Unit-5
TRANSPORT LEVEL SECURITY

WEB SECURITY REQUIREMENTS:

Transport-level security, often referred to as Transport Layer Security (TLS) or its predecessor Secure Sockets Layer (SSL), is crucial for ensuring the security of data during its transmission over the Internet. Here are some key web security requirements in transport-level security:

1. **Encryption:**
    - **SSL/TLS Protocols:** Use the latest versions of SSL or TLS protocols to encrypt data during transmission. It's essential to stay updated with the latest security standards, as older versions may have vulnerabilities.
    - **Strong Encryption Algorithms:** Utilize strong cryptographic algorithms for encryption, such as Advanced Encryption Standard (AES). Avoid using weak algorithms that are susceptible to attacks.
2. **Certificates and Public Key Infrastructure (PKI):**
    - **Digital Certificates:** Obtain and use digital certificates from reputable Certificate Authorities (CAs) to ensure the authenticity of your website. This helps users verify that they are connecting to the intended server and not a malicious one.
    - **Certificate Validity:** Regularly check and renew SSL/TLS certificates to ensure they are valid. Expired or compromised certificates can expose your website to security risks.
3. **Perfect Forward Secrecy (PFS):**
    - **PFS Support:** Implement Perfect Forward Secrecy to ensure that even if a private key is compromised, past communications remain secure. This is achieved by generating unique session keys for each session.
4. **Secure Cipher Suites:**
    - **Disable Weak Cipher Suites:** Disable outdated and weak cipher suites that may be susceptible to attacks. Use only strong and secure cipher suites to ensure the confidentiality and integrity of data.
5. **HSTS (HTTP Strict Transport Security):**
    - **HSTS Header:** Implement HSTS to ensure that web browsers always connect to your site using a secure connection (HTTPS). This helps prevent man-in-the-middle attacks that attempt to downgrade the connection to HTTP.
6. **Server Name Indication (SNI):**
    - **SNI Support:** If hosting multiple websites on the same server, ensure that SNI is supported. SNI allows the server to present different SSL certificates based on the domain name requested by the client.

7. **Secure Configuration:**
    - **Disable Insecure Protocols:** Disable deprecated and insecure protocols like SSLv2 and SSLv3. Only support the latest and secure TLS versions.
    - **Secure Default Configurations:** Ensure that the server's default configurations are secure. Unnecessary services and features should be disabled or configured securely.
8. **Certificate Pinning:**
    - **Public Key Pinning:** Implement certificate pinning to bind a certificate to a specific public key. This helps prevent the acceptance of fraudulent certificates issued by compromised CAs.
9. **Monitoring and Logging:**
    - **Security Monitoring:** Implement continuous monitoring of security events related to SSL/TLS, including failed connection attempts, cipher suite negotiations, and certificate issues.
    - **Logging:** Keep detailed logs of SSL/TLS-related events for auditing and troubleshooting purposes.
10. **Regular Audits and Vulnerability Scans:**
    - **Regular Audits:** Conduct regular security audits to identify and address potential vulnerabilities in the SSL/TLS implementation.
    - **Vulnerability Scans:** Use automated tools to scan for vulnerabilities and weaknesses in your web server's SSL/TLS configuration.

By adhering to these web security requirements in transport-level security, you can help create a secure and trustworthy communication channel between your web server and clients. Keep in mind that security is an ongoing process, and regular updates and improvements are essential to stay ahead of emerging threats.

TRANSPORT LAYER SECURITY:

In order to provide an open ***Internet*** standard of SSL, IETF released The Transport Layer Security (TLS) protocol in January 1999. TLS is defined as a proposed Internet Standard in RFC 5246.

