Firewalls, Tunnels, and Network Intrusion Detection
Firewalls

• A **firewall** is an integrated collection of security measures designed to prevent unauthorized electronic access to a networked computer system.

• A network firewall is similar to firewalls in building construction, because in both cases they are intended to isolate one "network" or "compartment" from another.
Firewall Policies

- To protect private networks and individual machines from the dangers of the greater Internet, a firewall can be employed to filter incoming or outgoing traffic based on a predefined set of rules called **firewall policies**.
Policy Actions

- Packets flowing through a firewall can have one of three outcomes:
  - **Accepted**: permitted through the firewall
  - **Dropped**: not allowed through with no indication of failure
  - **Rejected**: not allowed through, accompanied by an attempt to inform the source that the packet was rejected

- Policies used by the firewall to handle packets are based on several properties of the packets being inspected, including the protocol used, such as:
  - TCP or UDP
  - the source and destination IP addresses
  - the source and destination ports
  - the application-level payload of the packet (e.g., whether it contains a virus).
Blacklists and White Lists

• There are two fundamental approaches to creating firewall policies (or rulesets) to effectively minimize vulnerability to the outside world while maintaining the desired functionality for the machines in the trusted internal network (or individual computer).

• **Blacklist** approach
  – All packets are allowed through except those that fit the rules defined specifically in a blacklist.
  – This type of configuration is more flexible in ensuring that service to the internal network is not disrupted by the firewall, but is naïve from a security perspective in that it assumes the network administrator can enumerate all of the properties of malicious traffic.

• **Whitelist** approach
  – A safer approach to defining a firewall ruleset is the default-deny policy, in which packets are dropped or rejected unless they are specifically allowed by the firewall.
Firewall Types

• **packet filters (stateless)**
  – If a packet matches the packet filter's set of rules, the packet filter will drop or accept it

• **"stateful" filters**
  – It maintains records of all connections passing through it and can determine if a packet is either the start of a new connection, a part of an existing connection, or is an invalid packet.

• **application layer**
  – It works like a **proxy** it can “understand” certain applications and protocols.
  – It may inspect the contents of the traffic, blocking what it views as inappropriate content (i.e. websites, viruses, vulnerabilities, ...)
Stateless Firewalls

- A stateless firewall doesn’t maintain any remembered context (or “state”) with respect to the packets it is processing. Instead, it treats each packet attempting to travel through it in isolation without considering packets that it has processed previously.

![Diagram of network traffic and firewall rules]

- Allow outbound SYN packets, destination port=80
- Allow inbound SYN-ACK packets, source port=80
Stateless Restrictions

- Stateless firewalls may have to be fairly restrictive in order to prevent most attacks.

**Diagram:**
- Client connected to a trusted internal network.
- Attacker not trusted.
- **Firewall:**
  - Allow outbound SYN packets, destination port=80.
  - Drop inbound SYN packets.
  - Allow inbound SYN-ACK packets, source port=80.
Statefull Firewalls

- **Stateful firewalls** can tell when packets are part of legitimate sessions originating within a trusted network.

- **Stateful firewalls** maintain tables containing information on each active connection, including the IP addresses, ports, and sequence numbers of packets.

- Using these tables, stateful firewalls can allow only inbound TCP packets that are in response to a connection initiated from within the internal network.
Statefull Firewall Example

• Allow only requested TCP connections:

- Allow outbound TCP sessions, destination port=80

**Established TCP session:**
(128.34.78.55, 76.120.54.101)

**Firewall state table**
Tunnels

• The contents of TCP packets are not normally encrypted, so if someone is eavesdropping on a TCP connection, he can often see the complete contents of the payloads in this session.

• One way to prevent such eavesdropping without changing the software performing the communication is to use a tunneling protocol.

• In such a protocol, the communication between a client and server is automatically encrypted, so that useful eavesdropping is infeasible.
Tunneling Prevents Eavesdropping

- Packets sent over the Internet are automatically encrypted.

**Tunneling protocol**
(does end-to-end encryption and decryption)

Payloads are encrypted here

Untrusted Internet

Payloads are encrypted here
Secure Shell (SSH)

A secure interactive command session:

1. The client connects to the server via a TCP session.

2. The client and server exchange information on administrative details, such as supported encryption methods and their protocol version, each choosing a set of protocols that the other supports.

3. The client and server initiate a secret-key exchange to establish a shared secret session key, which is used to encrypt their communication (but not for authentication).

