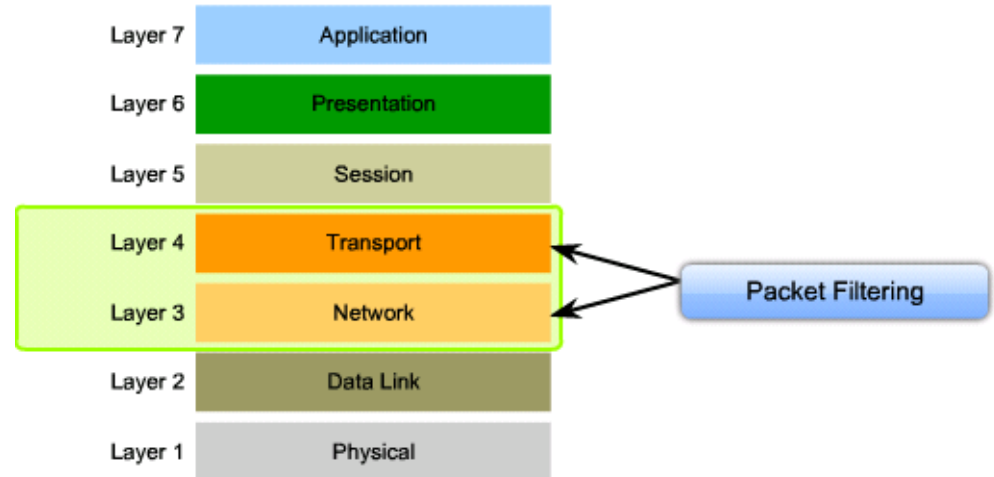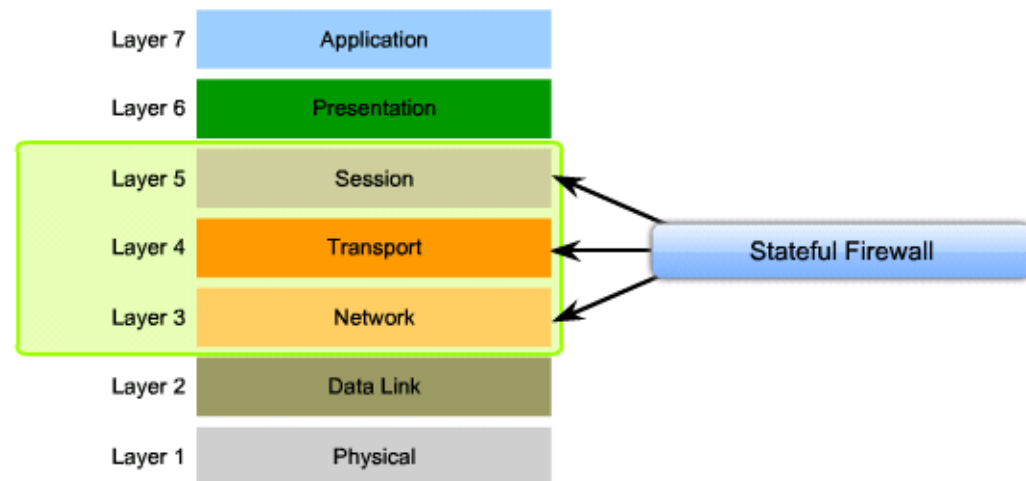# Implementing Firewall Technologies

Firewalls

- Network firewalls separate protected from non-protected areas preventing unauthorized users from accessing protected network resources.

- Technologies used:
  - ACLs
    - Standard, extended, numbered and named ACLs
  - Advanced ACLs
    - Stateful firewall - ACLs with the `established` keyword
    - Reflexive (dynamic) ACLs, timed-based ACLs
  - Zone-Based Firewall Feature

Common Types of Firewalls

- Packet-filtering firewall

Layer 7	Application
Layer 6	Presentation
Layer 5	Session
Layer 4	Transport
Layer 3	Network
Layer 2	Data Link
Layer 1	Physical

Packet Filtering (Layer 4 Transport, Layer 3 Network)

- Stateful firewall

Layer 7	Application
Layer 6	Presentation
Layer 5	Session
Layer 4	Transport
Layer 3	Network
Layer 2	Data Link
Layer 1	Physical

Stateful Firewall (Layer 5 Session, Layer 4 Transport, Layer 3 Network)

Standard ACLs

Types of Cisco ACLs

Standard ACLs filter IP packets based on the [   ]

```
access-list 10 permit 192.168.30.0 0.0.0.255
```

```
access-list {1-99} {permit | deny} source-addr [source-wildcard]
```

- **Note:**
  - Can be applied in an incoming or outgoing direction on an interface using the `ip access-group` command.
  - It can also be applied on a VTY port using the `access-class` command.

Standard ACLs

- Create a standard named ACL on R1 called **RESTRICT-VTY** that permits Telnet access to only the administrative host.

```
R1(config)# ip access-list standard RESTRICT-VTY
R1(config-std-nacl)# remark Permit only Admin host
R1(config-std-nacl)# permit host 192.168.1.10
R1(config-std-nacl)# exit
R1(config)# line vty 0 4
R1(config-line)# access-class RESTRICT-VTY
R1(config-line)# exit
```


Administrator
192.168.1.10

Fa0/0

R1

Internet

User
192.168.1.100

Workgroup Server
192.168.1.6
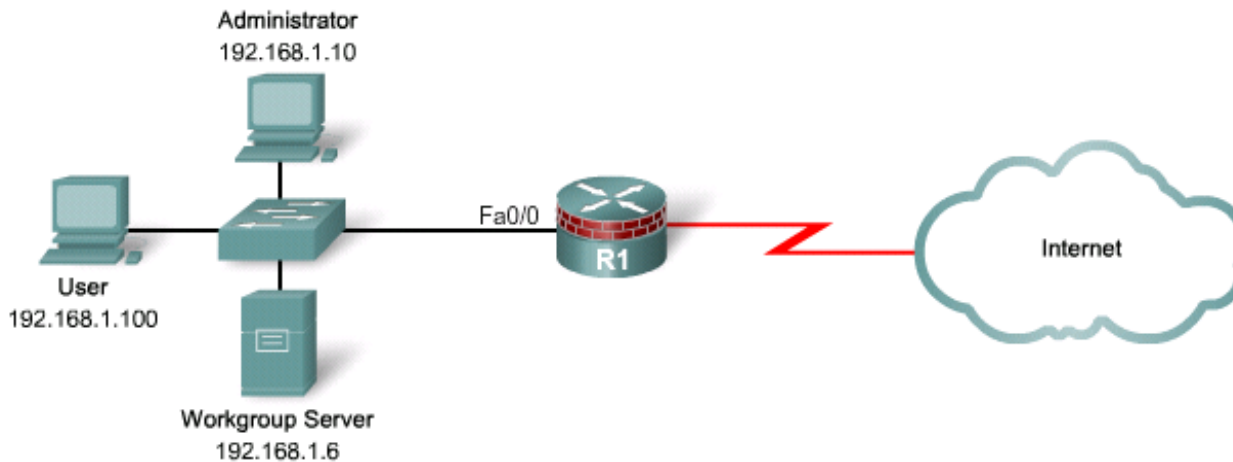
Extended ACLs

Extended ACLs filter IP packets based on several attributes, including the following:
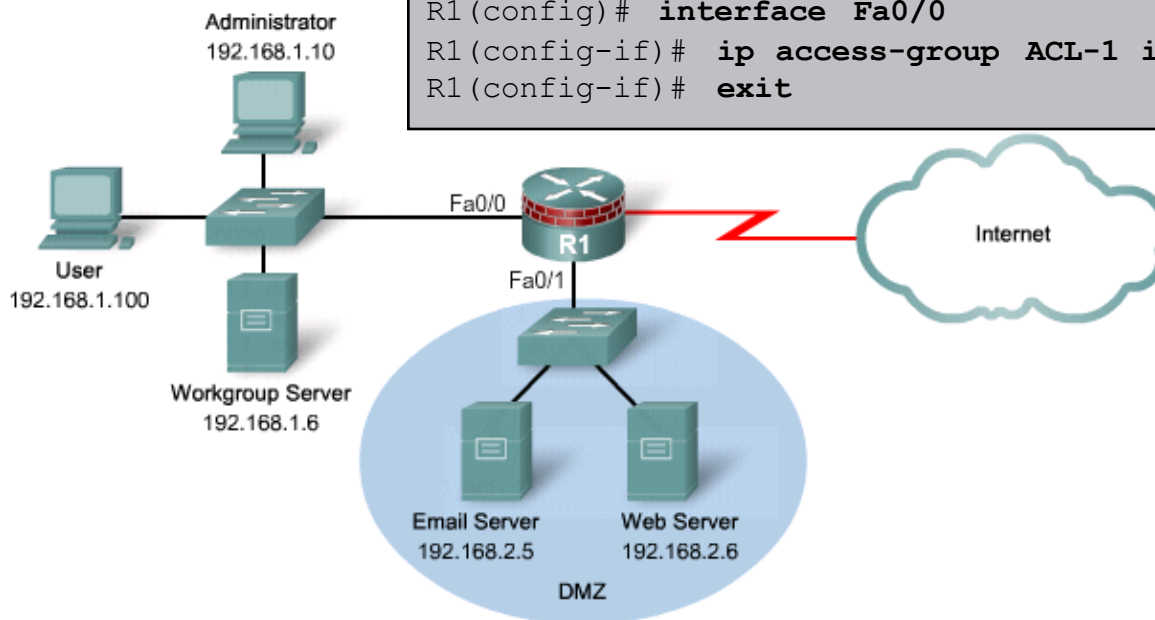
-
-
-

```
access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
```

```
access-list {100-199} {permit | deny} protocol source-addr
[source-wildcard][operator operand] destination-addr [destination-
wildcard] [operator operand][established]
```

Extended ACLs - 1

- Create an extended named ACL called **ACL-1**, applied incoming on the Fa0/0 interface, that denies the workgroup server outside access but permits the remainder of the LAN users outside access using the **established** keyword.
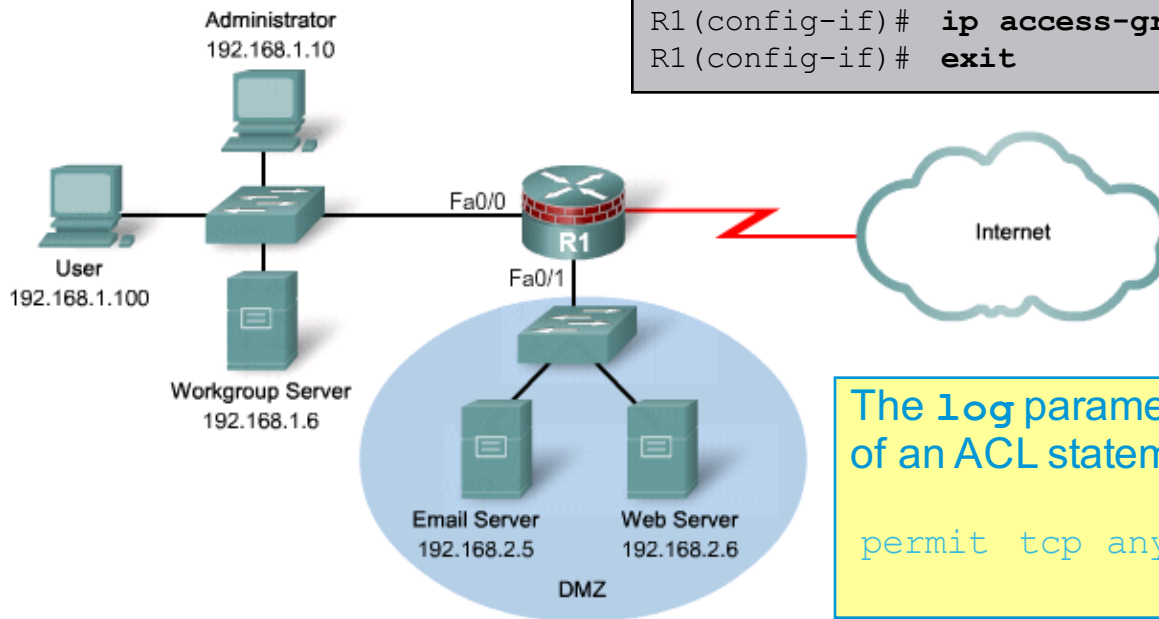
```
R1(config)# ip access-list extended ACL-1
R1(config-ext-nacl)# remark LAN ACL
R1(config-ext-nacl)# deny ip host 192.168.1.6 any
R1(config-ext-nacl)# permit tcp 192.168.1.0 0.0.0.255 any
established
R1(config-ext-nacl)# deny ip any any
R1(config-ext-nacl)# exit
R1(config)# interface Fa0/0
R1(config-if)# ip access-group ACL-1 in
R1(config-if)# exit
```


Administrator
192.168.1.10

User
192.168.1.100

Workgroup Server
192.168.1.6

Fa0/0

R1

Fa0/1

Internet

Email Server
192.168.2.5

Web Server
192.168.2.6

DMZ

Extended ACLs - 2

- Create an extended named ACL called **ACL-2**, applied outgoing on the Fa0/1 DMZ interface, permitting access to the specified Web and Email servers.

```
R1(config)# ip access-list extended ACL-2
R1(config-ext-nacl)# remark DMZ ACL
R1(config-ext-nacl)# permit tcp any host 192.168.2.5 eq 25
R1(config-ext-nacl)# permit tcp any host 192.168.2.6 eq 80
R1(config-ext-nacl)# deny ip any any
R1(config-ext-nacl)# interface Fa0/1
R1(config-if)# ip access-group ACL-2 out
R1(config-if)# exit
```

Administrator
192.168.1.10

Fa0/0

R1

Internet

User
192.168.1.100

Fa0/1

Workgroup Server
192.168.1.6

Email Server
192.168.2.5

Web Server
192.168.2.6

DMZ

The `log` parameter can be appended to the end of an ACL statement.

