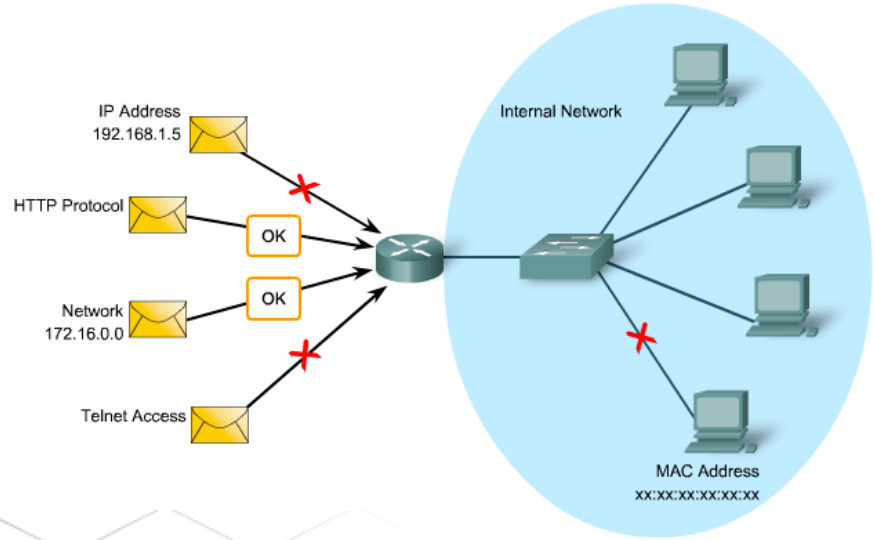
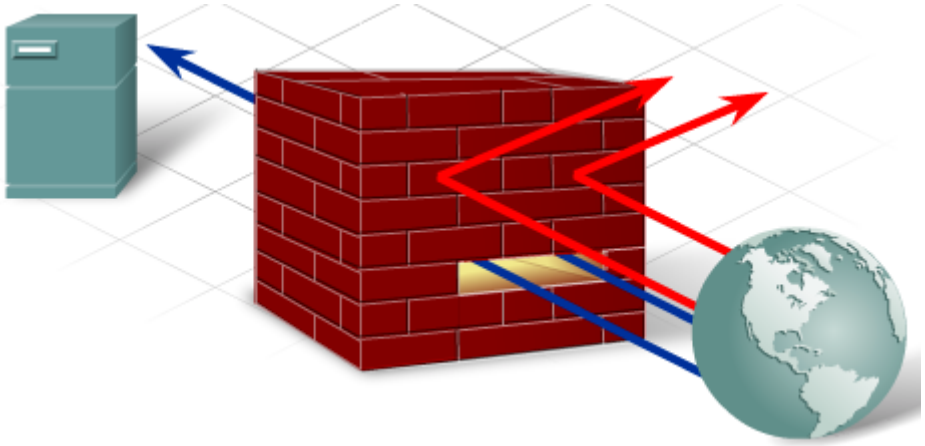
Firewall Technologies

Cisco.com

- Access Lists
- Firewalls





ACLs

- Standard
- Extended
- Numbered
- Named
- Stateful
  - Tables to track real-time state of end-end sessions
  - Session oriented nature of network traffic
  - TCP established
- Reflexive
  - Dynamically reflect certain types of inside-to-outside traffic
- Dynamic
  - Open a hole in the FW
  - Approved traffic
  - Finite period of time
- Time-based
  - Time of day
  - Day of week

ACLs

Protocol	Range
IP	1-99, 1300-1999
Extended IP	100-199, 2000-2699
Ethernet type code	200-299
DECnet and Extended DECnet	300-399
XNS	400-499
Extended XNS	500-599
AppleTalk	600-699
Ethernet address	700-799
IPX	800-899
Extended IPX	900-999
IPX SAP	1000-1099
Extended transparent bridging	1100-1199

Standard ACL

- Router(config)# access-list {1-99} {permit | deny} source-addr [source-wildcard]

Extended ACLs

- Router(config)# access-list {100-199} {permit | deny} protocol source-addr [source-wildcard] [operator operand] destination-addr [destination-wildcard] [operator operand] [established]
- All ACLs assume an implicit deny all,
- At least one permit statement

Apply an ACL

- This is the command to apply the ACL to an interface:
  - Router(config-if)# ip access-group access-list-number {in | out}
- This is the command to apply the ACL to a vty line:
  - Router(config-line)# access-class access-list-number {in | out}
- It is possible to create a named ACL instead of a numbered ACL. Named ACLs must be specified as either standard or extended.

Named ACL

- Router(config)# ip access-list [standard | extended] name_of_ACL

```
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 in
```

```
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-ext-nacl)# exit
R1(config)# interface Fa0/0
R1(config-if)# ip access-group ACL-1 in
R1(config-if)# exit
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)# exit
R1(config)# interface Fa0/1
R1(config-if)# ip access-group ACL-2 out
R1(config-if)# exit
```

