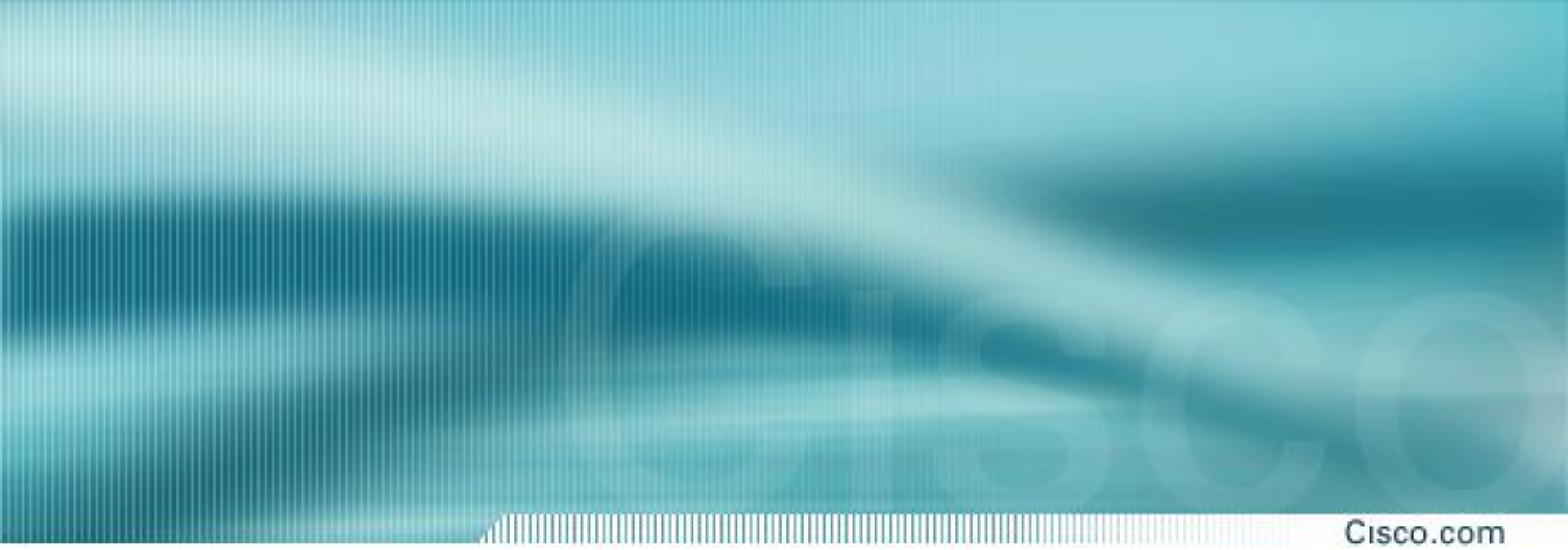
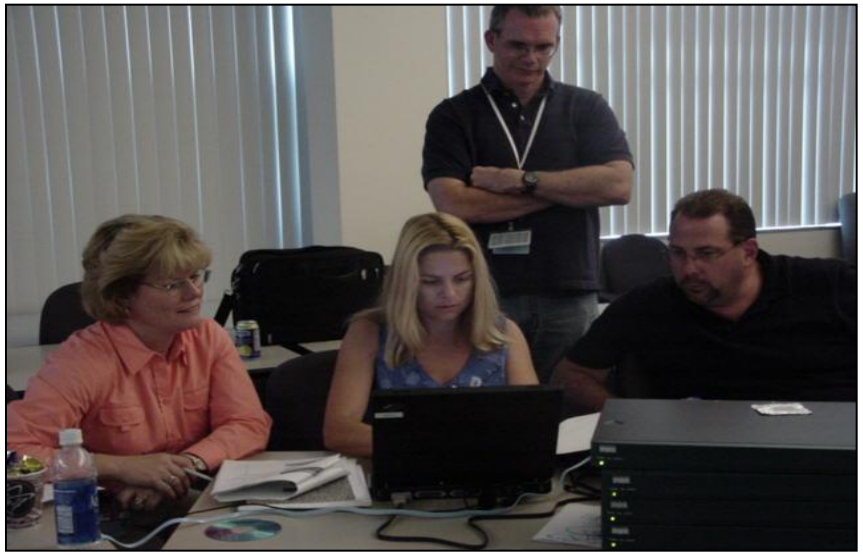


IP Access Control Lists (ACL)

Module Objectives

Upon completion, you will be able to:

- **Identify** the types of IP Access Control Lists.
- **Describe** typical uses for IP Access Lists.
- **Understand** Access List-related terms and concepts.
- Given a specific criteria, **select** the type and placement of Access Lists for best results.



What Are IP Access Control Lists?

- A Cisco IOS feature.
- Sequential list of “permit” or “deny” statements, which block or permit routed traffic.
- Used with:

