1) Configure a router using router commands and Configure Routing Information Protocol(RIP).

ACL stands for Access Control List. It is a set of rules defined on a network device (such as a router or a firewall) that controls the flow of traffic by permitting or denying network packets based on criteria such as source IP address, destination IP address, protocol, port numbers.

When activating an ACL on an interface, you must specify in which direction the traffic should be filtered:

- Inbound (as the traffic comes into an interface) packet will discarded after denied by filtering tests
- Outbound (before the traffic exits an interface)

Standard ACL: Standard IP ACLs can filter only on the source IP address inside a packet. You can permit or deny only source traffic.

Extended ACL: filter on the source and destination IP addresses in the packet.

There are two actions an ACL can take: permit or deny.

ACL statements are processed top-down.Once a match is found, no further statements are processed—therefore, order is important. If no match is found, the imaginary implicit deny statement at the end of the ACL drops the packet.

you can have only one ACL per protocol, per interface, per direction.

Type IP	Range
Standard IP	1–99
Extended IP	100–199
Standard Expanded Range	1300–1999
IP Extended Expanded Range	2000–2699

Router RIP Command:

1. Match a specific host
2. Match an entire subnet
3. Match an IP range
4. Match Everyone and anyone

1.Match Specific Host: block 10.0.0.3 from gaining access on 40.0.0.0

```
R2>enable
```

```
R2#configure terminal Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#access-list 1 deny host 10.0.0.3

R2(config)#access-list 1 permit any

R2(config)#interface fastEthernet 0/1

R2(config-if)#ip access-group 1 out
```

To test first do ping from 10.0.0.3 to 40.0.0.3 it should be request time out as this packet will filter by ACL. Then ping 30.0.0.3 it should be successfully replay.

The wildcard mask uses a binary format where each bit has a specific meaning:

A wildcard bit value of 0 means an exact match is required for the corresponding bit in the IP address. A wildcard bit value of 1 means that the corresponding bit in the IP address is ignored (it matches any value).

2.Match an entire subnet:

Wildcards are used with access lists to specify an individual host, a network, or a certain range of a network or networks. Formula to calculate wild card mask for access list

Wildcard mask = 255.255.255.255 – subnet

3. Match an IP range:

Your task is to block an ip range of 10.3.16.0 – 10.3.31.255 from gaining access to the network of 40.0.0.0 . take the higher IP and subtract from it the lower IP.

R2(config)#access-list 2 deny 10.3.16.0   0.0.15.255 here start from 10.3.16.0 to diff is 0.0.15.255 .

4. Match EveryOne : access-list 1 permit 0.0.0.0 255.255.255.255

We could use extended ACL to secure telnet session but if you did that, you'd have to apply it inbound on every interface, and that really wouldn't scale well to a large router with dozens, even hundreds, of interfaces.Here's a much better solution: Use a standard IP access list to control access to the VTY lines themselves. To perform this function, follow these steps: 1. Create a standard IP access list that permits only the host or hosts you want to be able to telnet into the routers. 2. Apply the access list to the VTY line with the access-class command .

IP (Internet Protocol):

IP is a fundamental protocol in the Internet protocol suite. It is responsible for routing packets across network boundaries.

IP provides a connectionless, best-effort delivery service, meaning it does not guarantee packet delivery or ensure packet sequencing.

TCP (Transmission Control Protocol):

TCP is a connection-oriented protocol that provides reliable, ordered, and error-checked delivery of data between applications.

It establishes a connection between the sender and receiver before data transfer and ensures that data packets arrive in the correct order and without errors.

UDP (User Datagram Protocol):

UDP is a connectionless protocol that provides a lightweight and unreliable data delivery service.

Unlike TCP, UDP does not establish a connection before transmitting data and does not guarantee delivery or sequencing of packets.

ICMP (Internet Control Message Protocol)

IGRP (Interior Gateway Routing Protocol): IGRP was a proprietary distance-vector routing protocol developed by Cisco Systems for routing within an autonomous system (AS).

Established: For inbound TCP only. Allows TCP traffic to pass if the packet is a response to an outbound-initiated session. This type of traffic has the acknowledgement (ACK) bits set.