Loggin – log parameter

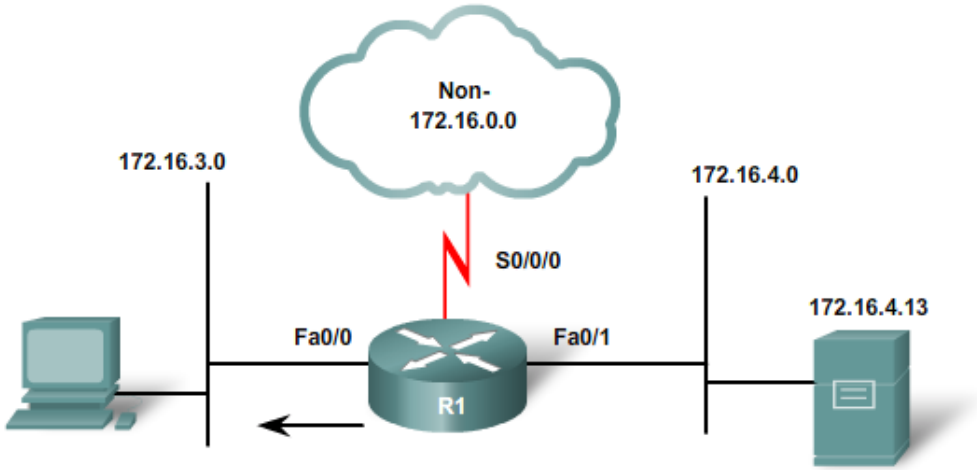
- R1(config)# access-list 101 permit tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 eq 22 log
- What is logged:
  - Action - permit or deny
  - Protocol - TCP, UDP, or ICMP
  - Source and destination addresses
  - For TCP and UDP - source and destination port numbers
  - For ICMP - message types

Features

- Implicit deny all
- Order of statements:
  - top-down, lowest number-up
- Directional filtering:
  - in-out
- Modifying ACLs
- One list per interface, per protocol, per direction

Using Standard ACLs

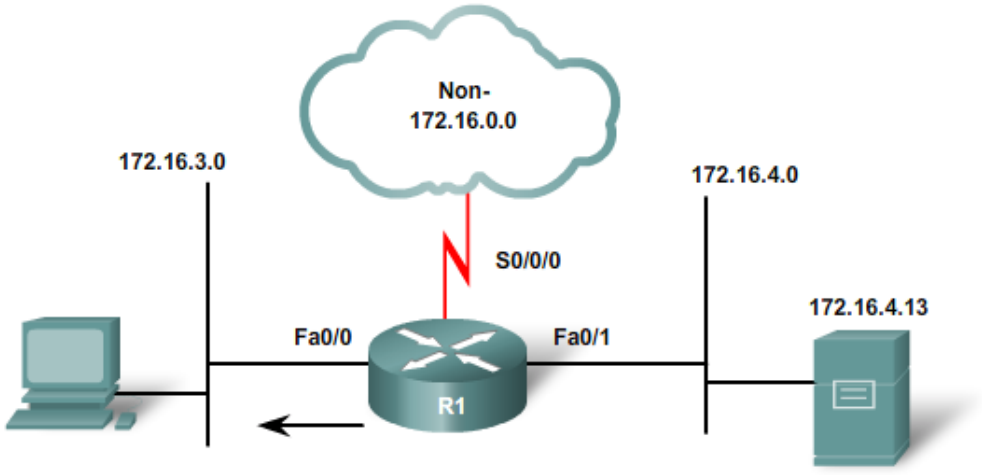
- R1(config)# access-list 1 deny 172.16.4.0 0.0.0.255
- R1(config)# access-list 1 permit any
- R1(config)# interface FastEthernet 0/0
- R1(config-if)# ip access-group 1 out



Using Extended ACLs

Cisco.com

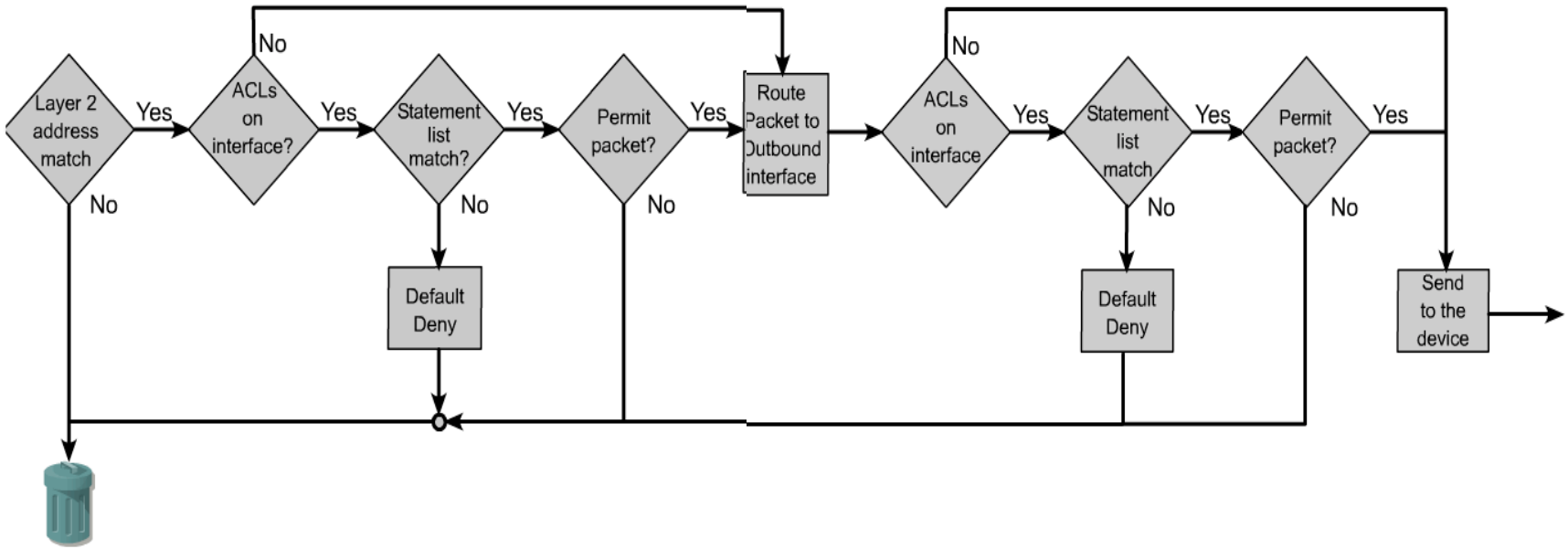
- R1(config)# access-list 101 deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 21
- R1(config)# access-list 101 deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 20
- R1(config)# access-list 101 permit ip any any
- Router(config)# interface fastethernet 0/1
- Router(config-if)# ip access-group 101 in