Salient Features

- TLS protocol has same objectives as SSL.
- It enables client/server applications to communicate in a secure manner by authenticating, preventing eavesdropping and resisting message modification.
- TLS protocol sits above the reliable connection-oriented transport TCP layer in the networking layers stack.
- The architecture of TLS protocol is similar to SSLv3 protocol. It has two sub protocols: the TLS Record protocol and the TLS Handshake protocol.
- Though SSLv3 and TLS protocol have similar architecture, several changes were made in architecture and functioning particularly for the handshake protocol.

Comparison of TLS and SSL Protocols

There are main eight differences between TLS and SSL protocols. These are as follows −

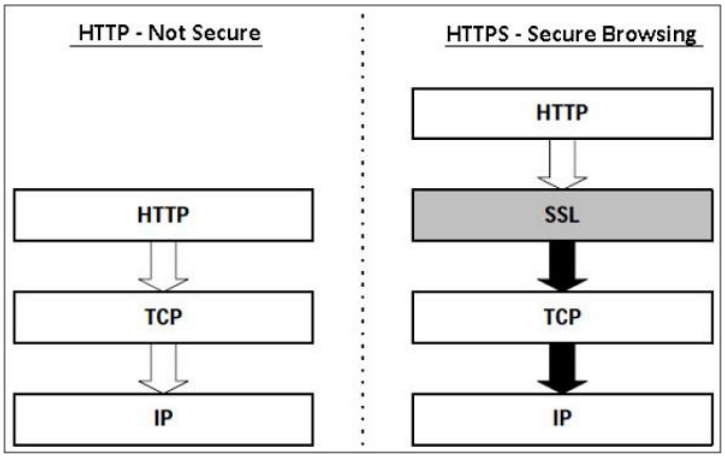
- **Protocol Version** − The header of TLS protocol segment carries the version number 3.1 to differentiate between number 3 carried by SSL protocol segment header.
- **Message Authentication** − TLS employs a keyed-hash message authentication code (H- MAC). Benefit is that H-MAC operates with any hash function, not just MD5 or SHA, as explicitly stated by the SSL protocol.
- **Session Key Generation** − There are two differences between TLS and SSL protocol for generation of key material.
    - Method of computing pre-master and master secrets is similar. But in TLS protocol, computation of master secret uses the HMAC standard and pseudorandom function (PRF) output instead of ad-hoc MAC.
    - The algorithm for computing session keys and initiation values (IV) is different in TLS than SSL protocol.
- Alert Protocol Message −
    - TLS protocol supports all the messages used by the Alert protocol of SSL, except *No certificate* alert message being made redundant. The client sends empty certificate in case client authentication is not required.
    - Many additional Alert messages are included in TLS protocol for other error conditions such as *record_overflow, decode_error* etc.
- **Supported Cipher Suites** − SSL supports RSA, Diffie-Hellman and Fortezza cipher suites. TLS protocol supports all suits except Fortezza.
- **Client Certificate Types** − TLS defines certificate types to be requested in a *certificate_request* message. SSLv3 support all of these. Additionally, SSL support certain other types of certificate such as Fortezza.
- Certificate Verify and Finished Messages −
    - In SSL, complex message procedure is used for the *certificate_verify* message. With TLS, the verified information is contained in the handshake messages itself thus avoiding this complex procedure.
    - Finished message is computed in different manners in TLS and SSLv3.

The above differences between TLS and SSLv3 protocols are summarized in the following table

	SSL v3.0	TLS v1.0
Protocol version in messages	3.0	3.1
Alert protocol message types	12	23
Message authentication	ad hoc	standard
Key material generation	ad hoc	PRF
CertificateVerify	complex	simple
Finished	ad hoc	PRF
Baseline cipher suites	includes Fortezza	no Fortezza

HTTPS

Hyper Text Transfer Protocol (HTTP) protocol is used for web browsing. The function of HTTPS is similar to HTTP. The only difference is that HTTPS provides “secure” web browsing. HTTPS stands for HTTP over SSL. This protocol is used to provide the encrypted and authenticated connection between the client web browser and the website server.