Port Number	IP Protocol
20 (TCP)	FTP data
21 (TCP)	FTP control
23 (TCP)	Telnet
25 (TCP)	Simple Mail Transfer Protocol (SMTP)
53 (TCP/UDP)	Domain Name System (DNS)
69 (UDP)	TFTP
80 (TCP)	HTTP

Three basic steps to configure Extended Access List

- Create an extended access list by specifying a number (from 100 to 199 or from 2000 to 2699)
- Define permit or deny statements to specify the traffic you want to permit or deny.
- permit tcp <source_ip> <source_wildcard_mask> <destination_ip> <destination_wildcard_mask> eq <port>1.

1. Block host to host:

```
R1>enable
R1#configure
        terminal
        R1(config)#access-list 101 deny ip host 10.0.0.3 40.0.0.3 0.0.0.0
        R1(config)#access-list 101 permit ip any
        R1(config)#interface fastEthernet 0/0
        R1(config-if)#ip access-group
        101 in R1(config-if)#exit
R1(config)#
```

2. Block host to network   10.0.0.3 40.0.0.0 0.255.255.255

3. Block Network to network   10.0.0.0 0.255.255.255 40.0.0.0 0.255.255.255

4. Block telnet access for critical resources of company

5. Limited ftp access for user

6. Stop exploring of private network form ping

7. Limited web access

8. Configure established keyword

The established keyword is a advanced feature that will allow traffic through only if it sees that a TCP session is already established. A TCP session is considered established if the three-way handshake is initiated first. You can use TCP established to deny all traffic into your network except for incoming traffic that was first initiated from inside your network. This is commonly used to block all originating traffic from the Internet into a company's network except for Internet traffic that was first initiated from users inside the company.

Grant FTP access to limited user:

1. Create an FTP user:
2. Configure vsftpd:
3. Set up chroot jail:
4. Set up directory restrictions:
5. Restart vsftpd:
6. Set permissions:

   sudo chown ftpuser:ftpuser /home/ftpuser

   sudo chmod 700 /home/ftpuser

Static NAT In this type we manually map each inside local IP address with inside global IP address.

Dynamic NAT In this type we create a pool of inside global IP addresses and let the NAT device to map inside local IP address with the available outside global IP address from the pool automatically.

PAT In this type a single inside global IP address is mapped with multiple inside local IP addresses using the source port address.

show ip nat translation

a) Configure EIGRP – Explore Neighbor-ship Requirements and Conditions, its K Values Metrics Assignment and Calculation:

An Autonomous System (AS) is network or group of networks that is managed by a single organization and operates under a single, consistent routing policy.

ERGRP, or Enhanced Interior Gateway Routing Protocol, is a routing protocol used in computer networks, particularly in interior networks (within an organization or an autonomous system).

Purpose: The primary purpose of ERGRP is to enable routers within a network to dynamically learn routes to destinations and efficiently forward packets based on this routing information. This dynamic routing capability allows for automatic adaptation to network changes, such as link failures or the addition of new routers.

IP Connectivity: Routers must have IP connectivity between them. This means they should be able to ping each other's IP addresses. EIGRP uses DUAL (Diffusing Update Algorithm) to provide the fastest route convergence among all protocols.

AS Number: Routers must be in the same Autonomous System (AS) and configured with the same EIGRP Autonomous System Number.

Hello and Hold Timers: Hello and hold timers define how often routers send hello packets and how long they wait for hellos from neighbors before considering the neighbor down.

K1: Bandwidth: take lowest bandwidth EIGRP first looks at bandwidth command. If bandwidth is set through this command, EIGRP will use it. If bandwidth is not set, it will use interface's default bandwidth.

K2: Load: • Txload for outgoing traffic • Rxload for incoming traffic

K3: Delay

K4: Reliability

K5: MTU: MTU stands for maximum transmission unit.