How they work

Cisco.com

ACL and routing processes in a router



Established ACLs

- established keyword
  - TCP builds a virtual circuit between two endpoints
  - support the two-way nature of TCP
  - blocked all traffic coming from the Internet except for the TCP reply traffic associated with established TCP traffic initiated from the inside of the network.

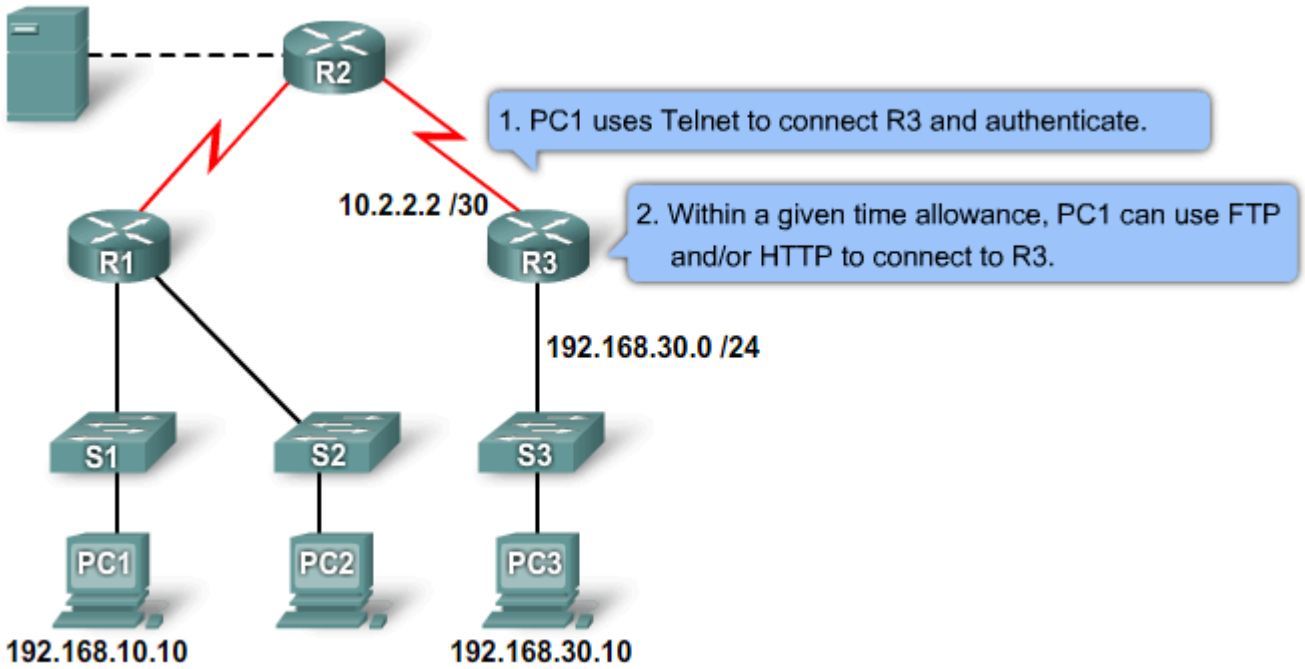
Reflexive ACLs

- Filter traffic based on
  - Source addresses
  - Destination addresses
  - Port numbers
  - Keep track of sessions.
- Temporary filters
  - Removed when a session is over

Reflexive ACLs

- R1(config)# ip access-list extended internal_ACL
- R1(config-ext-nacl)# permit tcp any any eq 80 reflect web-only-reflexive-ACL
- R1(config-ext-nacl)# permit udp any any eq 53 reflect dns-only-reflexive-ACL timeout 10
- R1(config)# ip access-list extended external_ACL
- R1(config-ext-nacl)# evaluate web-only-reflexive-ACL
- R1(config-ext-nacl)# evaluate dns-only-reflexive-ACL
- R1(config-ext-nacl)# deny ip any any
- R1(config)# interface s0/0/0
- R1(config-if)# description connection to the ISP.
- R1(config-if)# ip access-group internal_ACL out
- R1(config-if)# ip access-group external_ACL in

Dynamic ACLs



Dynamic ACLs

Step 1	`R3(config)# username Student password 0 cisco`
Step 2	`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`
Step 3	`R3(config)# interface serial 0/0/1` `R3(config-if)# ip access-group 101 in`

Step 4	`R3(config)# line vty 0 4`
	`R3(config-line)# login local`
	`R3(config-line)# autocommand access-enable host timeout 5`