`permit tcp any host 192.168.2.6 eq 80 log`

Logging

- When configured, the IOS software compares packets and finds a match to the statement.

- The router then logs it to any enabled logging facility, such as:
  - the console
  - the internal buffer
  - syslog server
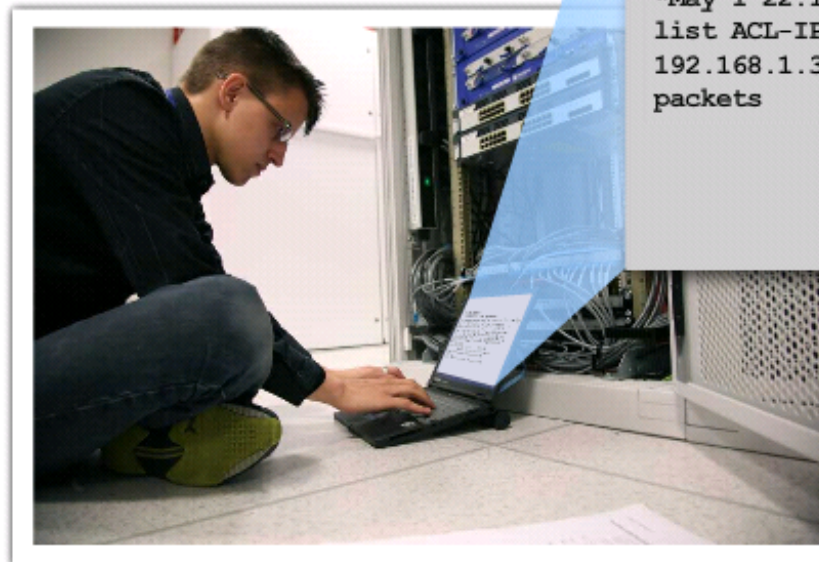
```
*May 1 22:12:13.243: %SEC-6-IPACCESSLOGP:
list ACL-IPv4-E0/0-IN permitted tcp
192.168.1.3(1024) -> 192.168.2.1(22), 1
packet

*May 1 22:17:16.647: %SEC-6-IPACCESSLOGP:
list ACL-IPv4-E0/0-IN permitted tcp
192.168.1.3(1024) -> 192.168.2.1(22), 9
packets
```
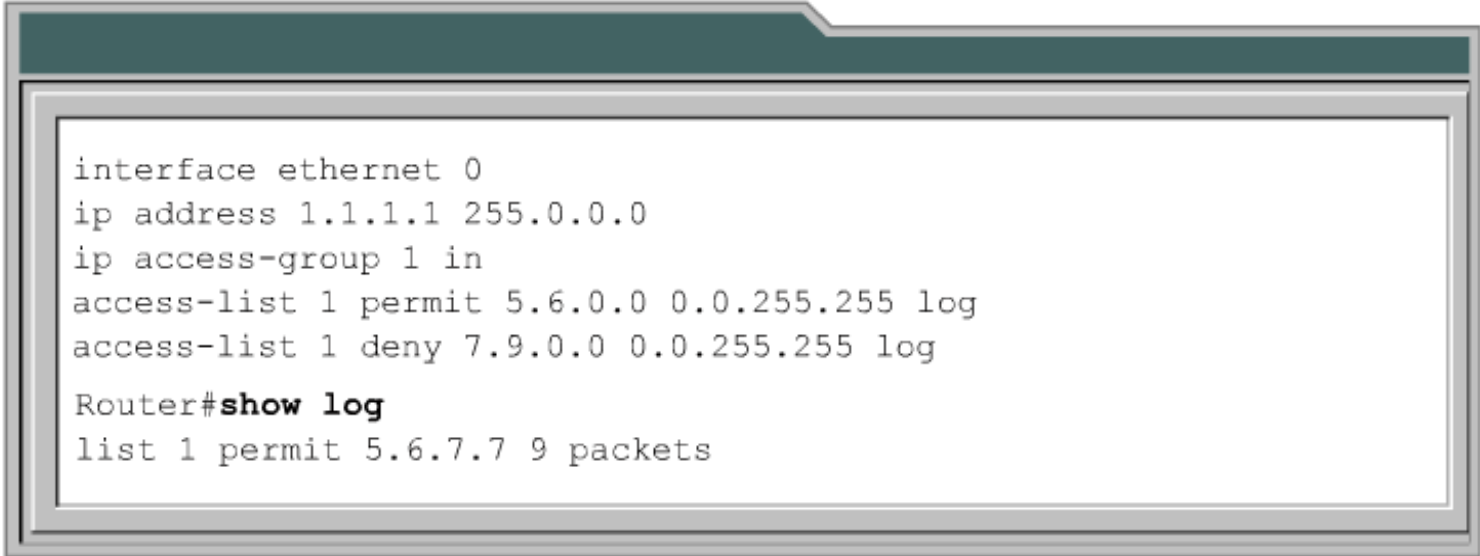
Logging

- Several pieces of information are logged:
  - Action - permit or deny
  - Protocol - TCP, UDP, or ICMP
  - Source and destination addresses
  - For TCP and UDP - source and destination port numbers
  - For ICMP - message types

- Log messages are ***processed switched*** on the first packet match and then at five minute intervals after that first packet match.

View ACL operation

- A useful command for viewing access list operation is the **`show log`** command.

- To reset counters, use the **`clear ip access-list counter`** **[***number* **|** *name***]** command.

```
interface ethernet 0
ip address 1.1.1.1 255.0.0.0
ip access-group 1 in
access-list 1 permit 5.6.0.0 0.0.255.255 log
access-list 1 deny 7.9.0.0 0.0.255.255 log

Router#show log
list 1 permit 5.6.7.7 9 packets
```

ACL Caveats

- Implicit deny all:
  - All Cisco ACLs end with an implicit "deny all" statement.

- Standard ACL packet filtering:
  - Standard ACLs are limited to packet filtering based on source addresses only.
  - Extended ACLs might need to be created to fully implement a security policy.

- Order of statements:
  - ACLs have a policy of first match; when a statement is matched, the list is no longer examined.
  - Ensure that statements at the top of the ACL do not negate any statements found lower.
  - Place specific ACL statements higher in the ACL and more general statements near the end.

ACL Caveats

- Directional filtering:
  - ACLs can be applied to inbound packets (toward the interface) or outbound packets (away from the interface).
  - Double-check the direction of data that an ACL is filtering.

- Special packets:
  - Router-generated packets, such as routing table updates, are not subject to outbound ACL statements on the source router.
  - If the security policy requires filtering these types of packets, inbound ACLs on adjacent routers or other router filter mechanism must be used.

- Modifying ACLs:
  - New entries are added to an ACL, are always added to the bottom.
  - Starting with Cisco IOS 12.3, sequence numbers can be used to edit an ACL.
  - The ACL is processed top-down based on the sequence numbers of the statements (lowest to highest).

ACL Sequence Numbers

- The default behavior when adding a statement to an ACL is that the statement is added to the end. Without sequence numbers the only way to add a statement between existing entries was to delete the ACL and recreate it.

- Likewise, the only way to delete an entry was to delete the entire ACL and recreate it.

- IP access list sequence numbers allow you to selectively remove a statement from an existing ACL or to add a new statement at any position within the ACL.

- This feature is not available on old-style numbered access lists, which existed before named access lists. Keep in mind that you can name an access list with a number, so numbers are allowed when they are entered in the standard or extended named access list configuration mode using **the ip access-list {standard | extended}** *access-list-name* command.

Modify an ACL using Sequence Numbers

- First use the **show** command to view the existing sequence numbers.

```
R1# show access-list 150
Extended IP acess list 150
    10 permit tcp any any eq www
    20 permit tcp any any eq telnet
    30 permit tcp any any eq smtp
    40 permit tcp any any eq pop3
    50 permit tcp any any eq 21
    60 permit tcp any any eq 20
```
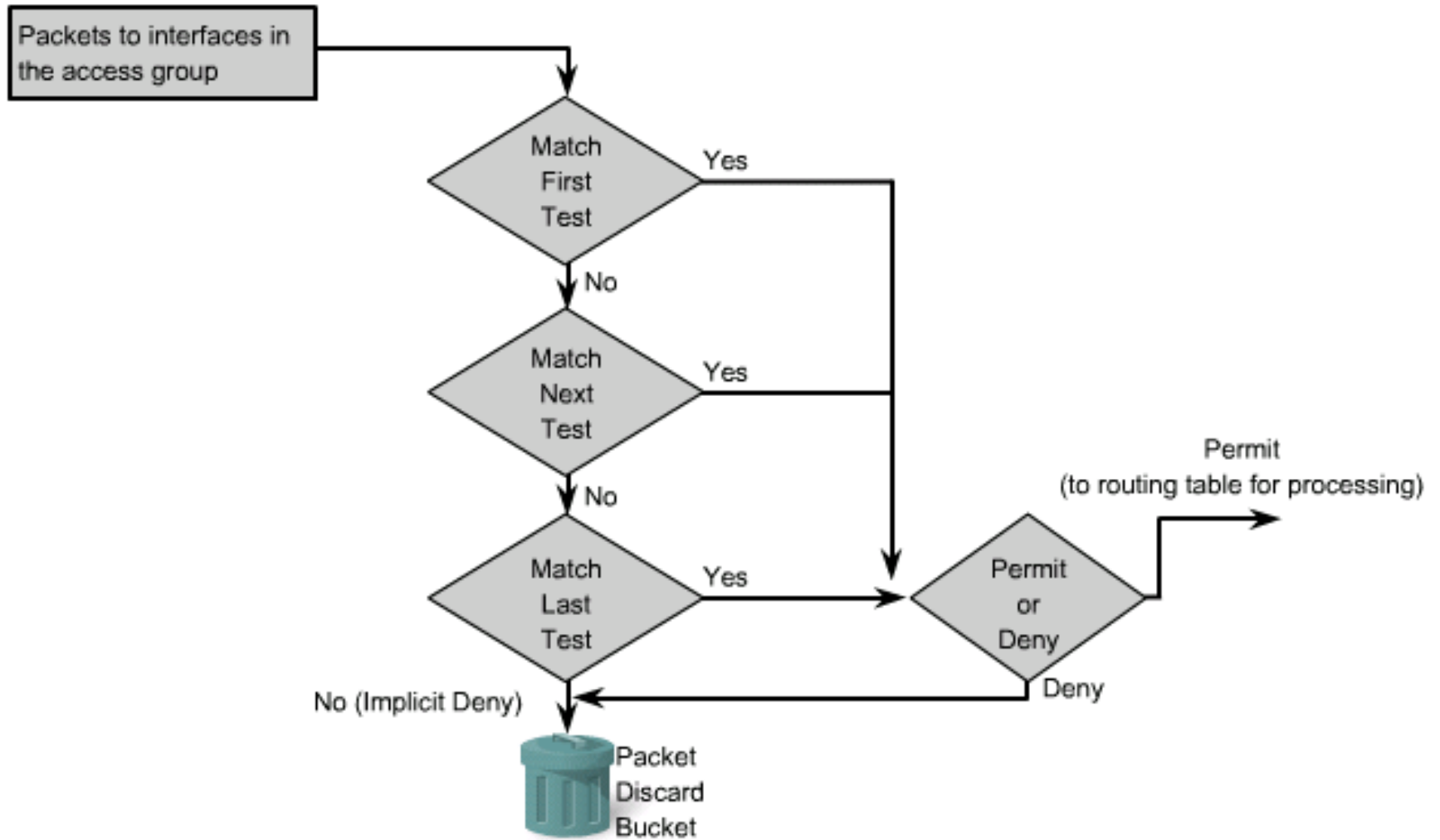
- Resequence if necessary.