EIGRP COnfirguration:

router eigrp <AS_number>

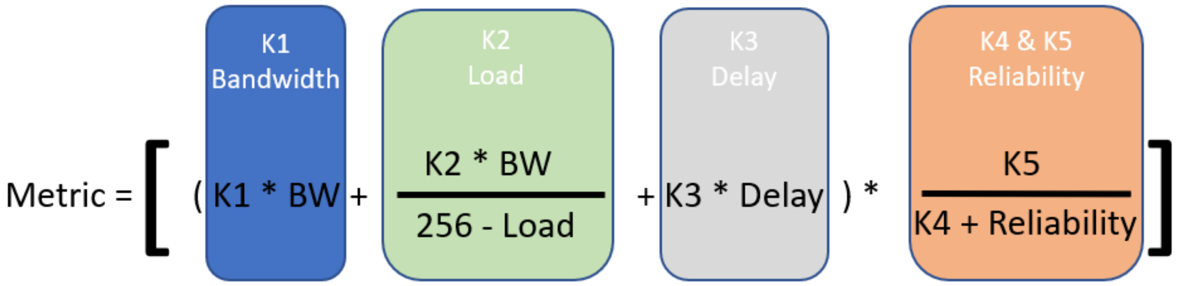
eigrp router-id <IP_address>

network <network_address> <wildcard_mask>

1. Neighbor Table

2. Topology Table: EIGRP uses this table to store all routes which it learned from neighbors. It contains a list of all destinations and routes advertised by neighboring routers. EIGRP selects single best route for each destination from this list.

3. Routing Table:



$$\text{Metric} = \left[\left(K1 * BW + \frac{K2 * BW}{256 - Load} + K3 * Delay \right) * \frac{K5}{K4 + Reliability} \right]$$

b) OSPF :

There are two types of routing protocols IGP and EGP. • IGP (Interior Gateway Protocol) is a routing protocol that runs in a single AS such as RIP, IGRP, EIGRP, OSPF and IS-IS. • EGP (Exterior Gateway Protocol) is a routing protocol that performs routing between different AS systems. Nowadays only BGP (Border Gateway Protocol) is an active EGP protocol.

1.OSPF routers exchange Link-State Advertisements (LSAs) to communicate their routing information. Each router maintains a link-state database containing information about the state of links in the network. LSAs are flooded throughout the OSPF area to ensure that all routers have consistent information about the network topology.

2.Dijkstra Algorithm: OSPF uses the Dijkstra algorithm to calculate the shortest path tree (SPT) for each router in the network. This tree is used to determine the best path to reach each network destination.

Cost = Reference Bandwidth / Interface Bandwidth

In order to become OSPF neighbor following values must be match on both routers.

- Area ID • Authentication • Hello and Dead Intervals • Stub Flag • MTU Size

In OSPF (Open Shortest Path First) networks, the concepts of DR (Designated Router) and BDR (Backup Designated Router) are used to optimize the exchange of routing information within multi-access networks such as Ethernet LANs.

DR: In multi-access OSPF networks (such as Ethernet), all routers must be fully adjacent (fully adjacent means forming neighbor relationships with all other routers in the network).

BDR: Backup Designated Router:

c) WLAN with static IP addressing and DHCP with MAC security and filters

WLAN with Static IP Addressing: Static IP addressing involves manually assigning IP addresses to devices within a network rather than dynamically assigning them through a DHCP server.

Dynamic Host Configuration Protocol (DHCP):DHCP automates the process of assigning IP addresses, subnet masks, default gateways, and other network parameters to devices on a network.

MAC (Media Access Control) Security: MAC security involves restricting network access based on the MAC address of the device. DHCP server maintains a list of allowed MAC addresses, and only devices with matching MAC addresses are assigned IP addresses.

TCP UDP:

Sockets are used for interprocess communication. Most of the interprocess communication follow a Client-Server

socket() creates an endpoint for communication and returns a file descriptor for the socket. It is used to create an endpoint for communication, which can be a connection-oriented socket (e.g., for TCP) or a connectionless socket (e.g., for UDP).

bind() assigns a socket to an address. bind() takes three arguments:

sockfd, a descriptor representing the socket to perform the bind on.