Time Based ACLs

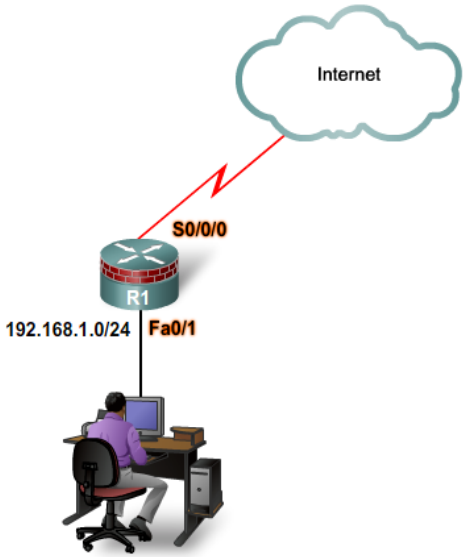
- R1(config)#time-range EVERYOTHERDAY
- R1(config-time.range)#periodic Monday Wednesday Friday 8:00 to 17:00
- R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq telnet time -range EVERYOTHERDAY
- R1(config)# interface s0/0/0
- R1(config-if)# ip access-group 101 out

Time Based ACLs

- 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

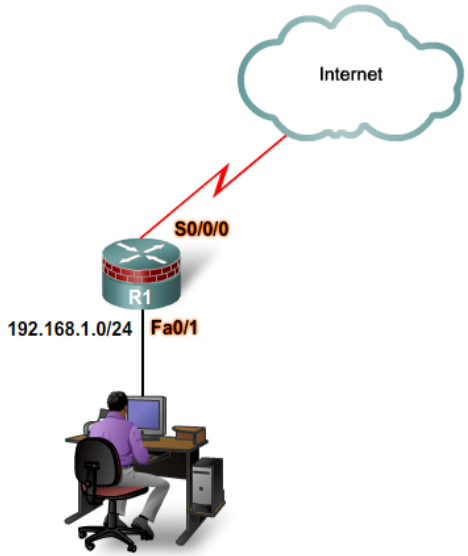
Mitigating attacks with ACLs

- 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 15.255.255.255 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
- R1(config)# interface s 0/0/0
- R1(config-if)# ip access-group 150 in

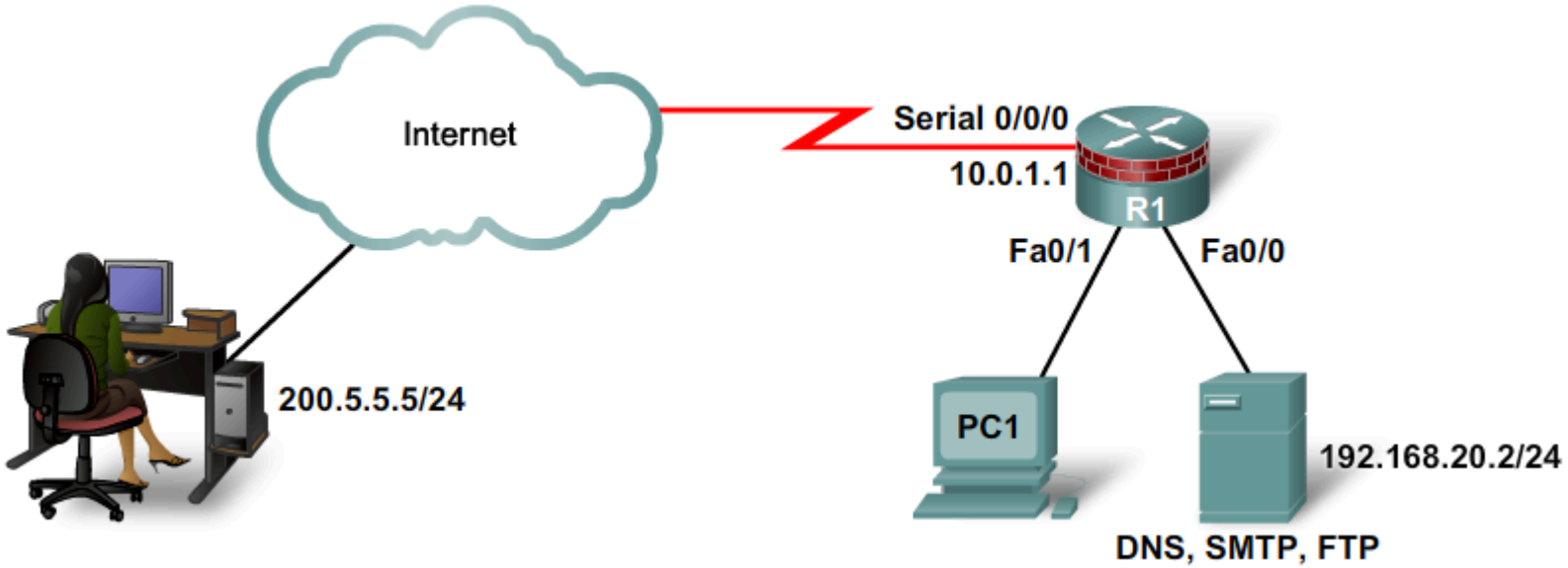


Mitigating attacks with ACLs

- R1(config)# access-list 105 permit ip 192.168.1.0 0.0.0.255 any
- R1(config)# interface fa 0/1
- R1(config-if)# ip access-group 105 in



Mitigating attacks with ACLs



Mitigating attacks with ACLs

Inbound on Serial 0/0/0

```
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 10.0.1.1 eq telnet
R1(config)# access-list 180 permit tcp host 200.5.5.5 host 10.0.1.1 eq 22
R1(config)# access-list 180 permit udp host 200.5.5.5 host 10.0.1.1 eq syslog
R1(config)# access-list 180 permit udp host 200.5.5.5 host 10.0.1.1 eq snmptrap
```

Inbound on S0/0/0

```
R1(config)# access-list 112 permit icmp any any echo-reply
R1(config)# access-list 112 permit icmp any any source-quench
R1(config)# access-list 112 permit icmp any any unreachable
R1(config)# access-list 112 deny icmp any any
R1(config)# access-list 112 permit ip any any
```

Inbound on Fa0/0

```
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any echo
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any parameter-problem
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any packet-too-big
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any source-quench
R1(config)# access-list 114 deny icmp any any
R1(config)# access-list 114 permit ip any any
```

Firewalls

- Packet filtering
- Stateful Firewall
- Firewalls in network design
- Context Based Access Control

Packet filtering

Cisco.com

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

Packet filters do not represent a complete firewall solution, but they are an important element.