- Use the **no** *sequence-number* command to delete a statement.

```
R1(config)# ip access-list extended 150
R1(config-ext-nacl)# no 20
```
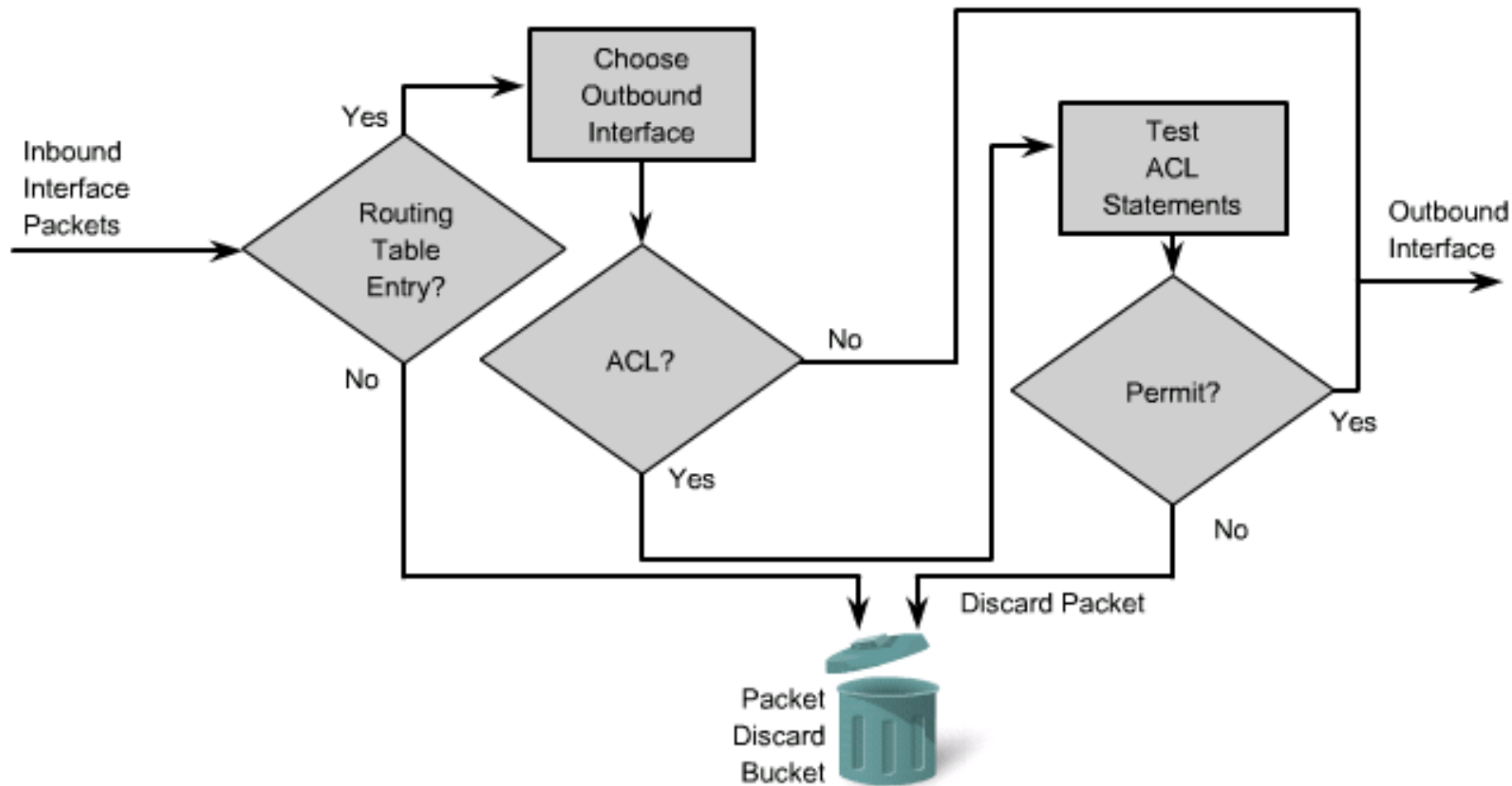
- Use the *sequence-number* **{permit | deny}** command to add a statement within the ACL.

```
R1(config)# ip access-list extended 150
R1(config-ext-nacl)# 20 permit tcp host 192.168.1.100  any eq telnet
```

Inbound ACL Operation Flow

Outbound ACL Operation Flow

Inbound Interface Packets → Routing Table Entry?
- Yes → Choose Outbound Interface → ACL?
  - No → Test ACL Statements → Permit?
    - Yes → Outbound Interface
    - No → Discard Packet
  - Yes → Packet Discard Bucket
- No → Packet Discard Bucket

ACL Placement

- Standard ACL placement:
    - Standard ACLs are placed **as close to the destination as possible**.
    - Standard ACLs filter packets based on the source address only so placing these ACLs too close to the source can adversely affect packets by denying all traffic, including valid traffic.

- Extended ACL placement:
    - Extended ACLs are placed on routers **as close to the source as possible** that is being filtered.
    - Placing Extended ACLs too far from the source is inefficient use of network resources because packets can be sent a long way only to be dropped or denied.
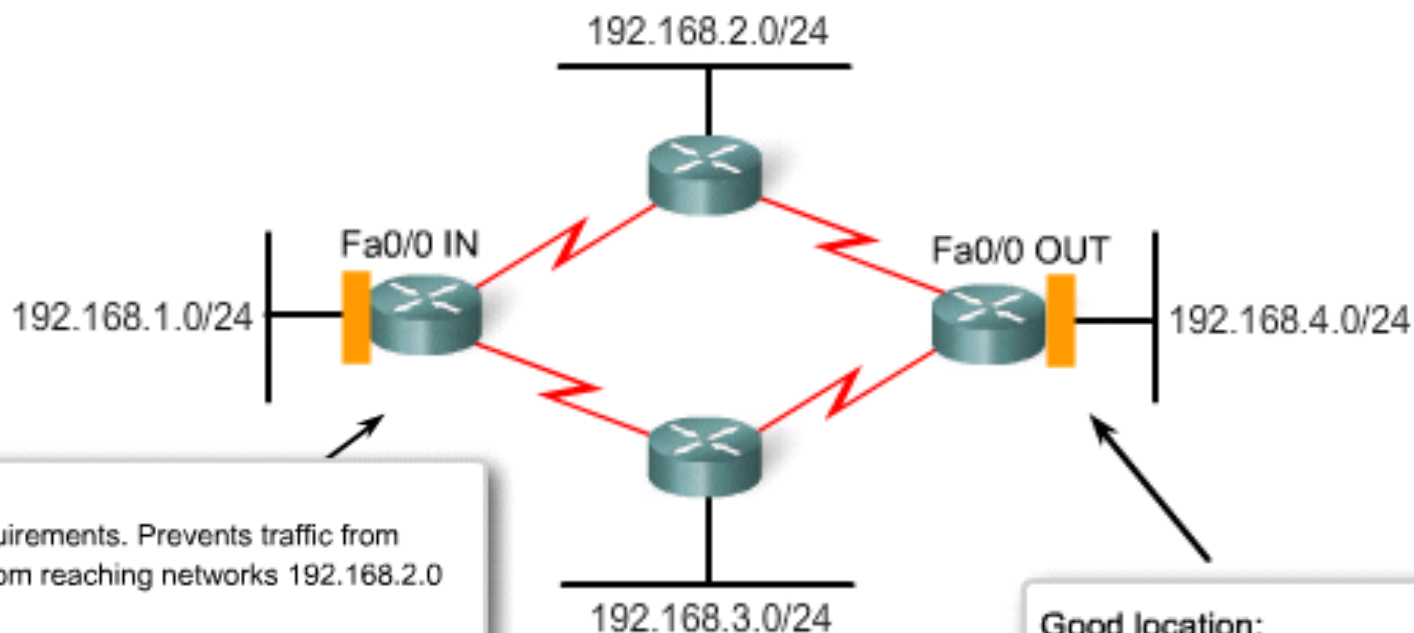
Where to place a Standard ACL?

Requirements:
Prevent traffic from the 192.168.1.0 network from entering the 192.168.4.0 network. Allow 192.168.1.0 to reach other networks.

```
ACL
access-list 9 deny 192.168.1.0 0.0.0.255
access-list 9 permit any
```

192.168.2.0/24

Fa0/0 IN

192.168.1.0/24

Fa0/0 OUT

192.168.4.0/24

Bad Location:
Meets some of the requirements. Prevents traffic from 192.168.1.0 network from reaching networks 192.168.2.0 and 192.168.3.0.

192.168.3.0/24

Good location:
Meets all requirements.

Where to place a Extended ACL?

Requirements:
Use Extended ACL to prevent traffic from the 192.168.1.0 network from entering the 192.168.4.0 network but allow it to reach other networks.

```
ACL
access-list 109 deny ip 192.168.1.0 0.0.0.255 192.168.4.0 0.0.0.255
access-list 109 permit ip any any
```

192.168.2.0/24

Fa0/0 IN   Fa0/0 OUT

192.168.1.0/24   192.168.4.0/24
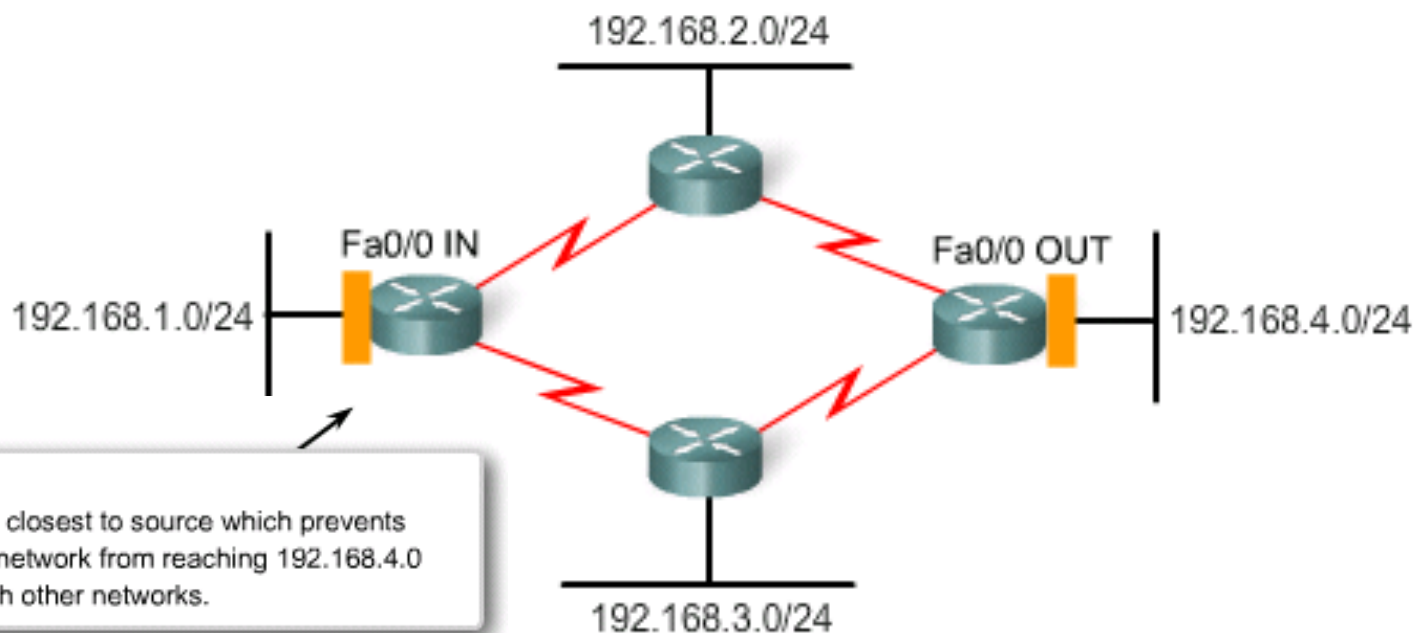
Good location:
Extended ACL is placed closest to source which prevents traffic from 192.168.1.0 network from reaching 192.168.4.0 but also allows it to reach other networks.