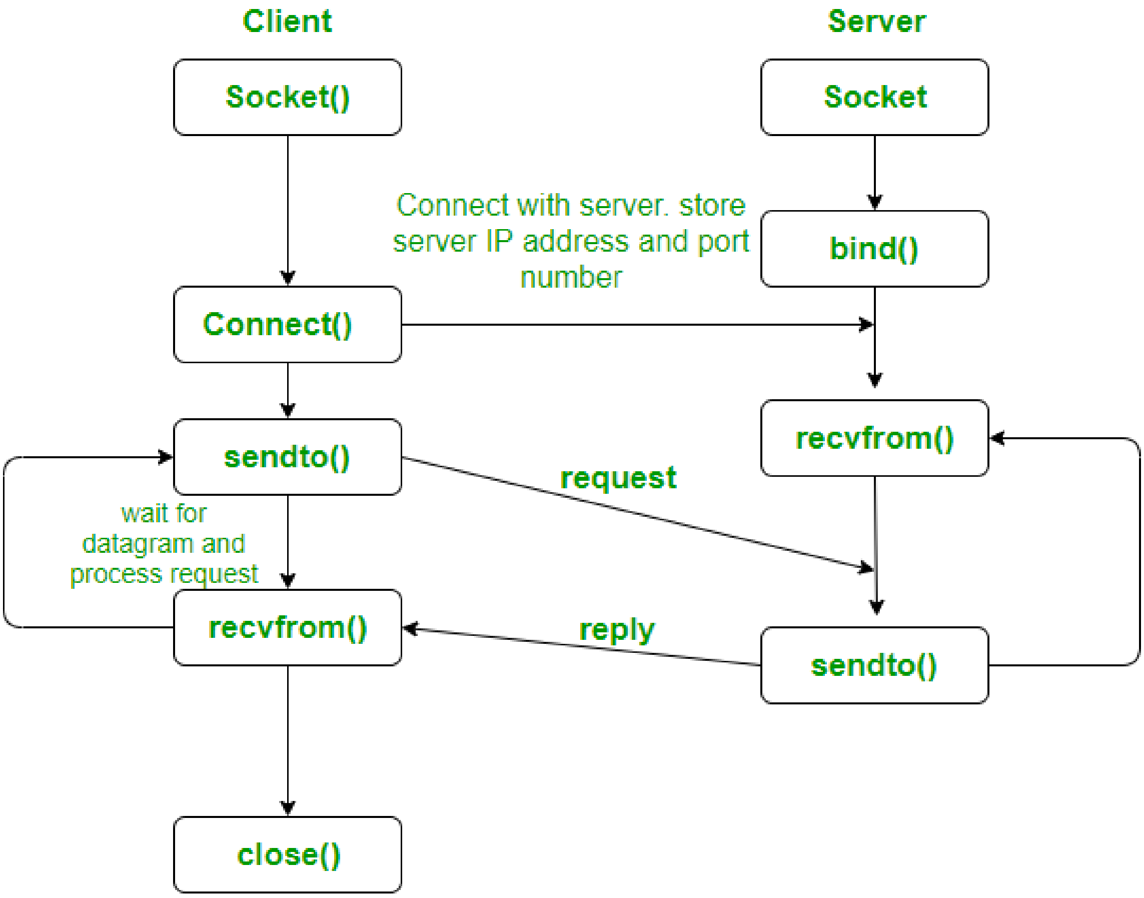
my_addr, a pointer to a sockaddr structure representing the address to bind to.

addrlen, a socklen_t field specifying the size of the sockaddr structure.

listen() is used on a socket that has been bound to a local address using bind(), and it marks the socket as a passive socket that can accept incoming connections.

accept()

connect()


Client
Server
Socket()
Socket
bind()
Connect with server. store server IP address and port number
Connect()
sendto()
recvfrom()
request
wait for datagram and process request
recvfrom()
reply
sendto()
close()

Regular FTP:

In regular FTP, the client establishes two separate TCP connections to the server: one for control (command) information and another for data transfer.

Port 21 is used for the control connection, where commands and responses are exchanged between the client and the server.

Port 20 is traditionally used for the data connection, where the actual file transfers occur. However, in some cases, the data connection may use other ports.

Active FTP: In Active FTP, the client initiates both the control and data connections to the server.

The client sends a PORT command over the control connection, specifying an IP address and port number for the server to connect to for the data transfer.

