1. **What is RIP?**
   RIP is a dynamic routing protocol that uses hop count as the routing metric. It updates routing tables every 30 seconds.
2. **What are the versions of RIP?**
   RIP has two versions: RIP v1 (classful) and RIP v2 (classless and supports subnetting).
3. **What is an ACL? Why is it used?**
   ACLs are rules applied to interfaces to control traffic flow. They enhance network security by allowing or denying packets.
4. **What is the difference between Standard and Extended ACLs?**
   Standard ACL filters only by source IP, while Extended ACLs filter by source/destination IP, protocol, and port.
5. **What is NAT and why is it used?**
   NAT translates private IP addresses to public ones for internet access. It conserves public IPs and adds security.
6. **What is Static NAT?**
   Static NAT maps a single private IP to a single public IP. It's used when the device needs constant access.
7. **What is Dynamic NAT?**
   Dynamic NAT uses a pool of public IPs for translating private IPs. Mapping is temporary.
8. **What is PAT?**
   PAT (or NAT Overload) allows multiple private IPs to share one public IP using unique port numbers.
9. **What does `ip nat inside` and `ip nat outside` mean?**
   These commands define which router interfaces are inside the private network and which face the public network.
10. **How do you verify NAT is working?**
    Use `show ip nat translations` and `show ip nat statistics`.

🔹 EIGRP:

1. **What is EIGRP?**
   EIGRP is a Cisco-proprietary, advanced distance-vector routing protocol that uses DUAL algorithm for loop-free routing.
2. **What are K values in EIGRP?**
   K-values define the metric formula for EIGRP. K1 and K3 (bandwidth and delay) are enabled by default.
3. **What is the EIGRP metric formula?**
   Metric = 256 × (Bandwidth + Delay)
4. **What must match for EIGRP neighbors to form?**
   AS number, K-values, subnet, and hello/dead timers must match.

🔹 OSPF:

5. **What is OSPF?**
   OSPF is a link-state routing protocol using the shortest-path-first (Dijkstra's) algorithm.
6. **What is the OSPF cost?**
   OSPF cost is 100,000,000 divided by the interface bandwidth in bps.
7. **What are OSPF neighbor states?**
   States include Down, Init, 2-Way, ExStart, Exchange, Loading, and Full.
8. **What is an OSPF area?**
   Logical grouping of routers. All areas must connect to the backbone (Area 0).

🔹 WLAN:

9. **What is MAC filtering in WLAN?**
   MAC filtering allows or blocks devices based on their MAC addresses for better security.
10. **What is the role of DHCP in WLAN?**
    DHCP dynamically assigns IP addresses to wireless clients, making network management easier.
11. **Difference between Static IP and DHCP?**
    Static IPs are manually set, while DHCP assigns IPs automatically to devices.

General:

1. **What is a socket in Python?**
   A socket is an endpoint that allows two machines to communicate over a network using IP and port.
2. **What is the difference between TCP and UDP?**
   TCP is reliable and connection-based; UDP is faster but doesn't guarantee delivery.
3. **Which library is used for socket programming in Python?**
   The built-in `socket` module is used for network communications.

TCP:

4. **What function does a TCP server use to accept connections?**
   The `accept()` function waits and accepts a client connection.
5. **How does the client connect to the server in TCP?**
   The client uses `connect((host, port))` to establish a TCP connection.
6. **Why is TCP called connection-oriented?**
   Because it first establishes a connection using a handshake before data transfer.

UDP:

7. **Why is UDP faster than TCP?**
   UDP does not use handshaking, acknowledgments, or error checking, so it's faster.
8. **How is data sent in UDP?**
   Using `sendto()` and `recvfrom()` functions which work without a connection.
9. **Is data delivery guaranteed in UDP?**
   No, UDP is unreliable and does not guarantee message delivery or order.

General Server Admin

1. **What is the role of a server?**
   A server provides resources or services (like websites or files) to client devices over a network.
2. **What is server administration?**
   It involves managing and maintaining server software, hardware, security, and performance.
3. **How do you check if a service is running in Linux?**
   Use `systemctl status <service>` or `ps -aux | grep <service>` to check status.

FTP Server

4. **What is FTP used for?**
   FTP is used to transfer files between computers over a network.
5. **What is the default FTP port?**
   Port 21 is the default port for FTP.
6. **What is the difference between active and passive FTP?**
   In active mode, the client opens a port to receive data; in passive mode, the server opens a port.

Web Server

7. **What is the purpose of a web server?**
   A web server hosts websites and delivers web pages to clients via HTTP.
8. **What is the default port of HTTP?**
   Port 80 is for HTTP, and port 443 is for HTTPS.
9. **What is the default web directory in Apache?**
   `/var/www/html` is the default directory where web files are stored.
10. **How do you restart the Apache server?**
    Use `sudo systemctl restart apache2`.