192.168.3.0/24

TCP Sessions

- In a modern network all traffic from the outside should be blocked from entering the inside unless:
  - It is explicitly permitted by an ACL.
  - It is returning traffic initiated from the inside of the network.

- Many common applications rely on TCP, which builds a virtual circuit between two endpoints.

- Traffic filtering solutions based on the two way connectivity of TCP were introduced:
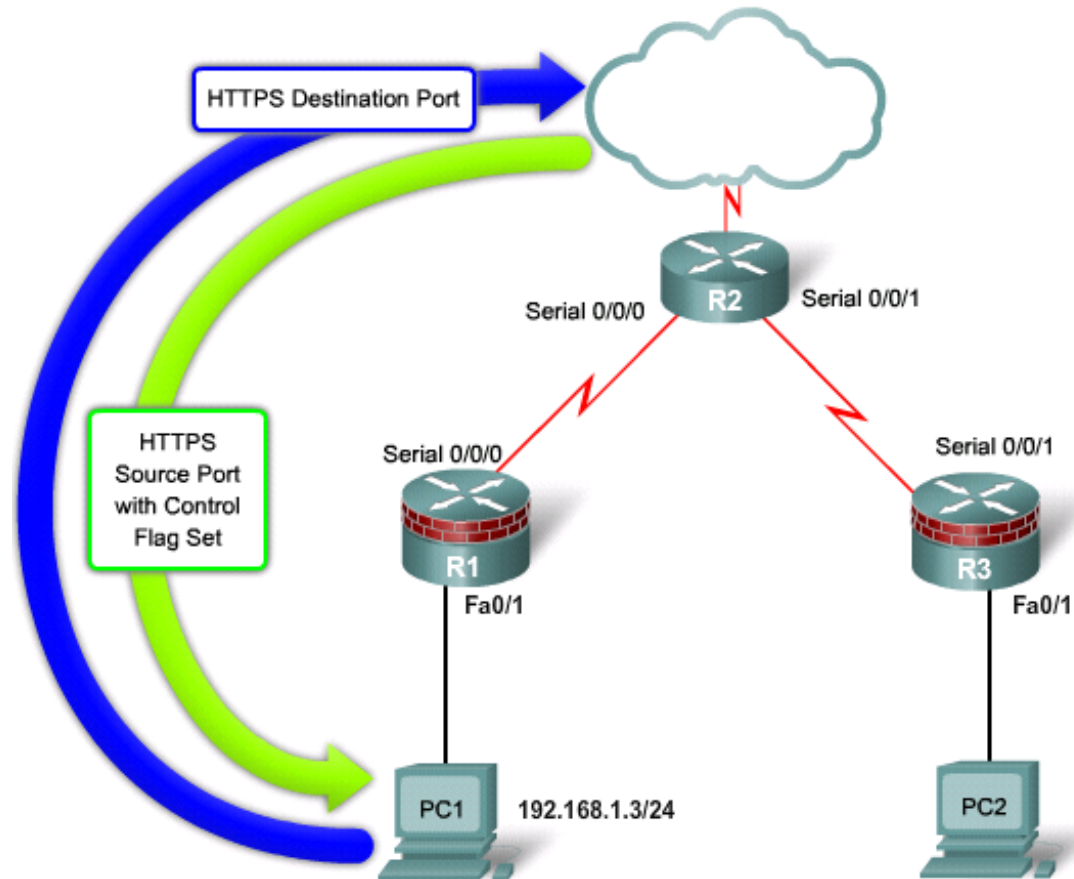 - TCP Established
  - Reflexive ACLs

TCP Established ACLs

- In 1995, the first generation IOS traffic filtering solution based on the TCP `established` keyword for extended IP ACLs.

  – The TCP `established` keyword blocks all traffic coming from the Internet except for the TCP reply traffic associated with established TCP traffic initiated from the inside of the network.

- The `established` keyword forces the router to check whether the TCP ACK or RST control flag is set.

  – If the ACK flag is set, the TCP traffic is allowed in.

  – If not, it is assumed that the traffic is associated with a new connection initiated from the outside.

TCP Established ACLs

- Using the **`established`** keyword does not implement a stateful firewall on a router.

  - The **`established`** parameter allows any TCP segments with the appropriate control flag.

  - No stateful information is maintained to keep track of traffic initiated from the inside of the network since the router does not keep track of conversations to determine whether the traffic is return traffic associated with a connection initiated from inside the network.

TCP Established ACLs


```
R1(config)#  access-list 100 permit tcp any eq 443 192.168.1.0  0.0.0.255 established
R1(config)#  access-list 100 deny ip any any
R1(config)#  interface s0/0/0
R1(config-if)#  ip access-group 100 in
```

Dynamic ACLs

- Dynamic ACLs are also called lock-and-key ACLs.

- Dynamic ACLs authenticate the user and then permits limited access through your firewall router for a host or subnet for a finite period.

- Dynamic ACLs are dependent on:
  - Telnet connectivity
  - Authentication (local or remote)
  - Extended ACLs

Implementing Dynamic ACLs

- An extended ACL is applied to block all traffic through the router except Telnet.

  – Users who want to traverse the router are blocked by the ACL until they use Telnet to connect to the router and are authenticated.

- Users authenticate using Telnet, and then dropped.

  – However, a single-entry dynamic ACL is added to the extended ACL that exists.

  – This permits traffic for a particular period; idle and absolute timeouts are possible.

Configuring Dynamic ACLs

When to Use Dynamic ACLs

- When you want a specific remote user or group of remote users to access a host within your network, connecting from their remote hosts via the Internet.

- When you want a subset of hosts on a local network to access a host on a remote network that is protected by a firewall.

Dynamic ACL Example

1. PC1 uses Telnet to connect R3 and authenticate.

2. Within a given time allowance, PC1 can use FTP and/or HTTP to connect to R3.
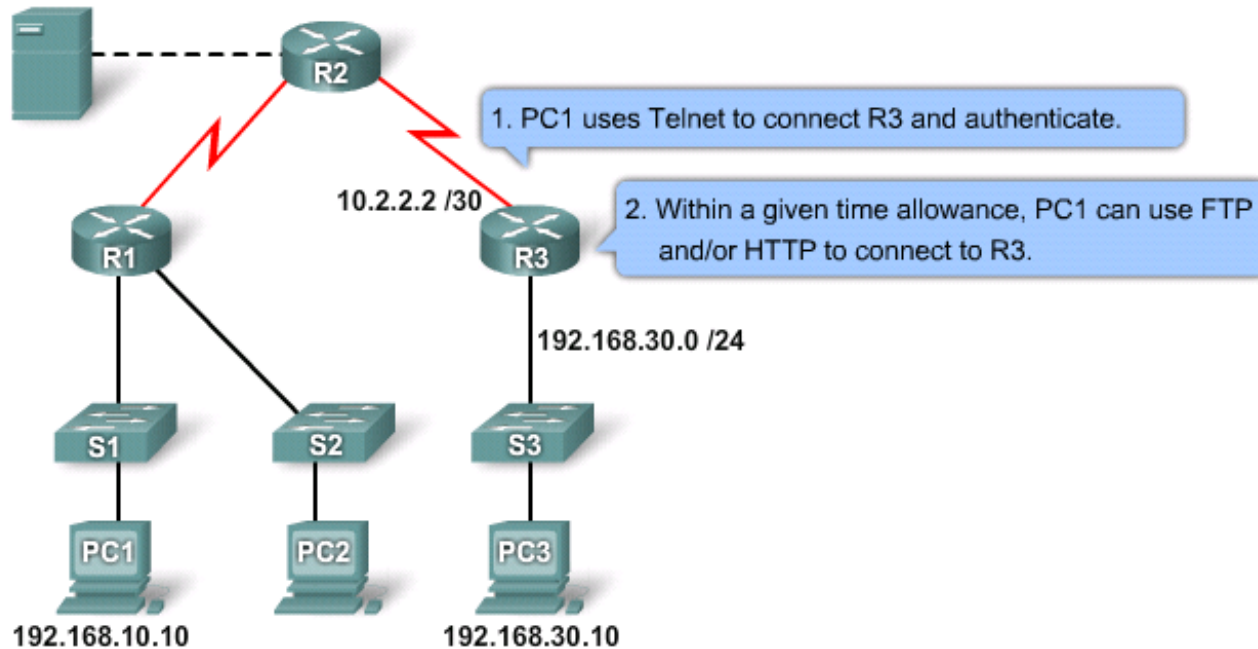
```
R3(config)# username Student password cisco
R3(config)# access-list 101 permit tcp any host 10.2.2.2 eq telnet
R3(config)# access-list 101 dynamic TESTLIST timeout 15 permit ip 192.168.10.0 0.0.0.255
192.168.30.0 0.0.0.255
R3(config)# interface s0/0/1
R3(config-if)# ip access-group 101 in
R3(config-if)# exit
R3(config)# line vty 0 4
R3(config-line)# login local
R3(config-line)# autocommand access-enable host timeout 15
```

Time-based ACLs

- Time-based ACLs allow for access control based on time.

Time-based ACLs:
Allow for access control based on the time of day and week
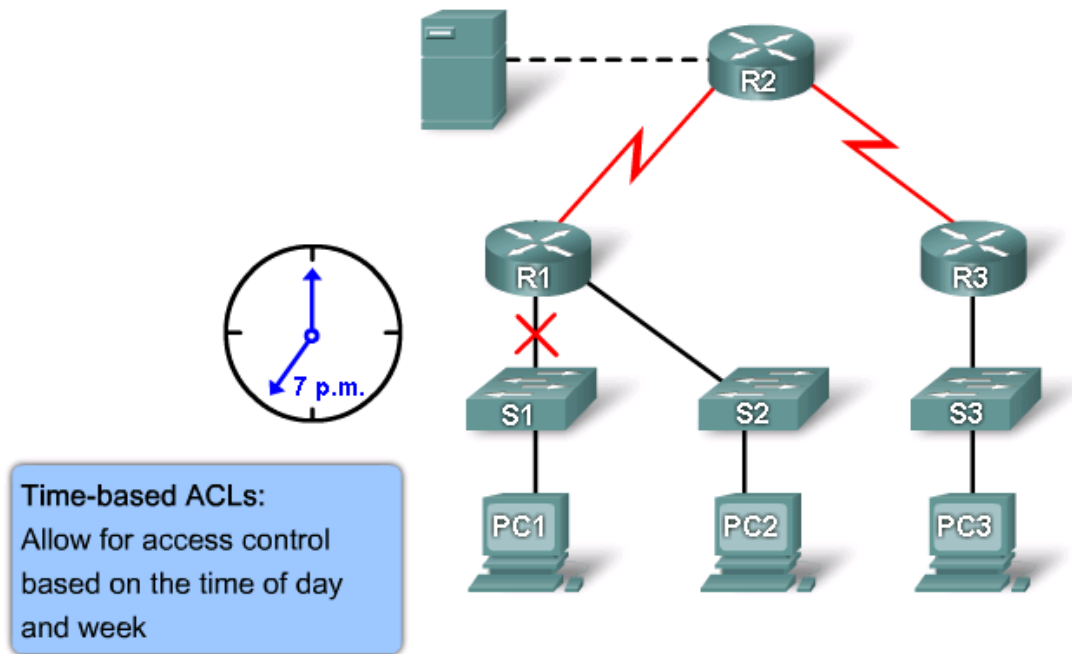
Time-based ACLs

- To implement time-based ACLs:
  - Create a time range that defines specific times of the day and week.
  - Identify the time range with a name and then refer to it by a function.
  - The time restrictions are imposed on the function itself.

| Step 1	`R1(config)#time-range EVERYOTHERDAY` `R1(config-time-range)#periodic Monday Wednesday Friday 8:00 to 17:00`
Step 2	`R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq telnet time-range EVERYOTHERDAY`
Step 3	`R1(config)#interface s0/0/0` `R1(config-if)#ip access-group 101 out`

Time-based ACL Example

- Users are not allowed to access the Internet during business hours, except during lunch and after hours between 5 p.m. and 7 p.m.