Passive FTP:

In Passive FTP, the client initiates the control connection to the server, but the data connection is established by the server.The client sends a PASV (Passive) command over the control connection to request a passive data transfer mode.The server responds with an IP address and port number that the client can connect to for the data transfer.The client then initiates a separate TCP connection to the server's specified IP address and port number for the data transfer.

Anonymous FTP:Anonymous FTP allows users to connect to an FTP server and log in without providing a username or password.

```
nano /etc/vsftpd.conf
```

local_enable=YES allows any user account defined in the /etc/passwd file access to the FTP server and is generally how most FTP users will connect.

Securing FTP with SSL (Secure Sockets Layer) or its successor TLS (Transport Layer Security) is commonly referred to as FTPS. FTPS provides encryption and authentication mechanisms to protect data transmitted between the FTP client and server.

CLIENT AND SERVER COMMUNICATION:

RSA cryptosystem:

The RSA cryptosystem is one of the most widely used public-key encryption algorithms, named after its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman. It is based on the mathematical properties of large prime numbers and is commonly used for secure data transmission and digital signatures.

Algorithm

1. Key Generation

1. Select p, q p and q both prime, p ! q

2. Calculate $n = p * q$

3. Calculate $\Phi(n) = (p - 1)(q - 1)$

4. Select integer e ,such that $\gcd(\Phi(n), e) = 1; 1 < e < \Phi(n)$

5. Calculate $d = e-1 \bmod \Phi(n)$

6. Public key $PU = \{e, n\}$

7. Private key $KR = \{d, n\}$

2. Encryption

Plaintext: $M < n$

Ciphertext: $C = Me \pmod n$

3. Decryption

Ciphertext: C

Plaintext: $M = Cd(\bmod n)$

RSA digital signature cryptosystem:

The RSA digital signature cryptosystem is a variant of the RSA encryption algorithm used for creating and verifying digital signatures

Key Generation:

- Choose two large prime numbers p and q.

- Compute the modulus $n = p * q$ and the Euler's totient function $\phi(n) = (p-1)(q-1)$.

- Choose a public exponent e that is relatively prime to $\phi(n)$, typically a small prime such as 65537.

- Calculate the private exponent d such that $d * e \equiv 1 \bmod \phi(n)$.

- The public key is (n, e) and the private key is (n, d).

Signing a Message:

- Calculate the hash of the message using a cryptographic hash function (e.g., SHA-256).

- Encrypt the hash value using the private key d to generate the digital signature S: $S \equiv H(M)^d \bmod n$.

- Attach the digital signature S to the message and send both to the recipient.

Verifying the Signature:

- Recalculate the hash of the received message using the same cryptographic hash function.

- Decrypt the digital signature S using the sender's public key (n, e): $S' \equiv S^e \bmod n$.

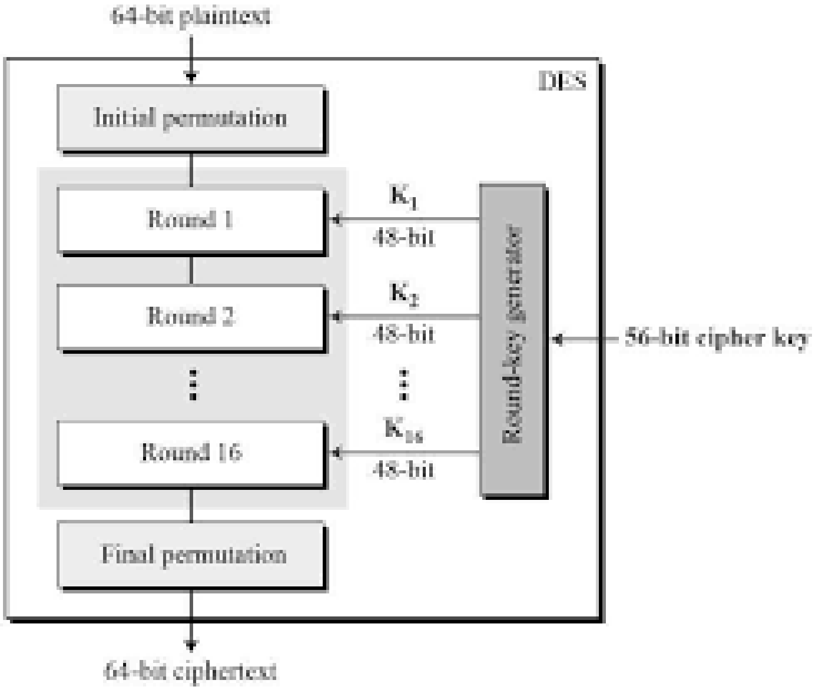
- If S' equals the recalculated hash value, then the signature is valid and the message has not been altered in transit.