Stateful Filtering

- Layer 3 packets and Layer 4 segments
  - TCP header
  - (SYN)
  - (RST)
  - (ACK)
  - (FIN)
  - other control codes
- Connection object
- Compare all inbound and outbound packets against session flows in the stateful session flow table

Stateful Filtering

Inside ACL (Outgoing Traffic)	Outside ACL (Incoming Traffic)

`permit ip 10.0.0.0 0.0.0.255 any`	`Dynamic: permit tcp host 209.165.201.3 eq 80 host 10.1.1.1 eq 1500`
	`permit tcp any host 10.1.1.2 eq 25`
	`permit udp any host 10.1.1.2 eq 53`
	`deny ip any any`

Context Based Access Control

- CBAC provides four main functions:
  - traffic filtering
  - traffic inspection
  - intrusion detection, and
  - generation of audits and alerts.

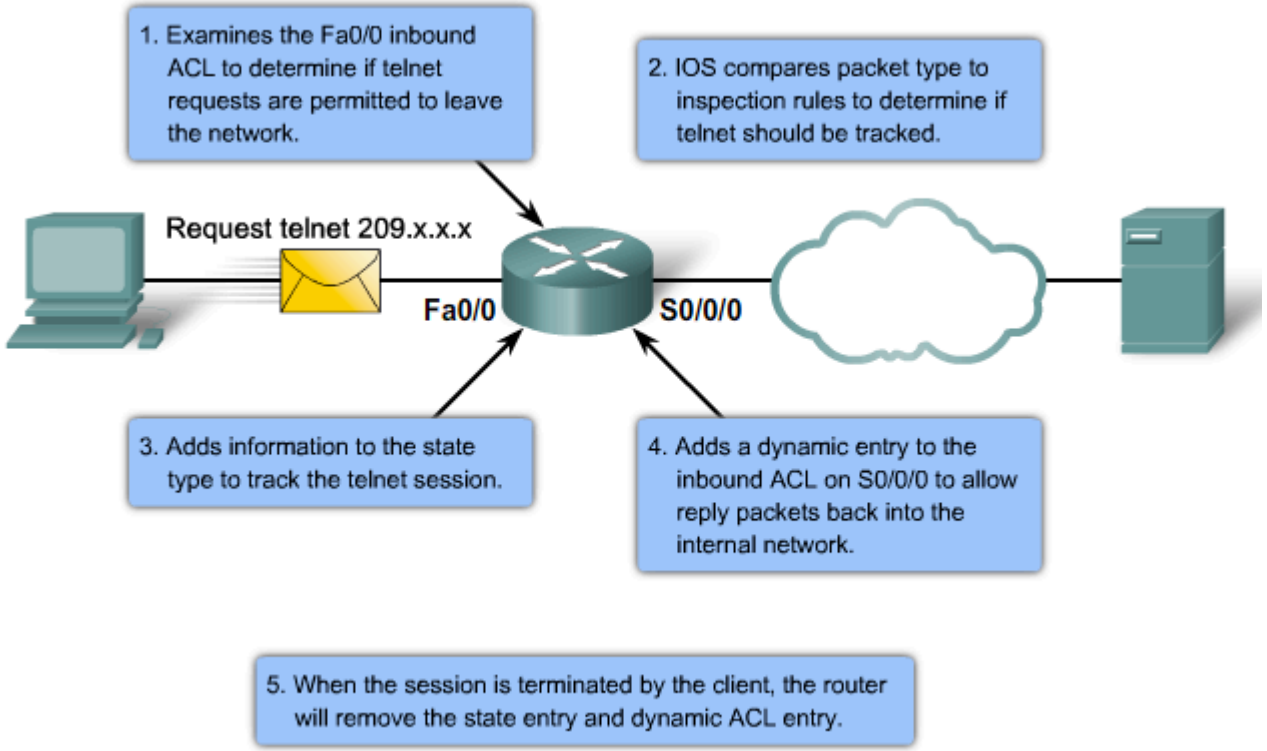
Context Based Access Control

- Characteristics
  - Monitors TCP connection setup
  - Tracks TCP sequence numbers
  - Inspects DNS queries and replies
  - Inspects common ICMP message types
  - Supports applications that rely on multiple connections
  - Inspects embedded addresses
  - Inspects Application Layer information