The secure browsing through HTTPS ensures that the following content are encrypted −

- URL of the requested web page.
- Web page contents provided by the server to the user client.
- Contents of forms filled in by user.
- Cookies established in both directions.

Working of HTTPS

HTTPS application protocol typically uses one of two popular transport layer security protocols - SSL or TLS. The process of secure browsing is described in the following points.

- You request a HTTPS connection to a webpage by entering https:// followed by URL in the browser address bar.
- Web browser initiates a connection to the web server. Use of https invokes the use of SSL protocol.
- An application, brows0065r in this case, uses the system port 443 instead of port 80 (used in case of http).
- The SSL protocol goes through a handshake protocol for establishing a secure session as discussed in earlier sections.
- The website initially sends its SSL Digital certificate to your browser. On verification of certificate, the SSL handshake progresses to exchange the shared secrets for the session.
- When a trusted SSL Digital Certificate is used by the server, users get to see a padlock icon in the browser address bar. When an Extended Validation Certificate is installed on a website, the address bar turns green.

- Once established, this session consists of many secure connections between the web server and the browser.

Use of HTTPS

- Use of HTTPS provides confidentiality, server authentication and message integrity to the user. It enables safe conduct of e-commerce on the Internet.
- Prevents data from eavesdropping and denies identity theft which is common attacks on HTTP.
- Present day web browsers and web servers are equipped with HTTPS support. The use of HTTPS over HTTP, however, requires more computing power at the client and the server end to carry out encryption and SSL handshake.

SECURE SHELL PROTOCOL (SSH):

The salient features of SSH are as follows −

- SSH is a network protocol that runs on top of the TCP/IP layer. It is designed to replace the TELNET which provided unsecure means of remote logon facility.
- SSH provides a secure client/server communication and can be used for tasks such as file transfer and e-mail.
- SSH2 is a prevalent protocol which provides improved network communication security over earlier version SSH1.

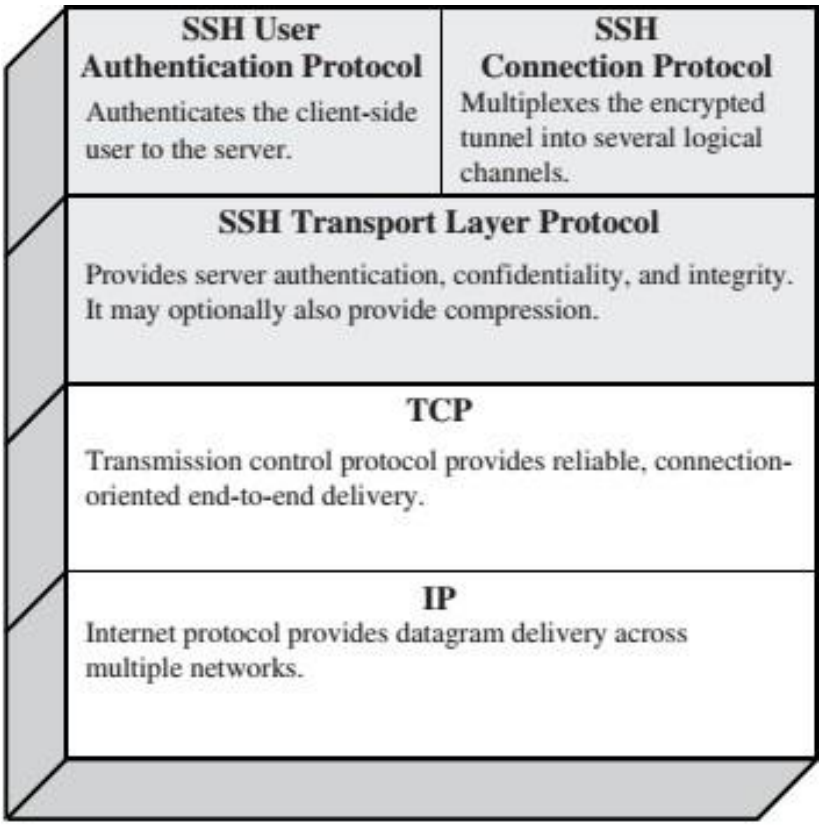


Figure 16.8 SSH Protocol Stack

SSH Services

SSH provides three main services that enable provision of many secure solutions. These services are briefly described as follows −