> I can't surf the web at 10:00 A.M. because of the time-based ACL!
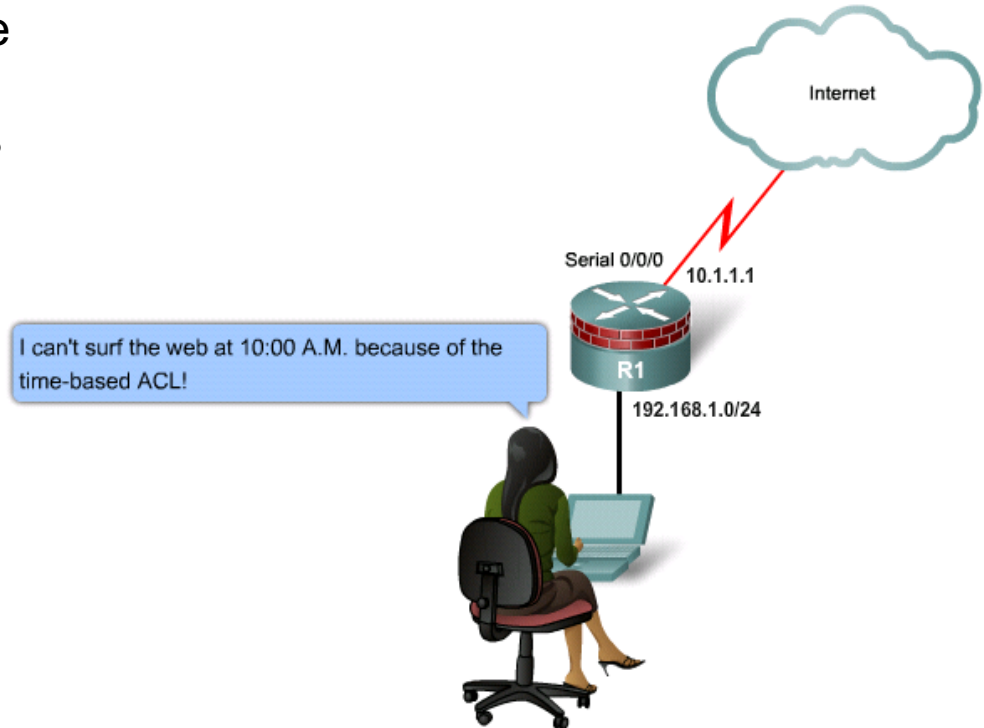
```
R1(config)# time-range EMPLOYEE-TIME
R1(config-time-range)# periodic weekdays 12:00 to 13:00
R1(config-time-range)# periodic weekdays 17:00 to 19:00
R1(config-time-range)# exit
R1(config)# access-list 100 permit ip 192.168.1.0 0.0.0.255 any time-range EMPLOYEE-TIME
R1(config)# access-list 100 deny ip any any
R1(config)# interface FastEthernet 0/1
R1(config-if)# ip access-group 100 in
R1(config-if)# exit
```

ACL Troubleshooting Commands

- Two commands are very useful for troubleshooting ACLs:
  - **show access-lists**
  - **debug ip packet (detail)**
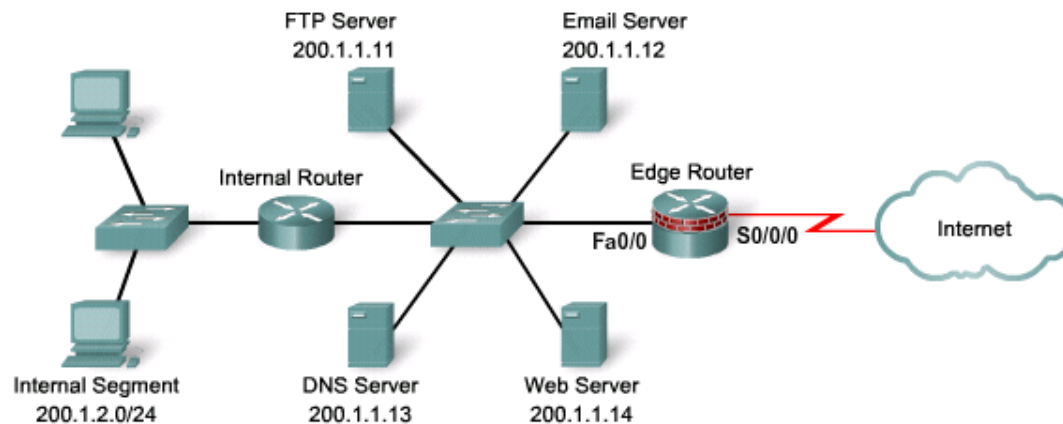

```
Perimeter# show access-list 100
Extended IP access list 100
  permit tcp any host 200.1.1.14 eq www   (189 matches)
  permit udp any host 200.1.1.13 eq domain (32 matches)
  permit tcp any host 200.1.1.12 eq smtp
  permit tcp any eq smtp host 200.1.1.12 established
  permit tcp any host 200.1.1.11 eq ftp
  permit tcp any host 200.1.1.11 eq ftp-data
  permit tcp any eq www 200.1.2.0 0.0.0.255 established
  permit udp any eq domain 200.1.2.0 0.0.0.255
  deny ip any any (1237 matches)
```
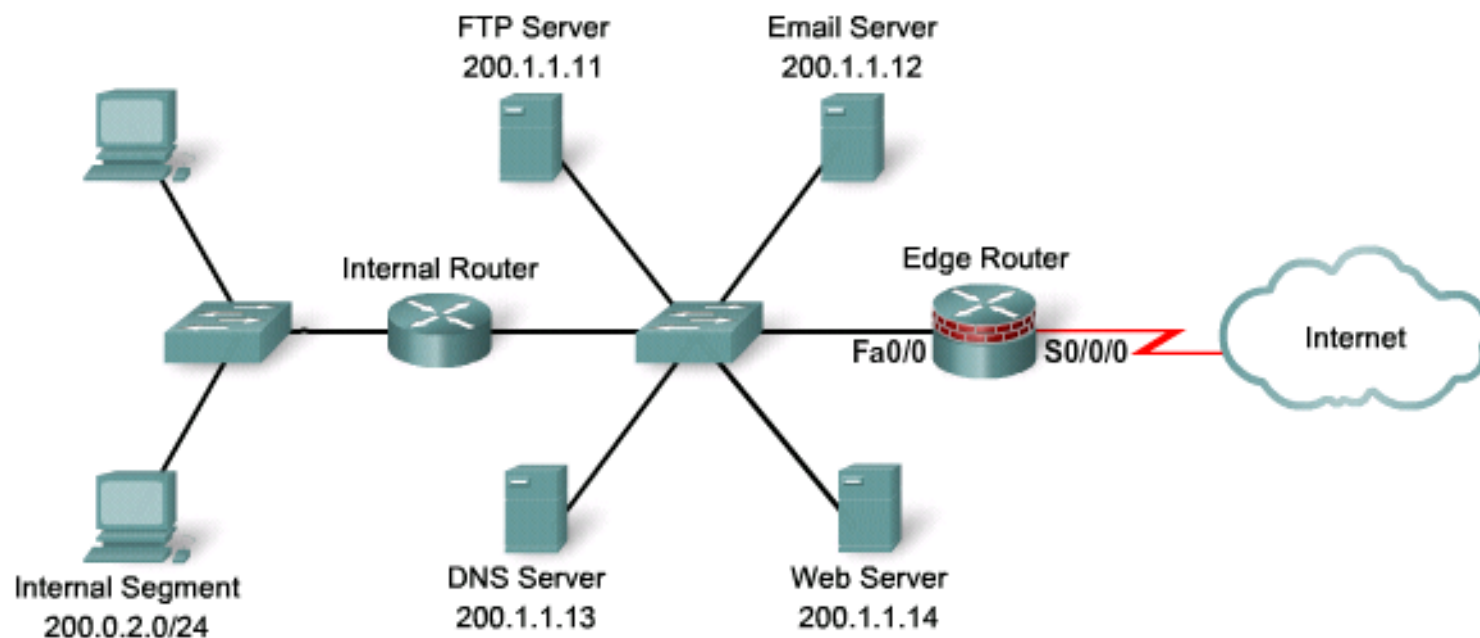
show access-lists
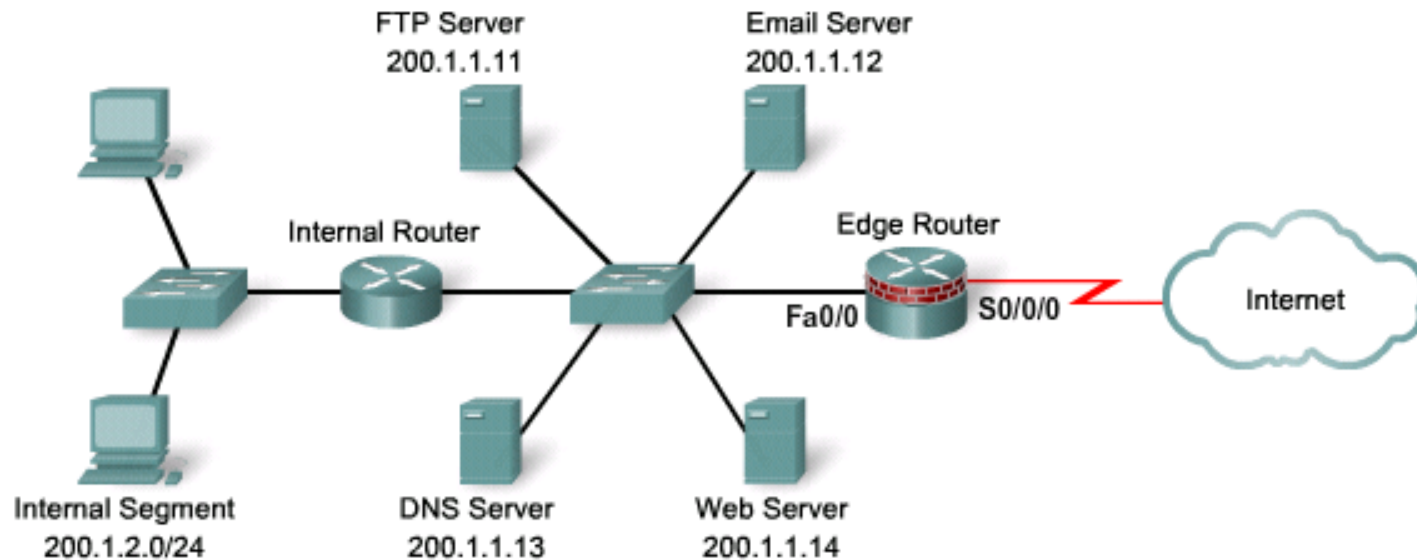

```
Perimeter# show access-list 100
Extended IP access list 100
  permit tcp any host 200.1.1.14 eq www   (189 matches)
  permit udp any host 200.1.1.13 eq domain (32 matches)
  permit tcp any host 200.1.1.12 eq smtp
  permit tcp any eq smtp host 200.1.1.12 established
  permit tcp any host 200.1.1.11 eq ftp
  permit tcp any host 200.1.1.11 eq ftp-data
  permit tcp any eq www 200.1.2.0 0.0.0.255 established
  permit udp any eq domain 200.1.2.0 0.0.0.255
  deny ip any any (1237 matches)
```

debug ip packet

FTP Server
200.1.1.11

Email Server
200.1.1.12

Internal Router

Edge Router

Fa0/0   S0/0/0

Internet

Internal Segment
200.0.2.0/24

DNS Server
200.1.1.13

Web Server
200.1.1.14

```
Edge# debug ip packet
IP packet debugging is on

IP: s-200.0.2.2 (FastEthernet0/0), d-172.69.2.42 (Serial0/0/0), g-172.69.16.2, forward
IP: s-200.0.2.2 (FastEthernet0/0), d-172.16.2.42 (Serial0/0/0), g-172.69.16.2, forward
IP: s-200.0.2.2 (FastEthernet0/0), d-172.69.43.126 (Serial0/0/0), g-172.69.16.2, forward
IP: s-200.0.2.2 (FastEthernet0/0), d-172.16.2.42 (Serial0/0/0), g-172.69.16.2, access denied
```

Mitigating Attacks with ACLs

- ACLs can be used to mitigate many network threats:
    - IP address spoofing, inbound and outbound
    - DoS TCP SYN attacks
    - DoS smurf attacks

- ACLs can also filter the following traffic:
    - ICMP messages, inbound and outbound
    - traceroute

Do Not Allow Addresses to be Spoofed

- Deny all IP packets containing the following IP addresses in their source field:

  - Any local host addresses (127.0.0.0/8)

  - Any reserved private addresses (RFC 1918)