Security:

- The security relies on the difficulty of computing the private exponent d from the public key (n, e) without knowledge of the prime factors p and q of n.

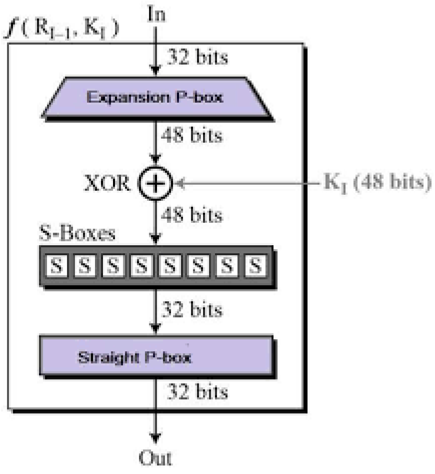
- The strength of the cryptographic hash function used for generating the message hash is also crucial for security.

DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration



The heart of this cipher is the DES function, f. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.

Avalanche effect − A small change in plaintext results in the very great change in the ciphertext.
Completeness − Each bit of ciphertext depends on many bits of plaintext.



```
KeyGeneration(key):
    // Generate 16 subkeys
    subkeys = KeySchedule(key)

InitialPermutation(plaintext):
    // Perform initial permutation
    permuted_plaintext = Permute(plaintext, initial_permutation_table)

FeistelNetwork(plaintext, subkeys):
    left_half, right_half = Split(plaintext, 32)
    for round in 1 to 16:
        expanded_right_half = Expand(right_half)
        subkey = subkeys[round]
        xor_result = XOR(expanded_right_half, subkey)
        sbox_result = Substitute(xor_result)
        permuted_result = Permute(sbox_result, permutation_table)
        new_right_half = XOR(permuted_result, left_half)
        left_half = right_half
        right_half = new_right_half
    return Concatenate(right_half, left_half)

FinalPermutation(ciphertext):
    // Perform final permutation
    permuted_ciphertext = Permute(ciphertext, final_permutation_table)

Decryption(ciphertext, subkeys):
    // Decrypt ciphertext using subkeys in reverse order
    reversed_subkeys = Reverse(subkeys)
    plaintext = FeistelNetwork(ciphertext, reversed_subkeys)
    return plaintext
```

```
// Example usage

plaintext = "Hello, World!"

key = GenerateKey()

subkeys = KeyGeneration(key)

encrypted_text = Encrypt(plaintext, subkeys)

decrypted_text = Decrypt(encrypted_text, subkeys)
```

main attacks on DES without details:

1. Brute Force Attack
2. Meet-in-the-Middle Attack
3. Differential Cryptanalysis
4. Linear Cryptanalysis
5. Known Plaintext Attack
6. Key Schedule Weaknesses

Snort is a widely used open-source intrusion detection system (IDS) that monitors network traffic in real-time and can perform various actions based on predefined rules. These rules, known as signatures, are used to detect and alert on suspicious or malicious network activity. Here's how you can use Snort to analyze traffic and create a signature to identify problem traffic:

3 factors of cyber security :

1. Confidentiality
2. Integrity
3. Availability

1. **Installation and Configuration**:

- Install Snort on a system within your network that has access to the traffic you want to monitor.

- Configure Snort to listen on the network interface where the traffic of interest passes through.

- Configure the rules directory in Snort's configuration file (`snort.conf`) to point to the directory where your custom rules will be stored.

2. **Monitoring Traffic**:

- Start Snort in IDS mode to monitor network traffic.

- Snort will analyze packets passing through the network interface based on the rules defined in its configuration file.

- Snort can generate alerts, log files, or even perform specific actions (like blocking or dropping packets) when it detects matching traffic.

