Overview of Firewalls
Internet Security Mechanisms

- Prevent: Firewall
- Detect: Intrusion Detection
- Survive/Response: Recovery, Forensics

- Goal: prevent if possible, detect quickly otherwise, confine the damage

Firewalls

- Provides secure connectivity between networks
- Implements and enforces a security policy for communication between networks

Firewalls (Cont’d)

- Many organizations have distinct needs
  - Access by anyone to public data concerning the company
  - Access only by employees to internal data
- Solution: inner and outer (DMZ) networks
Firewall Capabilities

- Controlled access
  - restrict incoming and outgoing traffic according to security policy
- Other functions
  - log traffic, for later analysis
  - network address translation (why is this useful?)
  - encryption / decryption
  - application (payload) transformations

Limitations of Firewalls

- Cannot protect against traffic that does not cross it
  - i.e., there may be other ingress points to the network, such as modems or wireless access points, that bypass the firewall
  - doesn’t protect against “inside” attacks
- Configuration of firewalls to accomplish a desired high-level security policy is non-trivial

Filtering and Proxy Firewalls
Filtering

- Compare traffic to patterns, then process traffic according to rules if matched
- Two styles
  - packet filtering
  - session filtering

Packet Filtering

- Patterns specify values in the header of a single packet, e.g.,
  - source IP address and port number
  - destination IP address and port number
  - transport protocol type

Packet Filtering (cont’d)

- Decisions made on a per-packet basis
  - no state information (about previous packets) is maintained or used
- Assessment
  - easy to implement
  - but limited capabilities
- May be subject to tiny-fragment attack
  - first fragment has only a few bytes
  - rest of TCP header in a second fragment, not examined by firewall
Session Filtering

• Packet decisions are made in the context of a connection or flow of packets
• If packet is the start of a new connection…
  – check against rules for new connections
• If packet is part of an existing connection…
  – check against state-based rules for existing connections
  – update state of this connection

Session Filtering (cont’d)

• Assessment
  – more powerful than packet filtering, can recognize more sophisticated threats or implement more complex policies
  – also more expensive to implement

Application: Telnet

• Client opens channel to server; tells server its port number. The ACK bit is not set when initiating the connection but will be set on the remaining packets.
• Server acknowledges.
Example: Firewall Access for Telnet

**Format:**
```
access-list <rule number> <permit|deny> <protocol> <SOURCE host with IP address| any|IP address and mask> [gt|eq port number] <DEST host with IP address| any|IP address and mask> [gt|eq port number]
```

Note: any packets not explicitly permitted in an access list assumed to be denied or dropped.

The following allows user to telnet from an IP address (172.168.10.11) to any destination, but not vice-versa:

```
access-list 100 permit tcp host 172.168.10.11 gt 1023 any eq 23
! Allows packets out to remote Telnet servers.
access-list 101 permit tcp any eq 23 host 172.168.10.11 established
! Allows returning packets to come back in. It verifies that the ACK bit is set.
```

```
interface Ethernet 0
access-list 100 out   ! Apply the first rule to outbound traffic.
access-list 101 in     ! Apply the second rule to inbound traffic.
```

Note: any packets not explicitly permitted in an access list assumed to be denied or dropped.

---

Example: Firewall Access for FTP

```
allow a user to FTP (not passive FTP) from any IP address to the FTP server (172.168.10.12):
```

```
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 21
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 20
! Allows packets from any client to the FTP control and data ports.
access-list 101 permit tcp host 172.168.10.12 eq 21 any gt 1023
access-list 101 permit tcp host 172.168.10.12 eq 20 any gt 1023
! Allows FTP server to send packets back to any IP address with TCP ports > 1023.
```

```
interface Ethernet 0
access-list 100 in     ! Apply the first rule to inbound traffic.
access-list 101 out    ! Apply the second rule to outbound traffic.
```

---
Proxy Firewalls

- Serve as *relays* for connections
- Two flavors
  1. application level
  2. circuit level

```
Applications
Presentations
Sessions
Transport
DataLink
Physical
Network
```

Application Proxies

- Understand specific application protocols, e.g., HTTP, SMTP, Telnet
  - proxy ‘impersonates’ both one side of connection to the other
- Can do arbitrary processing / inspection of application payloads
  - ex.: check mail for viruses before forwarding
- Computationally expensive
- Must write a new proxy application to support new protocols

Application Proxies (cont’d)

- May require hosts inside the organization to be configured to use the proxy
Circuit-Level Proxies

- Sets up two connections, one to inside user, one to outside server
  - i.e., proxy at the TCP level, rather than the application level
  - client programs must be aware they are using a circuit-level proxy, by linking to modified libraries
- Users must authenticate to proxy before connection to outside will be established
- Example protocol: SOCKS

Comparison

<table>
<thead>
<tr>
<th>Type of Service</th>
<th>Security</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Filtering</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Session Filtering</td>
<td></td>
<td>Dependent on vendor for dynamic support</td>
</tr>
<tr>
<td>Circuit-level Gateway</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Application-level Gateway</td>
<td></td>
<td>Limited number of simultaneous connections</td>
</tr>
</tbody>
</table>

Comparison (Cont’d)

<table>
<thead>
<tr>
<th>Modify Client Applications?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Filter</td>
</tr>
<tr>
<td>Session Filter</td>
</tr>
<tr>
<td>Circuit-level Gateway</td>
</tr>
<tr>
<td>Application-level Gateway</td>
</tr>
</tbody>
</table>