- **Secure Command-Shell (Remote Logon)** − It allows the user to edit files, view the contents of directories, and access applications on connected device. Systems administrators can remotely start/view/stop services and processes, create user accounts, and change file/directories permissions and so on. All tasks that are feasible at a machine's command prompt can now be performed securely from the remote machine using secure remote logon.

- **Secure File Transfer** − SSH File Transfer Protocol (SFTP) is designed as an extension for SSH-2 for secure file transfer. In essence, it is a separate protocol layered over the Secure Shell protocol to handle file transfers. SFTP encrypts both the username/password and the file data being transferred. It uses the same port as the Secure Shell server, i.e. system port no 22.
- **Port Forwarding (Tunneling)** − It allows data from unsecured TCP/IP based applications to be secured. After port forwarding has been set up, Secure Shell reroutes traffic from a program (usually a client) and sends it across the encrypted tunnel to the program on the other side (usually a server). Multiple applications can transmit data over a single multiplexed secure channel, eliminating the need to open many ports on a firewall or router.

Firewall

Firewall Design Principles

Firewall*: A firewall forms a barrier through which the traffic going in each direction must pass. A firewall security policy dictates which traffic is authorized to pass in each direction.*

A firewall may be designed to operate as a filter at the level of IP packets, or may operate at a higher protocol layer.

Design goals for a firewall:

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible, as explained later in this section.
2. Only authorized traffic, as defined by the local security policy, will be allowed to pass.

Various types of firewalls are used, which implement various types of security policies, as explained later in this section.

3. The firewall itself is immune to penetration.

Firewall Characteristics:

Major characteristics related to firewall protection are described below.

1. Various protection levels
2. Wireless network (Wi-fi) Protection
3. Internet and network access
4. Blockage against unauthorized access
5. Protection against malware
6. Provide access only to valid data packets
7. Provision of different configurations
8. Provision of numerous security policies

9. Allowing to pass authorized traffic that fulfils a set of rules
10. Firewall functions like an immune system for malware and unauthorized access; therefore, it ensures a secure system and an OS.

Access Control

Access Control policies are just one part of the Firewall Threat Defense (FTD) feature set that organizations use to control network traffic. As packets ingress the firewall, many checks occur. For example, is the packet part of an existing connection, and does the packet require decryption or network address translation? Once the packet has had these checks applied, it passes into the Access Control Policy (ACP).

An ACP can be assigned to one or more managed devices. However, a device can only have one ACP deployed at one time. The benefit of assigning a single ACP to more than one device is that a single change to the policy via the FMC UI can quickly be applied to multiple devices, reducing operational overheads.

Network data that is processed by managed devices can be filtered and controlled by a set of rules based on:

- Simple, easily determined transport and network layer characteristics: source and destination, ports,
  protocols and applications
- The latest contextual information on the traffic, including characteristics such as reputation, risk,
  business relevance, the application used, or URL visited
- Realm, user, user group, or ISE attribute
- Custom Security Group Tag (SGT)
- Characteristics of encrypted traffic; you can also decrypt this traffic for further analysis
- Whether unencrypted or decrypted traffic contains a prohibited file, detected malware, or intrusion
  attempt
- Time and day

Techniques that firewalls use to control access and enforce the site's security policy.

1. **Service control:** Determines the types of Internet services that can be accessed, inbound or outbound.
2. **Direction control:** Determines the direction in which particular service requests may be

initiated and allowed to flow through the firewall.

3. **User control:** Controls access to a service according to which user is attempting to access it.