3. **Analyzing Problem Traffic**:

- Use Snort's alert logs and output to identify patterns of suspicious or problematic traffic.

- Look for common attack signatures, unusual traffic patterns, or known indicators of compromise (IoCs) in the logs.

4. **Creating Signatures**:

- Once you've identified a pattern of problematic traffic that you want to detect, create a custom Snort signature to identify it.

- Snort signatures are written in a rule format that specifies the conditions under which a packet should trigger an alert.

- A Snort rule typically consists of a header, which defines the action, protocol, source and destination IP addresses, ports, and options, followed by rule options, which define the specific characteristics of the traffic to match.

- Use Snort's rule syntax to specify the conditions that indicate the presence of the problem traffic.

- Test the new rule to ensure that it correctly identifies the problem traffic without generating false positives.

5. **Deployment and Testing**:

- Deploy the new Snort rule on your IDS system.

- Monitor Snort alerts and logs to verify that the new rule accurately detects the problem traffic without generating false positives.

- Adjust the rule as necessary based on feedback and further analysis.

6. **Maintenance and Updates**:

- Regularly review and update your Snort rules to adapt to changes in network traffic patterns and emerging threats.

- Stay informed about new attack techniques and vulnerabilities to ensure that your Snort rules remain effective.

By following these steps, you can use Snort to analyze network traffic, identify problem traffic, and create custom signatures to detect and alert on suspicious or malicious activity within your network.

Host-based Intrusion Detection Systems (HIDS) and Network-based Intrusion Detection Systems (NIDS) are two types of intrusion detection systems that serve different purposes and operate at different levels within a network. Here's a comparison between HIDS and NIDS:

1. **Scope**:

- HIDS: Focuses on monitoring and analyzing activity on individual hosts (e.g., computers, servers, endpoints) within a network.

- NIDS: Focuses on monitoring and analyzing network traffic passing through network segments, such as LANs, WANs, or specific network devices like routers or switches.

2. **Location**:

- HIDS: Deployed directly on individual hosts as software agents or sensors. They run on the host's operating system and monitor activities such as file system changes, registry modifications, network connections, and system calls.

- NIDS: Deployed at strategic points within the network infrastructure, such as network perimeter, internal network segments, or data centers. They monitor network traffic by analyzing packets passing through network devices.

3. **Visibility**:

- HIDS: Provides granular visibility into the activities and behaviors of individual hosts, allowing detection of host-based threats, insider attacks, and unauthorized access attempts.

- NIDS: Provides visibility into network traffic flowing across the network, allowing detection of network-based threats, external attacks, port scans, and malicious communication.

4. **Detection Techniques**:

- HIDS: Uses host-based detection techniques, such as file integrity monitoring (FIM), system log analysis, registry monitoring, and behavioral analysis, to detect signs of malicious activity or security breaches on individual hosts.

- NIDS: Uses network-based detection techniques, such as signature-based detection, anomaly detection, and protocol analysis, to detect suspicious or malicious network traffic, including known attack patterns, network-based malware, and unauthorized access attempts.

5. **Alerting and Response**:

- HIDS: Generates alerts and notifications when suspicious activity is detected on individual hosts. Alerts are typically sent to a centralized management console or security operations center (SOC) for investigation and response.

- NIDS: Generates alerts and notifications when suspicious activity is detected in network traffic. Alerts are often sent to a centralized management console or SOC for analysis and response. NIDS may also take proactive actions, such as blocking or quarantining malicious traffic.

6. **Deployment Considerations**:

- HIDS: Suitable for environments where host-level security monitoring and protection are critical, such as endpoint security, server protection, and compliance monitoring.

- NIDS: Suitable for environments where monitoring network traffic across multiple hosts or network segments is necessary, such as network security monitoring, threat detection, and incident response

In summary, HIDS and NIDS are complementary technologies that provide different levels of visibility and detection capabilities within a network. While HIDS focuses on monitoring activity on individual hosts, NIDS focuses on monitoring network traffic flowing across the network infrastructure. Organizations often deploy both HIDS and NIDS as part of a comprehensive intrusion detection and prevention strategy to enhance security posture and detect a wide range of threats.