  - Any addresses in the IP multicast address range (224.0.0.0/4)
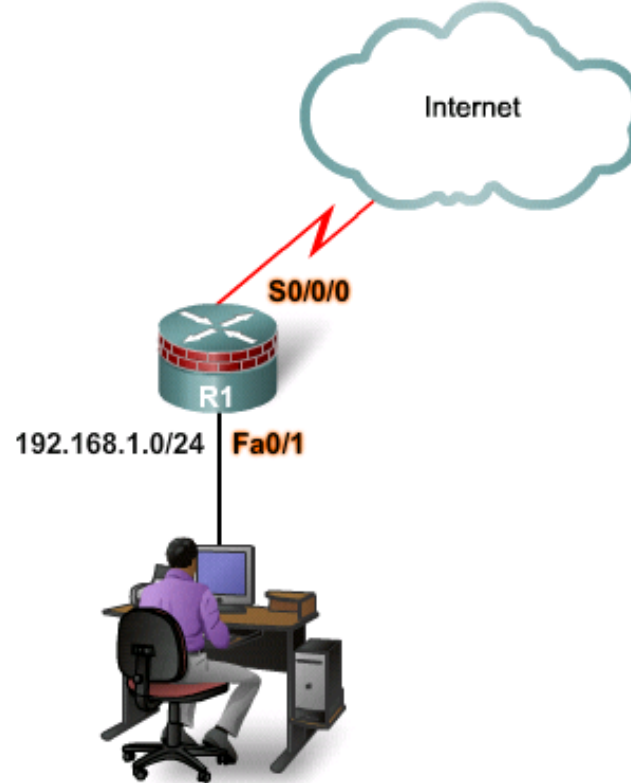
Inbound on S0/0/0

```
R1(config)#  access-list 150 deny ip 0.0.0.0 0.255.255.255  any
R1(config)#  access-list 150 deny ip 10.0.0.0 0.255.255.255  any
R1(config)#  access-list 150 deny ip 127.0.0.0 0.255.255.255  any
R1(config)#  access-list 150 deny ip 172.16.0.0 0.15.255.255  any
R1(config)#  access-list 150 deny ip 192.168.0.0 0.0.255.255  any
R1(config)#  access-list 150 deny ip 224.0.0.0 15.255.255.255  any
R1(config)#  access-list 150 deny ip host 255.255.255.255  any
```
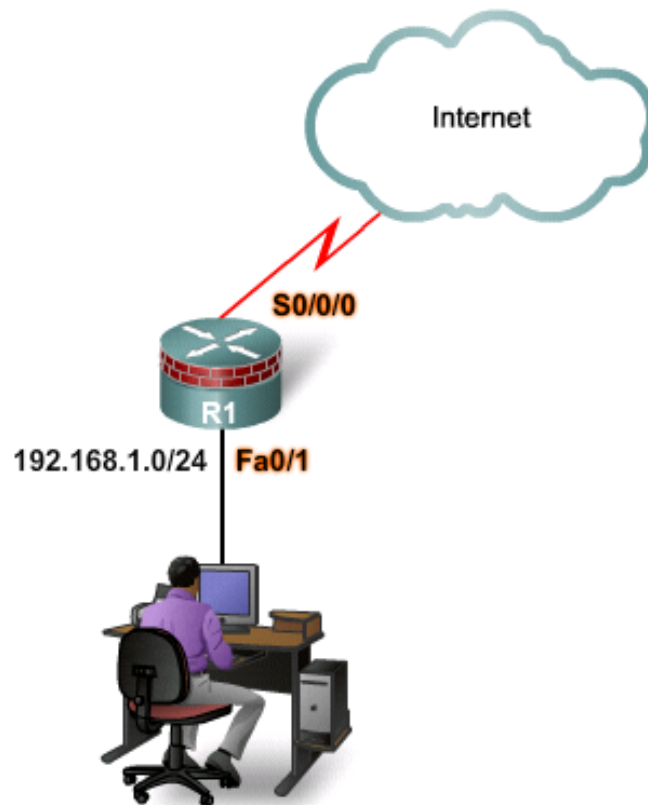
Do Not Allow Addresses to be Spoofed

- Do not allow any outbound IP packets with a source address other than a valid IP address of the internal network.

  - Create an ACL that permits only those packets that contain source addresses from inside the network and denies all others.
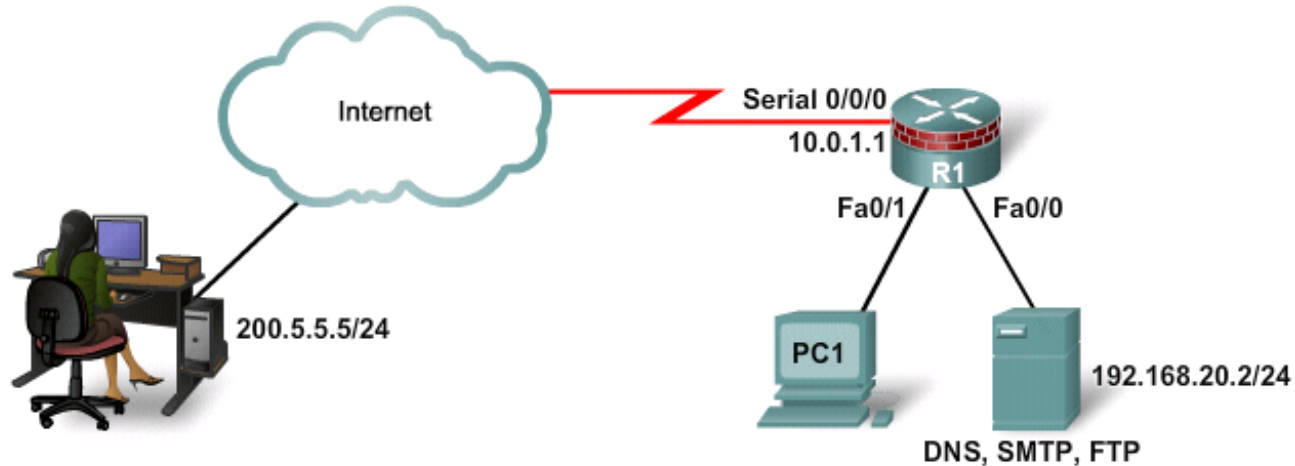

Inbound on Fa0/1

```
R1(config)# access-list 105 permit ip 192.168.1.0 0.0.0.255 any
```

Protect DNS, SMTP, and FTP

- DNS, SMTP, and FTP are common services that often must be allowed through a firewall.

Outbound on Fa0/0

```
R1(config)# access-list 180 permit udp any host 192.168.20.2 eq domain
R1(config)# access-list 180 permit tcp any host 192.168.20.2 eq smtp
R1(config)# access-list 180 permit tcp any host 192.168.20.2 eq ftp
R1(config)# access-list 180 permit tcp host 200.5.5.5 host 192.168.20.2 eq telnet
R1(config)# access-list 180 permit tcp host 200.5.5.5 host 192.168.20.2 eq 22
R1(config)# access-list 180 permit udp host 200.5.5.5 host 192.168.20.2 eq syslog
R1(config)# access-list 180 permit udp host 200.5.5.5 host 192.168.20.2 eq snmptrap
```

Filter ICMP Messages

- Hackers use ICMP packets for pings sweeps and DoS flood attacks.
  - Both ICMP echo and redirect messages should be blocked inbound by the router.
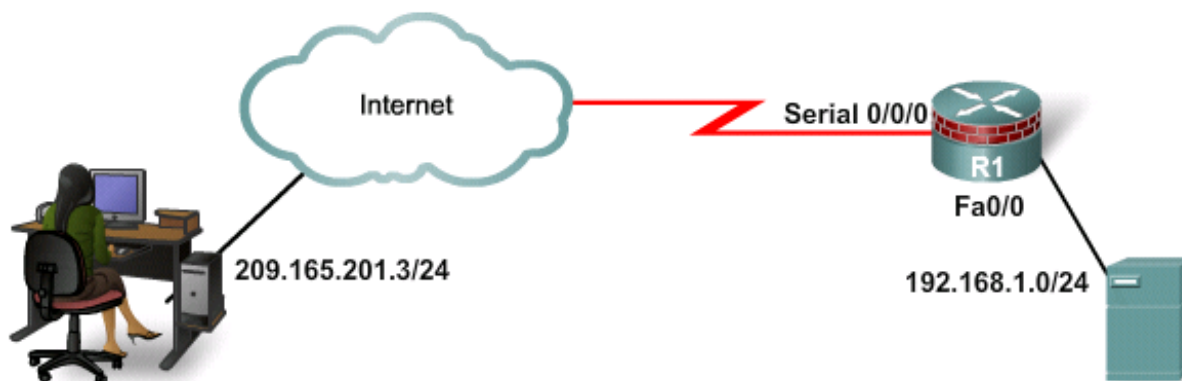
Filter ICMP Messages

- Several inbound ICMP messages are required for proper network operation:
  - **Echo reply** - Allows internal users to ping external hosts.
  - **Source quench** - Requests the sender to decrease the traffic rate.
  - **Unreachable** - Unreachable messages are generated for packets that are administratively denied by an ACL.
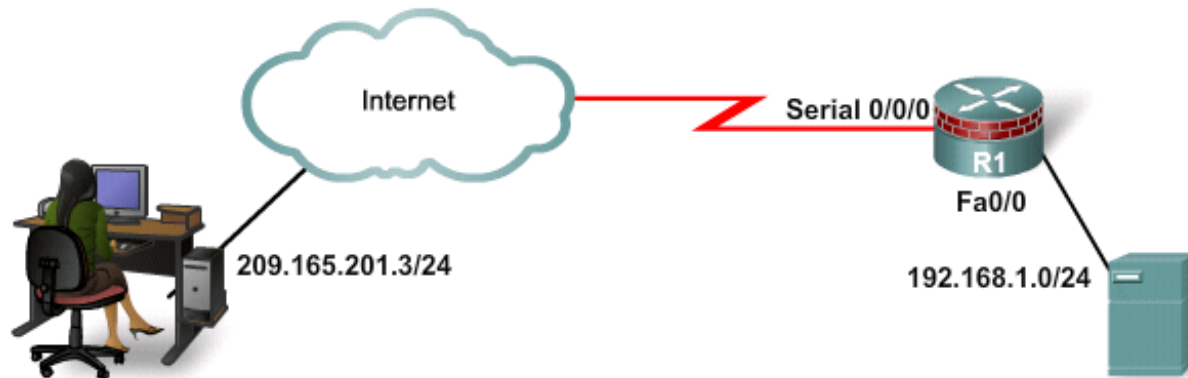
Inbound on S0/0/0

```
R1(config)# access-list 150 permit icmp any any echo-reply
R1(config)# access-list 150 permit icmp any any source-quench
R1(config)# access-list 150 permit icmp any any unreachable
R1(config)# access-list 150 deny icmp any any
R1(config)# access-list 150 permit ip any any
```

Filter ICMP Messages

- Several outbound ICMP messages are required for proper network operation:
  - **Echo** - Allows users to ping external hosts.
  - **Parameter problem** - Informs the host of packet header problems.
  - **Packet too big** - Required for packet MTU discovery.
  - **Source quench** - Throttles down traffic when necessary.
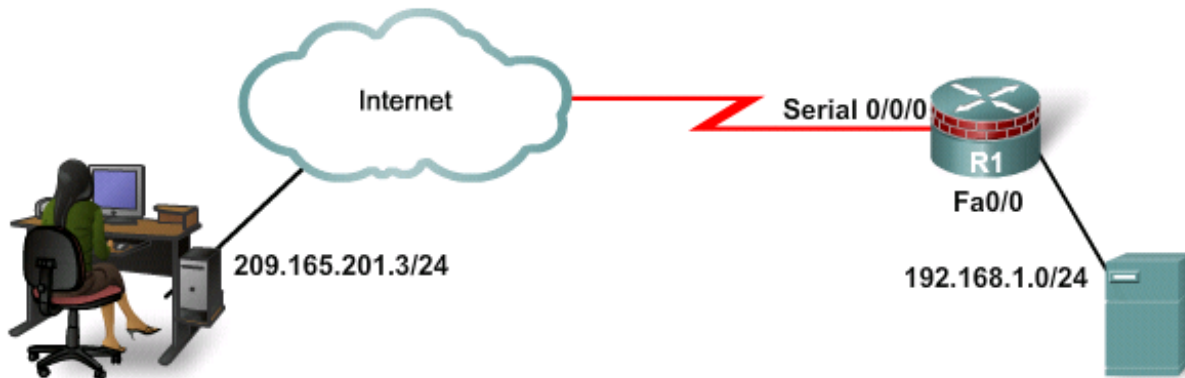
Inbound on Fa0/0

```
R1(config)# access-list 105 permit icmp 192.168.1.0 0.0.0.255 any echo
R1(config)# access-list 105 permit icmp 192.168.1.0 0.0.0.255 any parameter-problem
R1(config)# access-list 105 permit icmp 192.168.1.0 0.0.0.255 any packet-too-big
R1(config)# access-list 105 permit icmp 192.168.1.0 0.0.0.255 any source-quench
R1(config)# access-list 105 deny icmp any any
R1(config)# access-list 105 permit ip any any
```

Firewall

- A firewall prevents undesirable traffic from entering prescribed areas within a network.