4. **Behavior control:** Controls how particular services are used.

Capabilities are within the scope of a firewall:

1. A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks

2. A firewall provides a location for monitoring security-related events. Audits and alarms can be implemented on the firewall system.

3. A firewall can serve as the platform for IPSec.

Firewalls limitations

1. The firewall cannot protect against attacks that bypass the firewall.

2. The firewall does not protect against internal threats, such as an employee who unwittingly cooperates with an external attacker.

3. The firewall cannot protect against the transfer of virus-infected programs or files.

Types of Firewalls

There are three common types of firewalls:

1. Packet filters,
2. Application-level gateways,
3. Circuit-level gateways.

1. Packet-Filtering Router:

A packet-filtering router applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet.

Filtering rules are based on information contained in a network packet:

Source IP address: The IP address of the system that originated the IP packet (e.g., 192.178.1.1)

Destination IP address: The IP address of the system the IP packet is trying to reach (e.g. 192.168.1.2)

Source and destination transport-level address: The transport level (e.g., TCP or UDP) port number, which defines applications such as SNMP or TELNET .

The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header.

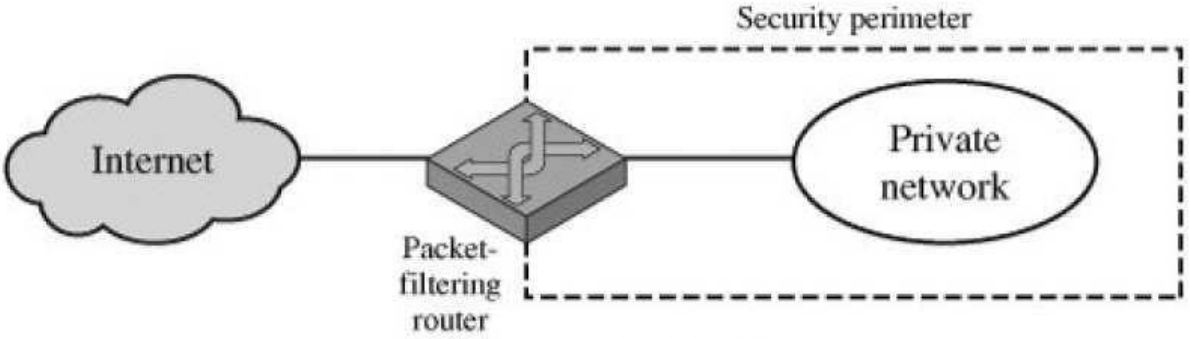
1. If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet.

2. If there is no match to any rule, then a default action is taken.

3. Two default policies are possible:

   a. Default = *discard*: That which is not expressly permitted is prohibited.

   b. Default = *forward*: That which is not expressly prohibited is permitted.



(a) Packet-filtering router

Example:

A	**action**	**ourhost**	**port**	**theirhost**	**port**	**comment**	
	block	*	*	SPIGOT	*	we don't trust these people	
	allow	OUR-GW	25	*	*	connection to our SMTP port	
B	**action**	**ourhost**	**port**	**theirhost**	**port**	**comment**	
	block	*	*	*	*	default	
C	**action**	**ourhost**	**port**	**theirhost**	**port**	**comment**	
	allow	*	*	*	25	connection to their SMTP port	
D	**action**	**src**	**port**	**dest**	**port**	**flags**	**comment**
	allow	{our hosts}	*	*	25		our packets to their SMTP port
	allow	*	25	*	*	ACK	their replies
	action	**src**	**port**	**dest**	**port**	**flags**	**comment**
	allow	{our hosts}	*	*	*		our outgoing calls
E	allow	*	*	*	*	ACK	replies to our calls
	allow	*	*	*	>1024		traffic to nonservers

Weaknesses of packet filter firewalls:

IP address spoofing: The intruder transmits packets from the outside with a source IP address field containing an address of an internal host. The attacker hopes that the use of a spoofed address will allow penetration of systems that employ simple source address security, in which packets from

specific trusted internal hosts are accepted.