1. **Class A**:
   - Range: 0.0.0.0 to 127.255.255.255.
2. **Class B**:
   - Range: 128.0.0.0 to 191.255.255.255.
3. **Class C**:
   - Range: 192.0.0.0 to 223.255.255.255.
4. **Class D**:
   - Range: 224.0.0.0 to 239.255.255.255.
5. **Class E**:
   - Range: 240.0.0.0 to 255.255.255.255.

AES (Advanced Encryption Standard) and DES (Data Encryption Standard):

1. **Key Length**:
   - AES: Supports key lengths of 128, 192, and 256 bits.
   - DES: Has a fixed key length of 56 bits (64 bits including parity bits), although DESede (Triple DES) can effectively use key lengths of 112 or 168 bits.
2. **Security**:
   - AES: Considered highly secure and resistant to cryptographic attacks. It is the standard encryption algorithm used by governments and organizations worldwide.

   - DES: Vulnerable to brute-force attacks due to its short key length. It is no longer considered secure for most applications, especially given advances in computing power.
3. **Speed**:
   - AES: Generally faster than DES, especially with hardware acceleration and optimized implementations. AES is designed for efficient encryption and decryption.
   - DES: Slower compared to AES, particularly when using Triple DES (DESede) due to its multiple encryption rounds.
4. **Algorithm Complexity**:
   - AES: Uses a substitution-permutation network (SPN) with multiple rounds of encryption, including byte substitution, row shifting, column mixing, and key addition.
   - DES: Utilizes a Feistel network structure with 16 rounds of encryption, including permutation, substitution, and key mixing.
5. **Block Size**:
   - AES: Supports a fixed block size of 128 bits.
   - DES: Operates on a block size of 64 bits.
6. **Adoption**:
   - AES: Widely adopted as the standard encryption algorithm in various applications, including data encryption, secure communication protocols (e.g., SSL/TLS), and disk encryption.
   - DES: Deprecated for most applications due to security concerns. Triple DES (DESede) is still used in legacy systems but is being replaced by AES.

The OSI (Open Systems Interconnection) model and the TCP/IP (Transmission Control Protocol/Internet Protocol) model are two conceptual frameworks used to understand and standardize the communication process in computer networks. Here's a comparison between OSI and TCP/IP:

1. **Layers**:
   - OSI Model: Consists of seven layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application.
   - TCP/IP Model: Consists of four layers: Network Interface, Internet, Transport, and Application.
2. **Scope**:
   - OSI Model: Provides a comprehensive framework for understanding network communication, with each layer responsible for specific functions and protocols.

- TCP/IP Model: Developed to standardize the protocols used on the Internet, focusing on the most common networking functions required for communication between computers.

3. **Layer Functionality**:
   - OSI Model: Each layer performs specific functions related to data transmission, encapsulation, addressing, routing, and application support. For example, the Network layer handles routing and logical addressing, while the Presentation layer is responsible for data translation and encryption.
   - TCP/IP Model: The layers are less strictly defined and often overlap with OSI layers. The Application layer includes protocols for communication between applications, the Transport layer manages end-to-end communication, the Internet layer handles packet routing, and the Network Interface layer deals with hardware-level communication.
4. **Protocols**:
   - OSI Model: Does not specify specific protocols but serves as a guideline for developing interoperable networking protocols. Common protocols associated with OSI include TCP/IP, Ethernet, and HTTP.
   - TCP/IP Model: Specifies the use of specific protocols at each layer, including TCP, UDP, IP, ARP, ICMP, and DNS.
5. **Adoption**:
   - OSI Model: Widely used as a conceptual framework for understanding networking concepts and designing network architectures. However, it is not strictly implemented in practice.
   - TCP/IP Model: The de facto standard for networking protocols on the Internet and most modern computer networks. TCP/IP protocols are widely adopted and form the basis of the modern Internet.
6. **Flexibility**:
   - OSI Model: Offers a more flexible and modular approach to network design and development, allowing for interoperability between different vendor systems.
   - TCP/IP Model: Provides a simpler and more practical approach, making it easier to implement and troubleshoot networking solutions, particularly for Internet-based applications.