RSA

1. **What is RSA?**
   RSA is an asymmetric cryptographic algorithm using a public-private key pair for secure communication.
2. **Why is RSA secure?**
   It relies on the computational difficulty of factoring large prime numbers.

Digital Signature

3. **What is a digital signature?**
   It's a cryptographic technique to verify the authenticity and integrity of a message using private keys.
4. **How is it verified?**
   The receiver uses the sender's public key to verify the signature against the hashed message.

DES + DH

5. **What is DES?**
   Data Encryption Standard is a symmetric-key algorithm used for encrypting data in 64-bit blocks.
6. **What is Diffie-Hellman used for?**
   It's a method of securely exchanging cryptographic keys over a public channel.

Snort IDS

7. **What is Snort?**
   Snort is a real-time intrusion detection and prevention system that analyzes network traffic.
8. **What is a Snort rule?**
   A snort rule defines how to match packets and what action to take when a match is found.

1. Client-Server Communication Using RSA Cryptosystem

🔐 Concept:

- RSA is an **asymmetric encryption** method using public and private keys.
- The **client encrypts** the message using the server's **public key**.
- Server **decrypts** it using its **private key**.

✅ Steps:

1. Generate RSA keys (`Crypto.PublicKey.RSA`)
2. Server shares **public key** with client
3. Client encrypts a message
4. Server decrypts it

🔧 Sample Code:

Server.py

python
CopyEdit

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
import socket


key = RSA.generate(2048)
private_key = key
public_key = key.publickey().export_key()


server = socket.socket()
server.bind(('localhost', 9999))
server.listen(1)
conn, addr = server.accept()


conn.send(public_key)


encrypted = conn.recv(4096)
cipher = PKCS1_OAEP.new(private_key)
decrypted = cipher.decrypt(encrypted)
print("Decrypted:", decrypted.decode())
```

Client.py

python
CopyEdit

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP

import socket


client = socket.socket()

client.connect(('localhost', 9999))


public_key = RSA.import_key(client.recv(4096))

cipher = PKCS1_OAEP.new(public_key)


msg = "Hello Secure World"

encrypted = cipher.encrypt(msg.encode())

client.send(encrypted)
```

---

2. Client-Server RSA Digital Signature Authentication

✍️ Concept:

- Client signs a message with **private key**
- Server verifies it using client's **public key**
- This authenticates the **sender**

✅ Steps:

1. Generate key pair on client
2. Sign message hash using private key
3. Server verifies signature with public key

🔧 Sample Conceptual Code:

Client:

python
CopyEdit

```
from Crypto.Signature import pkcs1_15
from Crypto.Hash import SHA256

from Crypto.PublicKey import RSA
```

```
msg = b"Authenticate me"
key = RSA.generate(2048)
hash_msg = SHA256.new(msg)
signature = pkcs1_15.new(key).sign(hash_msg)

# Send: msg, signature, public_key
```

Server:

python
CopyEdit

```
from Crypto.Signature import pkcs1_15
from Crypto.Hash import SHA256
from Crypto.PublicKey import RSA

hash_msg = SHA256.new(msg)
try:
    pkcs1_15.new(public_key).verify(hash_msg, signature)
    print("Verified")
except:
    print("Invalid Signature")
```

---

3. Encrypt Message Using DES + Key Exchange Using Diffie-Hellman

🔒 Concept:

- DES is symmetric; **same key** is used for encryption/decryption.
- Key is exchanged securely using **Diffie-Hellman**.
- Ensures confidentiality over insecure channel.

✅ Steps:

1. Perform DH Key Exchange to derive common secret
2. Use that as DES key
3. Encrypt message and send
4. Decrypt on server

🔧 Sample Conceptual Code:

Key Exchange:

python
CopyEdit

```
# Agreed prime & base
p, g = 23, 5


# Private keys
a, b = 6, 15
A = pow(g, a, p)
B = pow(g, b, p)


# Shared key
key_client = pow(B, a, p)
key_server = pow(A, b, p)
```

DES:

python
CopyEdit

```
from Crypto.Cipher import DES
from Crypto.Util.Padding import pad, unpad


des = DES.new(b'12345678', DES.MODE_ECB)
cipher_text = des.encrypt(pad(b'Hello', 8))
plain = unpad(des.decrypt(cipher_text), 8)
```

---

4. Snort Intrusion Detection System

🛡️ Concept:

- Snort is an open-source **NIDS** used to detect suspicious network activity.
- Can capture packets and match them with **custom rules**.

✅ Usage:

1. Install Snort: `sudo apt install snort`
2. Monitor traffic: `snort -A console -i eth0 -c /etc/snort/snort.conf`
3. Create rule:

   cpp
   CopyEdit

   ```
   alert tcp any any -> any 80 (msg:"HTTP Access Detected"; sid:1000001;)
   ```

4. Place it in `/etc/snort/rules/local.rules` and include in `snort.conf`.
5. Restart snort: `sudo systemctl restart snort`