- A firewall is a system or group of systems that enforces an access control policy between networks.
  - For example:
    - A packet filtering router
    - A switch with two VLANs
    - Multiple hosts with firewall software

- In 1989, AT&T Bell Laboratories developed the first stateful firewall.
  - A stateful firewall is able to determine if a packet belongs to an existing flow of data.

Firewall

- **Allow** web traffic from any external address to the web server

- **Allow** traffic to FTP server

- **Allow** traffic to SMTP server

- **Allow** traffic to internal IMAP server

- **Deny** all inbound traffic with network addresses matching internal-registered IP addresses

- **Deny** all inbound traffic to server from external addresses

- **Deny** all inbound ICMP echo request traffic

- **Deny** all inbound MS Active Directory

- **Deny** all inbound MS SQL server ports

- **Deny** all MS Domain Local Broadcasts
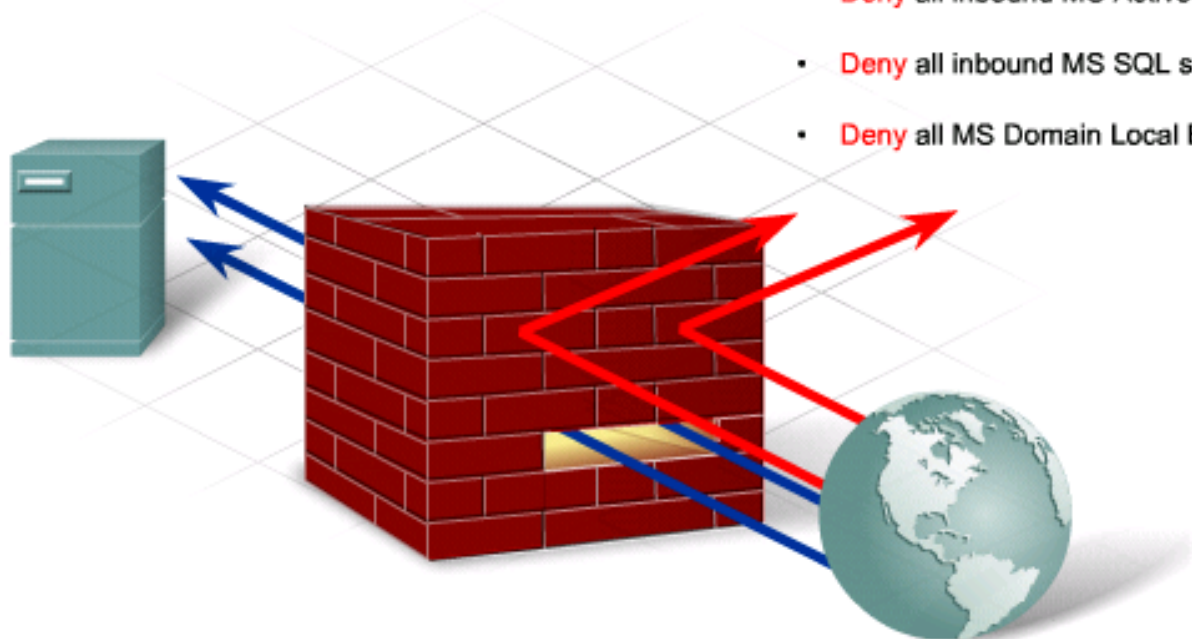
Stateless versus Stateful Packet Filtering

- Stateless packet filtering:
  - ACLs filter traffic based on source and destination IP addresses, TCP and UDP port numbers, TCP flags, and ICMP types and codes.

- Stateful packet filtering:
  - Inspection remembers certain details, or the state of that request.
  - Device maintains records of all connections passing through the firewall, and is able to determine whether a packet is the start of a new connection, or part of an existing connection.
  - A stateful firewall monitors the state of connections, whether the connection is in an initiation, data transfer, or termination state.

- Note:
  - A packet-filtering firewall typically can filter up to the transport layer, while a stateful firewall can filter up to the session layer.

Packet Filtering Firewalls

- Packet-filtering firewalls are usually part of a router firewall and primarily uses ACLs.

  – It examines a packet based on the information in a packet header.

- Packet-filtering firewalls use a simple policy table lookup that permits or denies traffic based on specific criteria:

  – Source IP address

  – Destination IP address

  – Protocol

  – Source port number

  – Destination port number

  – Synchronize/start (SYN) packet receipt

Common Types of Firewalls

- Packet-filtering firewall

Layer 7	Application	
Layer 6	Presentation	
Layer 5	Session	
Layer 4	Transport	Packet Filtering
Layer 3	Network	
Layer 2	Data Link	
Layer 1	Physical	

- Stateful firewall

Layer 7	Application	
Layer 6	Presentation	
Layer 5	Session	Stateful Firewall
Layer 4	Transport	
Layer 3	Network	
Layer 2	Data Link	
Layer 1	Physical	
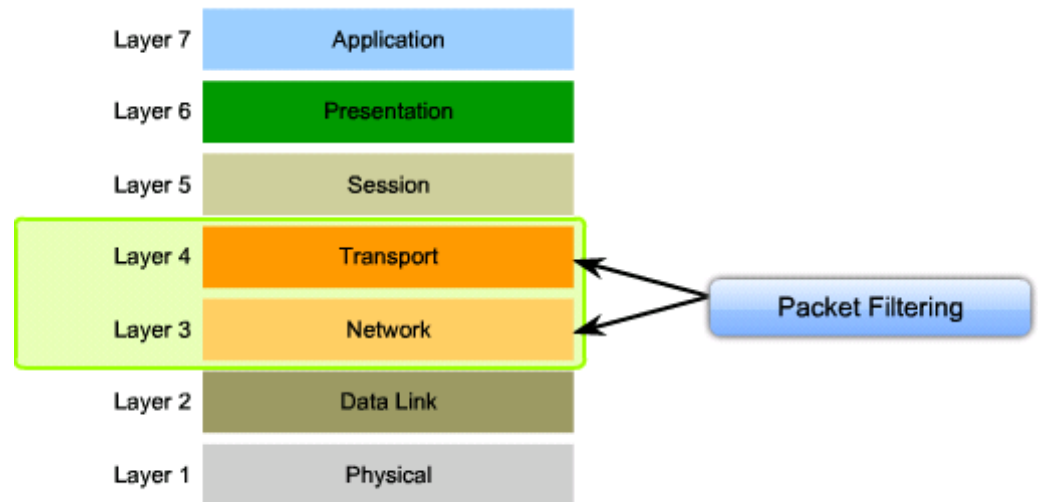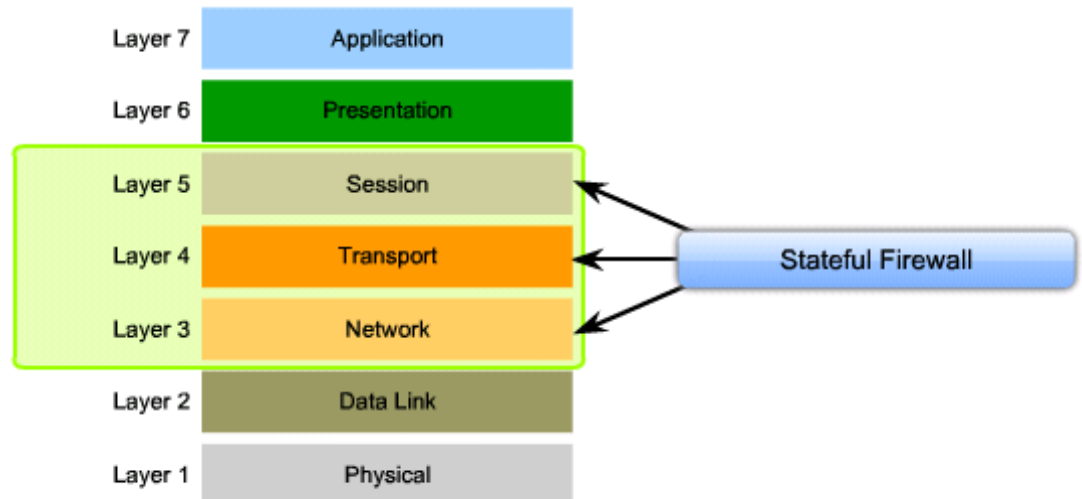
Stateful Firewalls

- Stateful firewalls are the most versatile and common firewall technology in use.

- Stateful filtering tracks each connection traversing all interfaces of the firewall and confirms that they are valid.
  - The firewall examines information in the headers of Layer 3 packets and Layer 4 segments.

Stateful Firewalls

- Also called "stateful packet filters" and "application-aware packet filters."

- Stateful firewalls have two main improvements over packet filters:
  - They maintain a session table (state table) where they track all connections.
  - They recognize dynamic applications and know which additional connections will be initiated between the endpoints.

Stateful Firewalls

- Stateful firewalls inspect every packet, compare the packet against the state table, and may examine the packet for any special protocol negotiations.

- Stateful firewalls operate mainly at the Transport (TCP and UDP) layer.

Cisco IOS Firewall TCP Handling

Source Address — 10.0.0.11
Destination Address — 172.30.0.50
Source Port — 1026
Destination Port — 23
Initial Sequence # — 49091
Ack —
Flag — Syn

#1
No Data
#3

10.0.0.11

10.0.0.11
172.30.0.50
1026
23
49092
92514
Ack

Stateful inspection algorithm:

- Source IP
- Source port
- Destination IP
- Destination port
- Sequence number

172.30.0.50
10.0.0.11
23
1026
92513
49092
Syn-Ack

#2

172.30.0.50

IP Header
TCP Header

324P_202

Cisco IOS Firewall UDP Handling

Source Address	10.0.0.11
Destination Address	172.30.0.50
Source Port	1028
Destination Port	45000

#1

#2

Stateful inspection algorithm:

- Source IP
- Source port
- Destination IP
- Destination port
- UDP Responses
- Within a Timeout

