server: Processing received message...
Client has verified the integrity of the message:  Hello, Client!
PS C:\Users\om\Desktop\sem5\FCS> S
```

3. Implement a basic security feature in your program. Before sending the message, calculate its SHA-256 hash and send both the message and the hash to the server. The server should verify the message's integrity by calculating its hash and comparing it to the received hash. If they match, respond with "Message Verified"; otherwise, respond with "Message Tampered." (6 Marks)



```
PS C:\Users\om\Desktop\sem5\FCS> python.exe  .\2021033_q4.py
which one would you like to run
3
Client or Server:
server
Server: Creating socket...
Server: Binding to 127.0.0.1:8080...
Server: Listening for connections...
Server: Accepting a connection...
Server: Received connection from ('127.0.0.1', 61412)...
Server: Processing received message...
Client has verified the integrity of the message:  Hello, Server!
Do you want to receive another message? (y/n):
```



```
PS C:\Users\om\Desktop\sem5\FCS> python .\2021033_q4.py
which one would you like to run
3
Client or Server:
client
Client: Creating socket...
Client: Connecting to 127.0.0.1:8080...
Client: Sending message to server...
Client: Waiting for response...
Server has verified the integrity of the message.
Do you want to send another message? (y/n):
```

All of the above questions have been answered in the code attached to this.