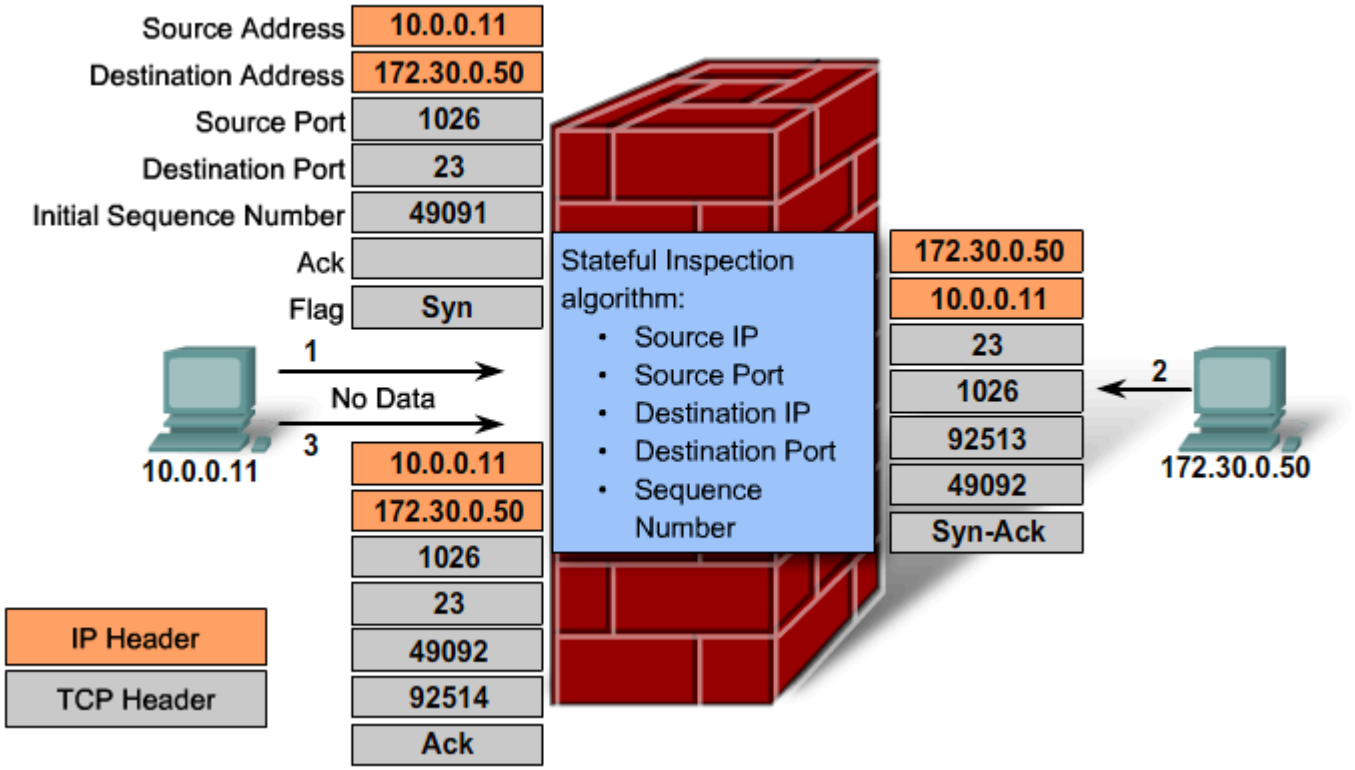
Context Based Access Control

- How does CBAC work?
  - CBAC creates openings in ACLs at firewall interfaces by adding a temporary ACL entry for a specific session. These openings are created when specified traffic exits the internal protected network through the firewall. The temporary openings allow returning traffic that would normally be blocked and additional data channels to enter the internal network back through the firewall. The traffic is allowed back through the firewall only if it is part of the same session and has the expected properties as the original traffic that triggered CBAC when exiting through the firewall. Without this temporary ACL entry, this traffic would be denied by the preexisting ACL. The state table dynamically changes and adapts with the traffic flow.

Context Based Access Control



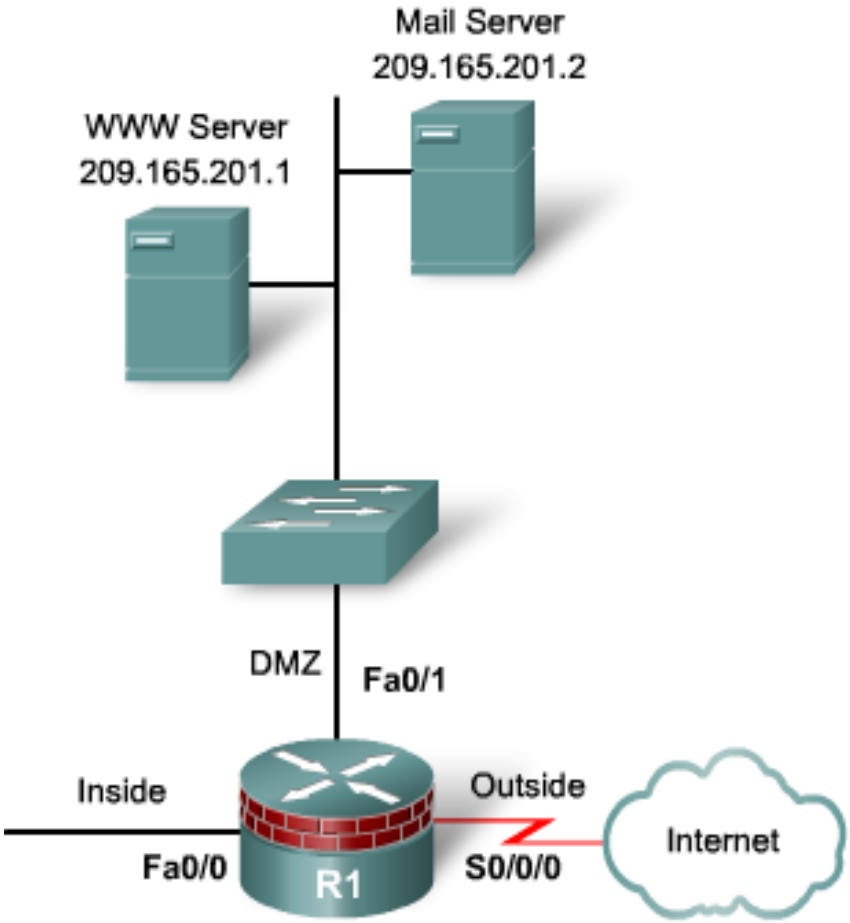
Context Based Access Control



Context Based Access Control

- Configuring:
  - There are four steps to configure CBAC:
  - Step 1. Pick an interface - internal or external.
  - Step 2. Configure IP ACLs at the interface.
  - Step 3. Define inspection rules.
  - Step 4. Apply an inspection rule to an interface.