10.0.0.11

172.30.0.50

172.30.0.50
10.0.0.11
45000
1028

IP Header

TCP Header

324P_203
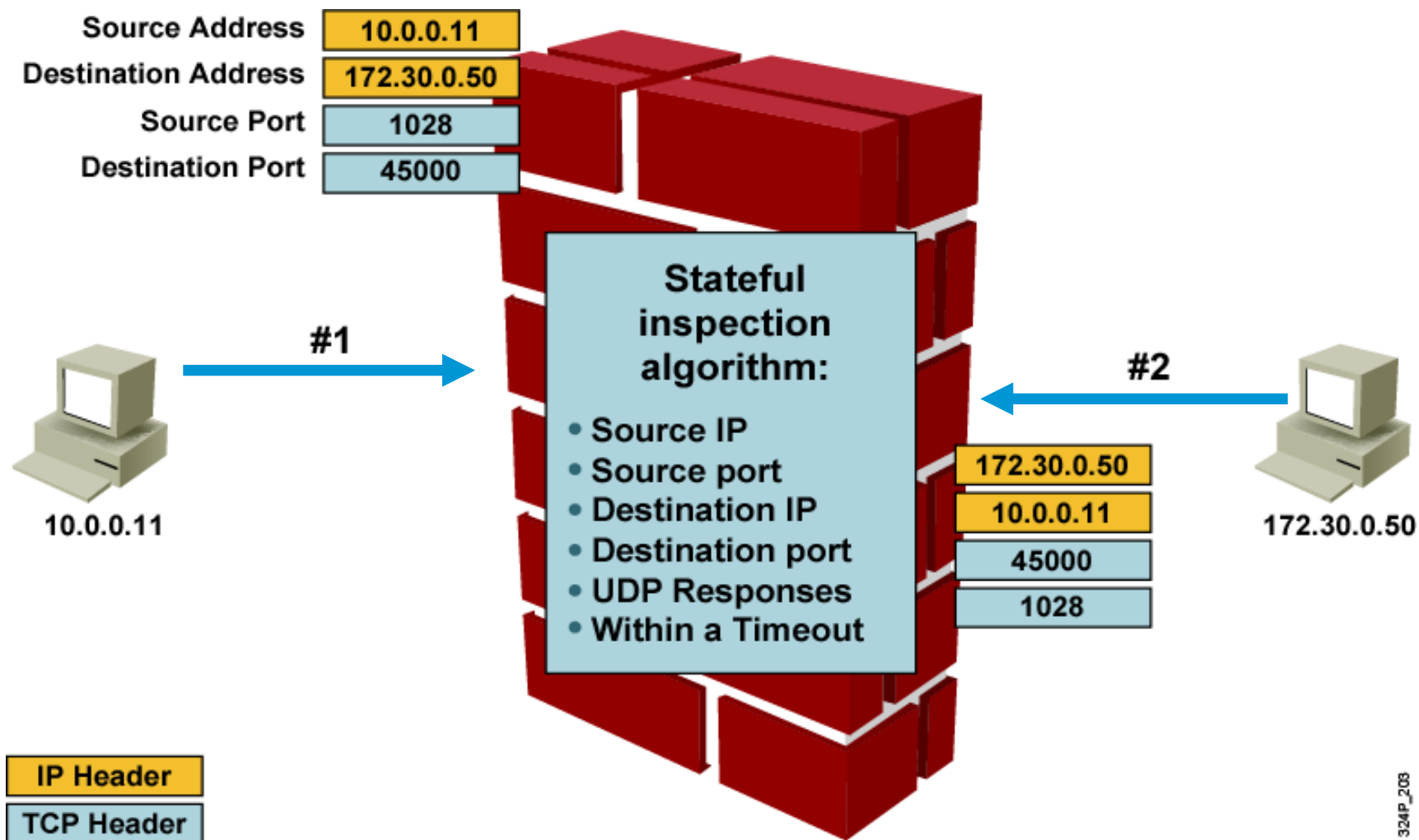
Other Types of Firewalls

- **Application gateway firewall (proxy firewall):**
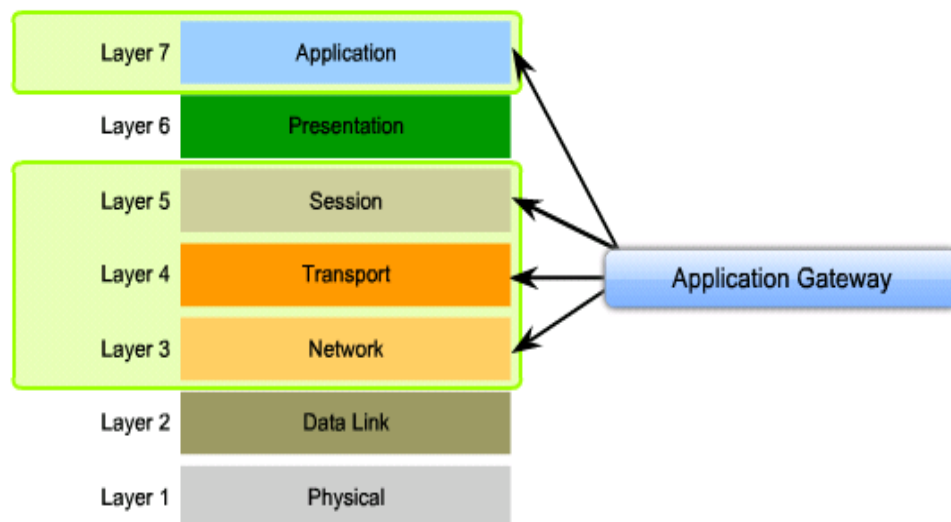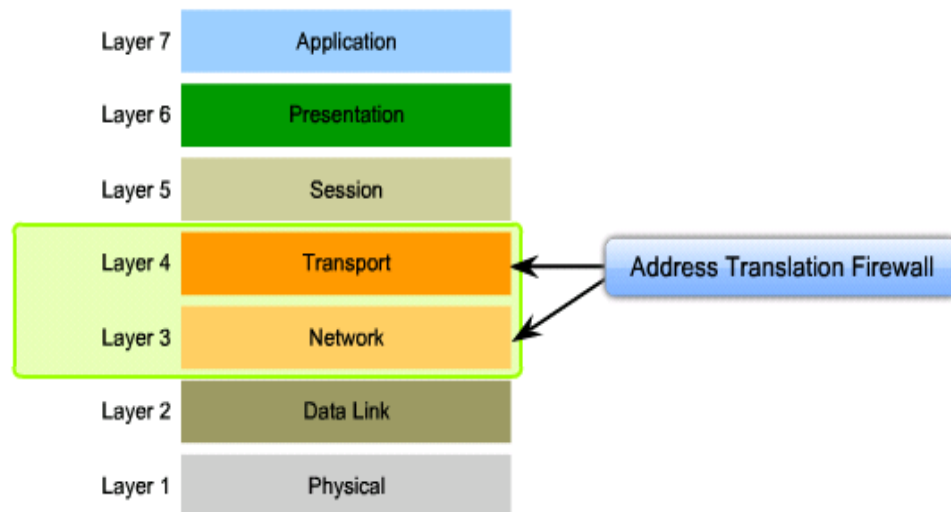 - Filters information at OSI Layers 3, 4, 5, and 7.
  - Firewall control and filtering is done in software.

Layer 7	Application
Layer 6	Presentation
Layer 5	Session
Layer 4	Transport
Layer 3	Network
Layer 2	Data Link
Layer 1	Physical

Application Gateway

- **Address-translation firewall:**
  - A firewall that expands the number of IP addresses available and hides network addressing design.

Layer 7	Application
Layer 6	Presentation
Layer 5	Session
Layer 4	Transport
Layer 3	Network
Layer 2	Data Link
Layer 1	Physical

Address Translation Firewall

Other Types of Firewalls

- **Host-based (server and personal) firewall:**
  - A PC or server with firewall software running on it.

- **Transparent firewall:**
  - A firewall that filters IP traffic between a pair of bridged interfaces.

- **Hybrid firewall:**
  - A firewall that is a combination of the various firewalls types.
  - For example, an application inspection firewall combines a stateful firewall with an application gateway firewall.

Modern Firewall Design

Traffic can originate from either the private to the DMZ or vice versa.

DMZ

Traffic originating from the public to the DMZ is allowed, but restricted.

Traffic originating from the DMZ to the public is not allowed.

Trusted (private)

Internet

Untrusted (public)

Traffic originating from the private to public is allowed.

Traffic originating from the public to the private is not allowed.

Zone-Based Policy Firewall

Benefits of ZPF

- Not dependent on ACLs.

- The router security posture is to block unless explicitly allowed.

- Policies are easy to read and troubleshoot with C3PL.

- One policy affects any given traffic, instead of needing multiple ACLs and inspection actions.

Zone-based policy firewall

- Zone-based policy firewall configuration model (ZPF or ZBF or ZFW) was introduced in 2006 with Cisco IOS Release 12.4(6)T.

- With ZPF the interfaces are assigned to zones and then an inspection policy is applied to traffic moving between the zones.

  - The default policy is to block all traffic unless explicitly allowed (CBACs default was allow all).

  - It supports previous firewall features, including SPI, application inspection, URL filtering, and DoS mitigation.

Basic ZPF Zone Topology
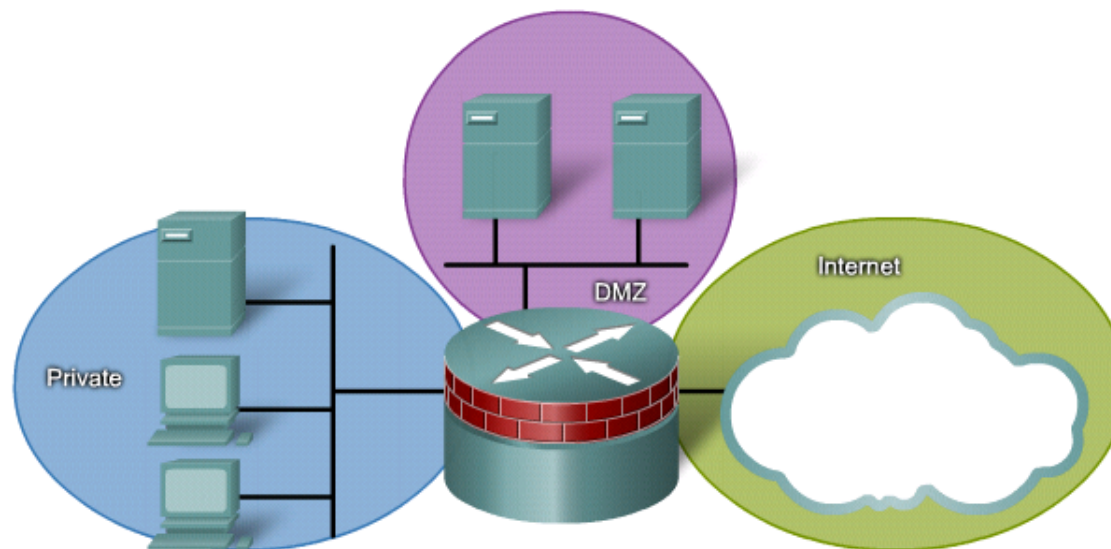
- If a new interface is added to the Private zone, the hosts on the new interface can pass traffic to all hosts in the Private zone.

- The new interface also inherits all existing Private zone policies when passing traffic to other zones.

CBAC or ZPF?

- Both CBAC and zones can be enabled concurrently on a router, just not on the same interface.

- For example, an interface cannot be configured as a security zone member and configured for IP inspection simultaneously.

3 Actions of ZPF

- Inspect
  - Configures Cisco IOS SPI (equivalent to ip inspect command).
  - It automatically allows for return traffic and potential ICMP messages.
  - For protocols requiring multiple parallel signaling and data sessions (for example, FTP or H.323), the inspect action also handles the proper establishment of data sessions.

- Pass
  - Analogous to a permit statement in an ACL.
  - It does not track the state of connections or sessions within the traffic.
  - Pass allows the traffic only in one direction.
  - A corresponding policy must be applied to allow return traffic to pass in the opposite direction.

- Drop
  - Analogous to a deny statement in an ACL.
  - A log option is available to log the rejected packets.

ZPF Rules

- A zone must be configured before it can be assigned to a zone.

- We can assign an interface to only one security zone.

- If traffic is to flow between all interfaces in a router, each interface must be a member of a zone.

- Traffic is implicitly allowed to flow by default among interfaces that are members of the same zone.

- To permit traffic to and from a zone member interface, a policy allowing or inspecting traffic must be configured between that zone and any other zone.

- Traffic cannot flow between a zone member interface and any interface that is not a zone member.

- We can apply pass, inspect, and drop actions only between two zones.

- Interfaces that have not been assigned to a zone function can still use a CBAC stateful packet inspection configuration.

- If we do not want an interface to be part of the zone-based firewall policy, it might still be necessary to put that interface in a zone and configure a pass-all policy (also known as a dummy policy) between that zone and any other zone to which traffic flow is desired.

The Self Zone

- The ZPF rules for a zone-based policy firewall are different when the router is the source or the destination of the traffic.

  - When an interface is configured to be a zone member, the hosts that are connected to the interface are included in the zone.

  - However, traffic to the router is not subject to the zone policies.

  - By default, all router IP interfaces are part of the self zone.

- A zone-pair that includes the self zone and associated policy, applies to router generated or traffic destined to the router.

  - It does not apply to traffic traversing the router.

- A policy can be defined using the self zone as either the source or the destination zone.

  - The self zone is a system-defined zone.

  - It does not require any interfaces to be configured as members.

Configuring ZPF

1. Create the Zones for the firewall.
   - `zone security`

2. Define Traffic Classes.
   - `class-map type inspect`

3. Specify Firewall Policies.
   - `policy-map type inspect`

4. Apply Firewall Policies to pairs of source and destination zones.
   - `zone-pair`

5. Assign Router Interfaces to zones.
   - `zone-member security`

Final ZPF Configuration

```
policy-map type inspect InsideToOutside
 class FOREXAMPLE
  inspect
!
```
3. Specify Firewall Policies

```
zone security Inside
 description Inside network
zone security Outside
 description Outside network
```
1. Create the Zones

```
zone-pair security InsideToOutside source Inside destination Outside
 service-policy type inspect InsideToOutside
!
```

```
interface FastEthernet0/0
 zone-member security Inside
!
```
4. Apply Firewall Policies

```
interface Serial0/0/0.100 point-to-point
 zone-member security Outside
!
```
5. Assign Interfaces to Zones

```
class-map type inspect FOREXAMPLE
 match access-group 101
 match protocol tcp
 match protocol udp
 match protocol icmp

 access-list 101 permit ip 10.0.0.0 0.0.0.255 any
```
2. Define the Traffic Classes