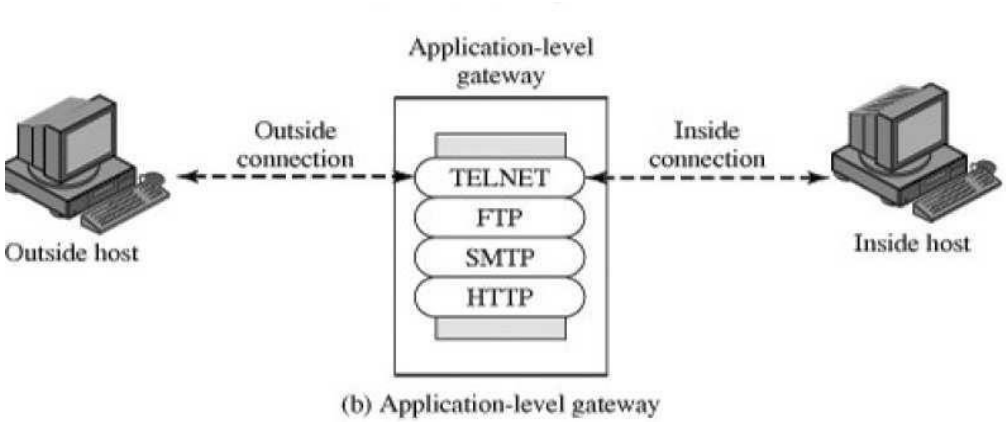
Stateful Firewall

A stateful inspection packet filter tightens up the rules for TCP traffic by creating a directory of outbound TCP connections

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.9.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established

2. Application-Level Gateway:

- Application-level gateways tend to be more secure than packet filters
- An application-level gateway, also called a proxy server, acts as a relay of application-level traffic.
- The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed.
- When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host containing the application data between the two endpoints.
- If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall.



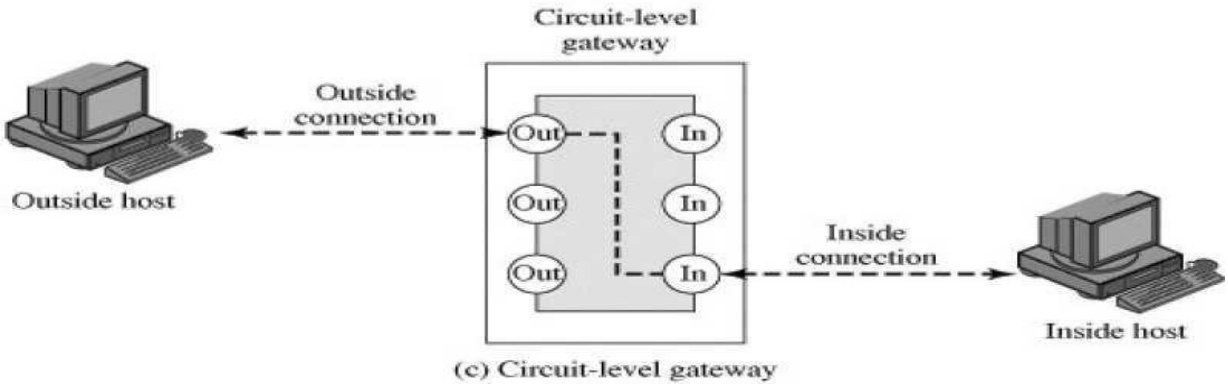
(b) Application-level gateway

3. Circuit-Level Gateway

- A typical use of circuit-level gateways is a situation in which the system administrator trusts the internal users.
- This can be a stand-alone system or it can be a specialized function performed by an application- level

gateway for certain applications.

- A circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host.
- Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents.
- The security function consists of determining which connections will be allowed.