Context Based Access Control



Context Based Access Control

- An administrator needs to permit inside users to initiate TCP, UDP, and ICMP traffic with all external sources. Outside clients are allowed to communicate with the SMTP server (209.165.201.1) and HTTP server (209.165.201.2) that are located in the enterprise DMZ. It is also necessary to permit certain ICMP messages to all interfaces. All other traffic from the external network is denied.

Context Based Access Control

- For this example, first create an ACL that allows TCP, UDP, and ICMP sessions and denies all other traffic.
  - R1(config)# access-list 101 permit tcp 10.10.10.0 0.0.0.255 any
  - R1(config)# access-list 101 permit udp 10.10.10.0 0.0.0.255 any
  - R1(config)# access-list 101 permit icmp 10.10.10.0 0.0.0.255 any
  - R1(config)# access-list 101 deny ip any any

Context Based Access Control

- This ACL is applied to the internal interface in the inbound direction. The ACL processes traffic initiating from the internal network prior to leaving the network.
  - R1(config)# interface Fa0/0
  - R1(config-if)# ip access-group 101 in

Context Based Access Control

- Next, create an extended ACL in which SMTP and HTTP traffic is permitted from the external network to the DMZ network only, and all other traffic is denied.
  - R1(config)# access-list 102 permit tcp any 209.165.201.1 0.0.0.0 eq 80
  - R1(config)# access-list 102 permit tcp any 209.165.201.2 0.0.0.0 eq smtp
  - R1(config)# access-list 102 permit icmp any any echo-reply
  - R1(config)# access-list 102 permit icmp any any unreachable
  - R1(config)# access-list 102 permit icmp any any administratively-prohibited
  - R1(config)# access-list 102 permit icmp any any packet-too-big
  - R1(config)# access-list 102 permit icmp any any echo
  - R1(config)# access-list 102 permit icmp any any time-exceeded
  - R1(config)# access-list 102 deny ip any any

Context Based Access Control

- This ACL is applied to the interface connecting to the external network in the inbound direction.
  - R1(config)# interface S0/0/0
  - R1(config-if)# ip access-group 102 in

Context Based Access Control

- If the configuration stopped here, all returning traffic, with the exception of ICMP messages, is denied because of the external ACL. Next, create inspection rules for TCP inspection and UDP inspection.
  - R1(config)# ip inspect name MYSITE tcp
  - R1(config)# ip inspect name MYSITE udp
- These inspection rules are applied to the internal interface in the inbound direction.
  - R1(config)# interface Fa0/0
  - R1(config-if)# ip inspect MYSITE in

Context Based Access Control

- The inspection list automatically creates temporary ACL statements in the inbound ACL applied to the external interface for TCP and UDP connections. This permits TCP and UDP traffic that is in response to requests generated from the internal network.

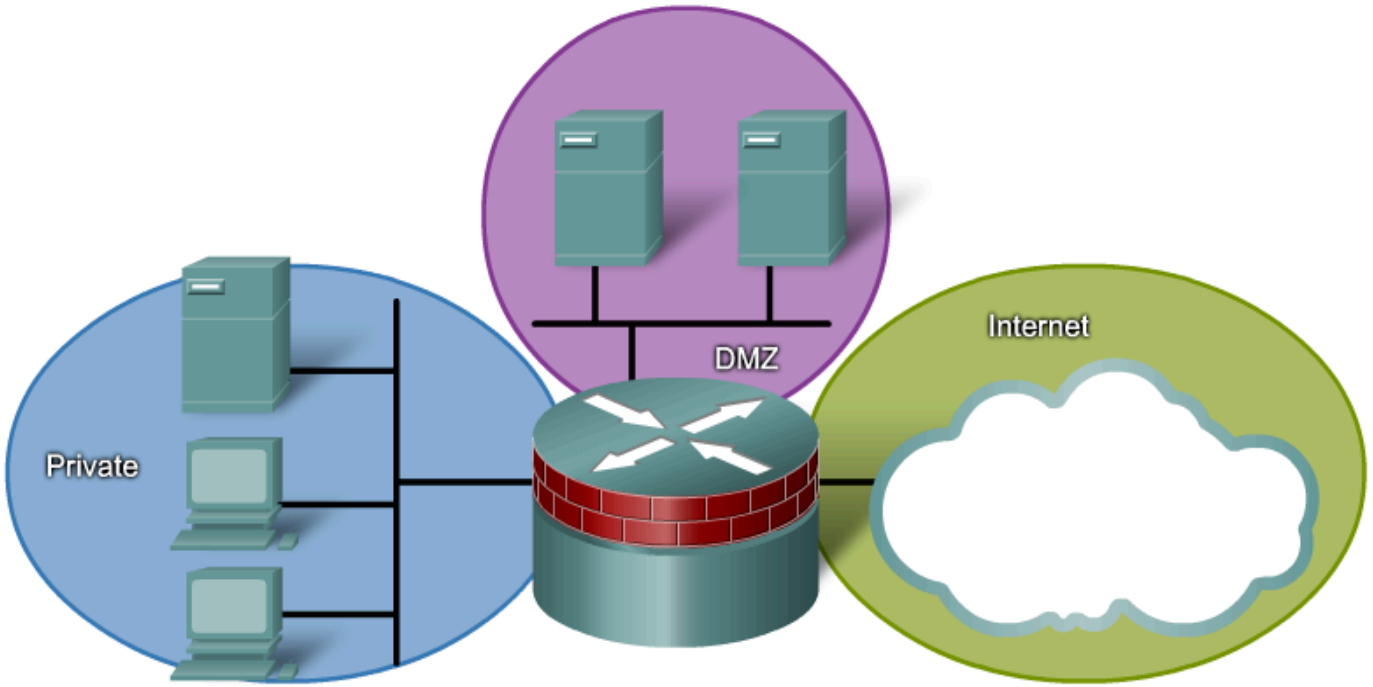
Audit


```
R1(config)# logging on
R1(config)# logging host 10.0.0.3
R1(config)# ip inspect audit-trail
R1(config)# no ip inspect alert-off
```