4. This session key is used in conjunction with a chosen block cipher (typically AES, 3DES) to encrypt all further communications.
SSH

1- Client and server establish a TCP session

2- Client and server exchange information (e.g. encryption, protocol version)

3- Secret key exchange protocol and establish a shared key

4- negotiate for authentication type

4.1 Client send a public key

4.2 server send a challenge $E_{P_k}(RN)$

4.3 client decrypt the challenge $D_{P_r}(E_{P_k}(RN))$

4.4 client send the challenge (RN)

Encrypted data (e.g. 3DES or AES)
SSH

4. The server sends the client a list of acceptable forms of authentication, which the client will try in sequence. The most common mechanism is to use a password or the following public-key authentication method:

   a) If public-key authentication is the selected mechanism, the client sends the server its public key.

   b) The server then checks if this key is stored in its list of authorized keys. If so, the server encrypts a challenge using the client’s public key and sends it to the client.

   c) The client decrypts the challenge with its private key and responds to the server, proving its identity.

5. Once authentication has been successfully completed, the server lets the client access appropriate resources, such as a command prompt.
IPSec

- IPSec defines a set of protocols to provide confidentiality and authenticity for IP packets.
- Each protocol can operate in one of two modes, **transport mode** or **tunnel mode**.
  - In **transport mode**, additional IPsec header information is inserted before the data of the original packet, and only the payload of the packet is encrypted or authenticated.
  - In **tunnel mode**, a new packet is constructed with IPsec header information, and the entire original packet, including its header, is encapsulated as the payload of the new packet.
IPSEC

- The two parties must first set up a set of security associations (SAs)
  - SAs contains encryption keys and encryption algorithms, and other parameters

1- Client and server established SAs

<table>
<thead>
<tr>
<th>SPI1_in+IP</th>
<th>Inbound SADB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(SPI1)_{in}+IP</td>
</tr>
<tr>
<td></td>
<td>(SPI2)_{in}+IP</td>
</tr>
<tr>
<td></td>
<td>(SPI3)_{in}+IP</td>
</tr>
<tr>
<td></td>
<td>(SPI4)_{in}+IP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPI1_out+IP</th>
<th>Outbound SADB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(SPI1)_{out}+IP</td>
</tr>
<tr>
<td></td>
<td>(SPI2)_{out}+IP</td>
</tr>
<tr>
<td></td>
<td>(SPI3)_{out}+IP</td>
</tr>
<tr>
<td></td>
<td>(SPI4)_{out}+IP</td>
</tr>
</tbody>
</table>
Internet Key Exchange (IKE)

- IPSEC use the IKE protocol to handle the negotiation of SAs
- SAs includes hash function, encryption algorithm, and authentication methods

1- client and server execute the IKE
2- client and server established a shared key
3- client and server exchanged encrypted $E_k$ (SAs)
IPSEC - The Authentication Header (AH)

- The Authentication header protocol is used to authenticate the IP header, and provide data integrity.
- Data is not encrypted

![Diagram of IPSEC Authentication Header](image)

**Figure 6.15:** The authentication header.
Components of the Authentication Header

• **AH contains SPI** a randomly initialized sequence number

• AH contains Integrity check value (**ICV**)
  – ICV is computed using MAC based applying **SHA-256** hash function on packet information (IP header and payload) that do not change during routing.

• **AH do not support NAT**. Because the source IP address is part of AH header.
The Encapsulating security protocol (ESP)

- ESP provide **data confidentiality** by encrypting the packet payload
- ESP encapsulate its payload by adding a **header** and a **trailer**
- **ESP use block cipher** (AES, 3DES, or Blowfish)
- **ESP authenticate the payload and ESP header** and not the IP header
## ESP header

<table>
<thead>
<tr>
<th>Bits 0-7</th>
<th>Bits 8-15</th>
<th>Bits 16-23</th>
<th>Bits 24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Security Parameters Index (SPI)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sequence Number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Padding)</td>
<td>Pad Length</td>
<td>Next Header</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Authentication Data</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 6.16**: The ESP header.
Virtual Private Networking (VPN)

- Virtual private networking (VPN) allows private networks to be safely extended over long physical distances by making use of a public network.
- VPN provides guarantees of data confidentiality, integrity, and authentication, despite the use of an untrusted network for transmission.
- There are two primary types of VPNs, remote access VPN and site-to-site VPN.
Types of VPNs

• **Remote access** VPNs allow authorized clients to access a private network that is referred to as an **intranet**.
  – To accomplish this, the organization sets up a VPN endpoint, known as a **network access server, or NAS**.
  – **Point-to-point tunneling protocol (PPTP)**
  – **Layer 2 tunneling protocol (L2TP)**
VPN (site-to-site)

- **Site-to-site** VPN solutions are designed to provide a secure bridge between two or more physically distant networks.
  - Before VPN, organizations wishing to safely bridge their private networks purchased expensive leased lines to directly connect their intranets with cabling.
Risk from allowing tunneling

- VPN can be used to bypass firewall polices
- VPN can be used for information leakage attacks
- It can be used as wrapper for forbidden protocol
- An insider can establish a tunnel to external VPN server and use the external VPN server to route traffic to the target.