(c) Circuit-level gateway

Bastion Host:

A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security. Typically, the bastion host serves as a platform for an application-level or circuit level gateway. Common characteristics of a bastion host include the following:

1. The bastion host hardware platform executes a secure version of its operating system, making ita trusted system.
2. Only the services that the network administrator considers essential are installed on the bastionhost. These include proxy applications such as Telnet, DNS, FTP, SMTP, and user authentication.
3. The bastion host may require additional authentication before a user is allowed access to theproxy services.

FIREWALL LOCATION AND CONFIGURATIONS

Firewalls are crucial components in network security, providing a barrier between a trusted internal network and untrusted external networks, such as the internet. Their location and configuration depend on the network architecture and security requirements. Here's a general overview:

Firewall Locations:

1. **Perimeter (Network Edge) Firewall:**
    - **Location:** Typically placed at the network perimeter between an organization's internal network and the internet.

- **Purpose:** Protects the entire internal network from external threats. It filters and monitors traffic entering and leaving the network.

2. **Internal Firewall:**
   - **Location:** Positioned within the internal network, dividing it into security zones.
   - **Purpose:** Provides additional segmentation and control within the internal network. It can restrict traffic between different parts of the internal network.
3. **Host-Based Firewall:**
   - **Location:** Installed on individual devices (e.g., computers, servers).
   - **Purpose:** Protects a specific device by monitoring and controlling incoming and outgoing network traffic based on an organization's configured security rules.

Firewall Configuration:

1. **Default Deny Rule:**
   - **Configuration:** Set a default rule to deny all traffic unless explicitly allowed. This ensures that only necessary and authorized traffic is permitted.
2. **Access Control Lists (ACLs):**
   - **Configuration:** Define ACLs based on IP addresses, ports, and protocols to control the flow of traffic. ACLs specify what traffic is allowed or denied.
3. **Stateful Inspection:**
   - **Configuration:** Enable stateful inspection to track the state of active connections. This allows the firewall to make context-aware decisions based on the current state of a connection.
4. **Proxy Services:**
   - **Configuration:** Use proxy services to inspect and filter application-layer traffic, providing an additional layer of security by analyzing the content of the traffic.
5. **Intrusion Detection and Prevention Systems (IDPS):**
   - **Configuration:** Integrate intrusion detection and prevention capabilities into the firewall to identify and block malicious activity.
6. **Virtual LANs (VLANs):**
   - **Configuration:** Utilize VLANs for network segmentation, and configure the firewall to control traffic between VLANs. This helps contain the impact of security incidents.
7. **Network Address Translation (NAT):**
   - **Configuration:** Implement NAT to hide internal IP addresses from external networks, enhancing security by obfuscating the internal network structure.
8. **Logging and Monitoring:**
   - **Configuration:** Enable logging for firewall events and regularly review logs for security incidents. Monitoring tools can provide real-time insights into network activity.
9. **Regular Updates:**
   - **Configuration:** Keep firewall firmware, software, and rule sets up to date to address security vulnerabilities and ensure optimal performance.
10. **Remote Access Policies:**
    - **Configuration:** If applicable, configure remote access policies to control and secure remote connections, such as Virtual Private Network (VPN) settings.
11. **User Authentication:**
    - **Configuration:** Implement user authentication to control access based on user credentials. This adds an extra layer of security beyond IP-based controls.

12. **Testing and Auditing:**
    - **Configuration:** Regularly test the firewall configuration and conduct security audits to identify and address potential vulnerabilities.
13. **Emergency Response Plan:**
    - **Configuration:** Have a plan in place to quickly reconfigure the firewall in response to security incidents. This may involve blocking specific IP addresses, adjusting rule sets, or implementing emergency measures.

Firewall configuration is highly dependent on the specific needs and risks of an organization. Regularly reviewing and updating the configuration based on evolving threats and network changes is essential for maintaining effective security.