Zone Based Policy Firewall

In this example, if an additional interface is added to the private zone, the hosts connected to the new interface in the private zone can pass traffic to all hosts on the existing interface in the same zone. Additionally, hosts connected to the new interface in the private zone must adhere to all existing “private” policies related to that zone when passing traffic to other zones.

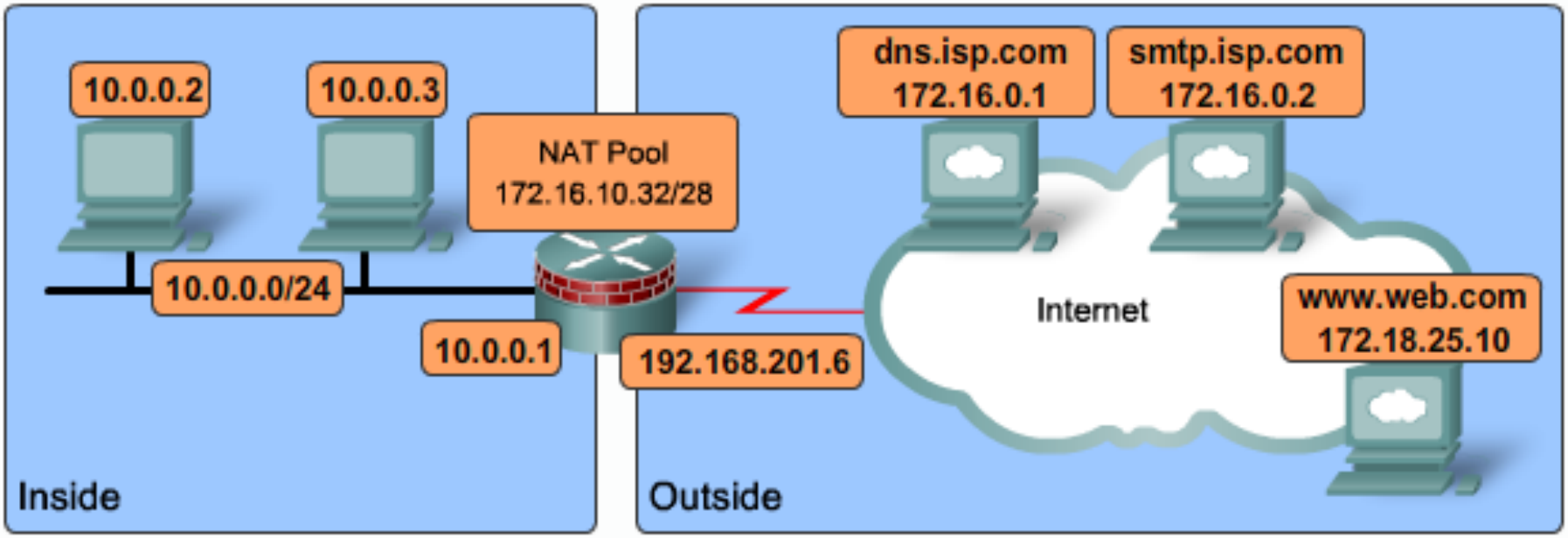
Basic Security Zone Topology



ZPF

- The default policy between zones is deny all.
- No policy – all traffic btwn zones is blocked
- CBAC implicitly allowed until blocked with ACL

Configure ZPF



Configure ZPF

- Step 1. Create the zones for the firewall with the zone security command.

```
FW(config)# zone security Inside
FW(config-sec-zone)# description Inside network
FW(config)# zone security Outside
FW(config-sec-zone)# description Outside network
```

Configure ZPF

- Step 2. Define traffic classes with the class-map type inspect command.

```
FW(config)# class-map type inspect FOREXAMPLE
FW(config-cmap)# match access-group 101
FW(config-cmap)# exit
FW(config)# access-list 101 permit ip 10.0.0.0 0.0.0.255 any
```

Configure ZPF

- Step 3. Specify firewall policies with the policy-map type inspect command.

```
FW(config)# policy-map type inspect InsideToOutside
FW(config-pmap)# class type inspect FOREXAMPLE
FW(config-pmap-c)# inspect
```

Configure ZPF

- Step 4. Apply firewall policies to pairs of source and destination zones using the zone-pair security command.
- Step 5. Assign router interfaces to zones using the zone-member security interface command.

```
FW(config)# zone-pair security InsideToOutside source Inside destination Outside
FW(config-sec-zone-pair)# description Internet Access
FW(config-sec-zone-pair)# service-policy type inspect InsideToOutside
FW(config-sec-zone-pair)# interface F0/0
FW(config-if)# zone-member security Inside
FW(config-if)# interface S0/0/0.100 point-to-point
FW(config-if)# zone-member security Outside
```