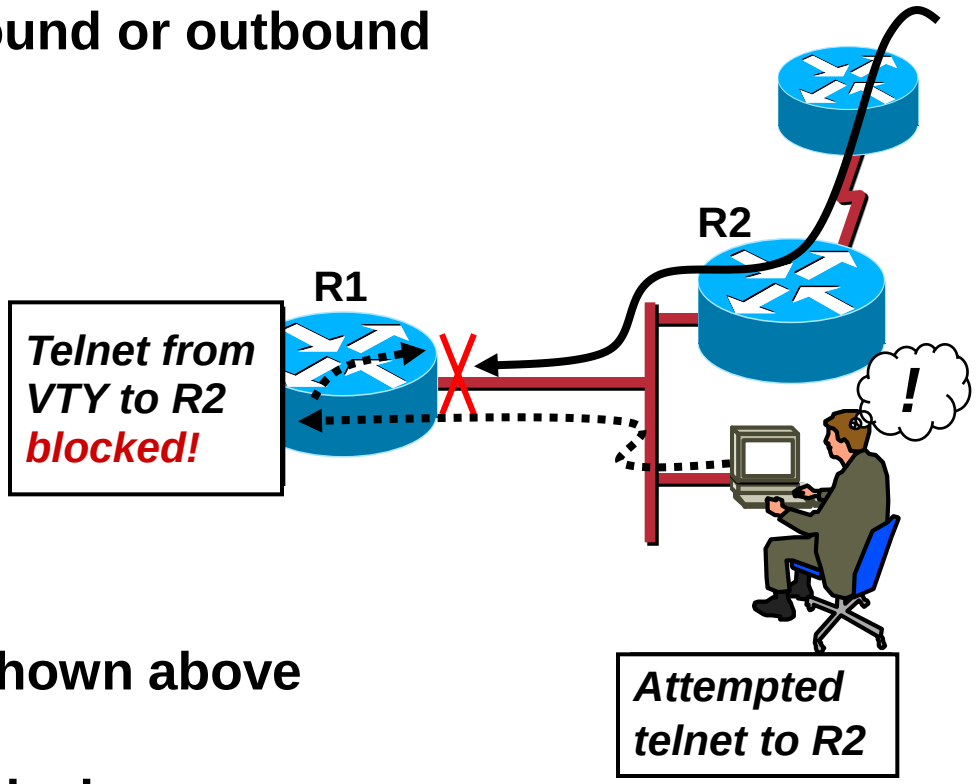
  Interfaces

  VTYs

  Routing Protocols

What Problems do Access Control Lists Solve?

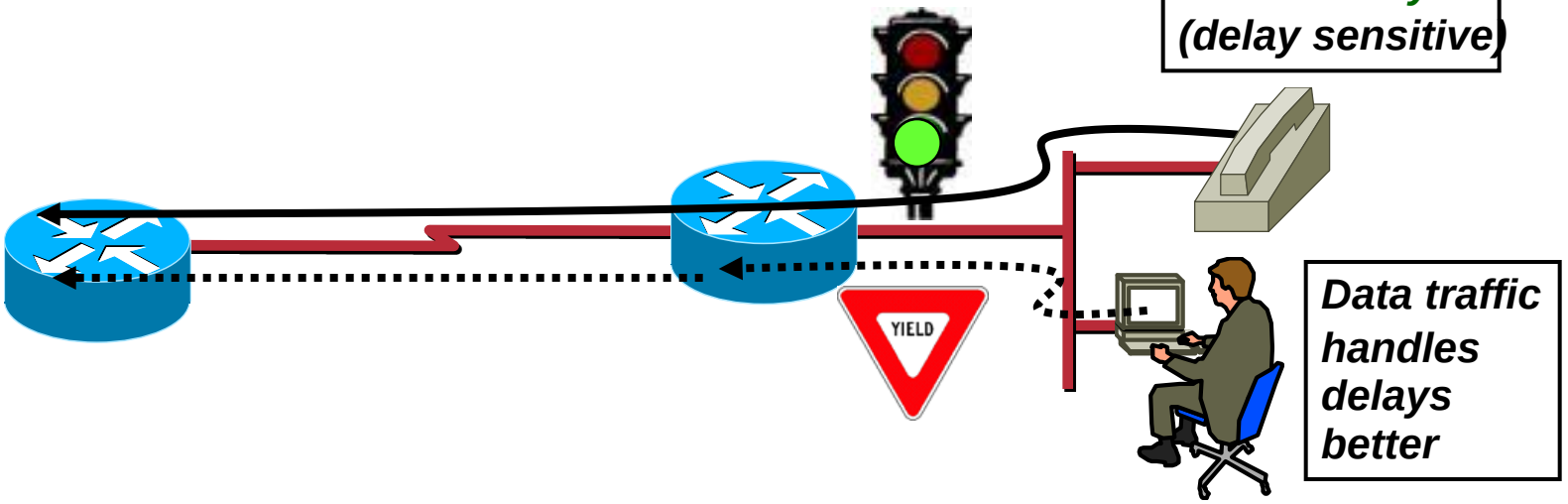
- **Block Unwanted Traffic – inbound or outbound**
  - Basic network security
  - Bandwidth control
  - Enforce network policy
  - Control routes sent, received, and/or redistributed



- **Permit the Good Stuff**
  - The good side of the list shown above
- **Control Access to IOS-based devices**
  - Block &/or permit VTY (telnet) from certain nodes/networks
  - Block &/or permit telnet to *other* devices (out from a VTY)

What Problems do Access Control Lists Solve?

- **Identify or classify traffic for advanced features**
  - –Congestion Avoidance
    - Setting IP Precedence (for Voice or Video)
  - – Congestion Management
    - Queuing Types
  - – Network Address Translation (NAT)
  - – Dialing (and disconnect) Criteria



Types of IP ACLs

Most Common (90%):

- Standard ACLs
- Extended ACLs

Less Common:

- Lock and Key (dynamic ACLs)
- Reflexive ACLs
- Time-based ACLs using time ranges
- Commented IP ACL entries
- Context-based ACL
- Authentication proxy
- Named ACLs
- Turbo ACLs
- Distributed time-based ACLs

Standard IP ACL Syntax

access-list access-list-number {permit|deny} {host | source source-wildcard | any}

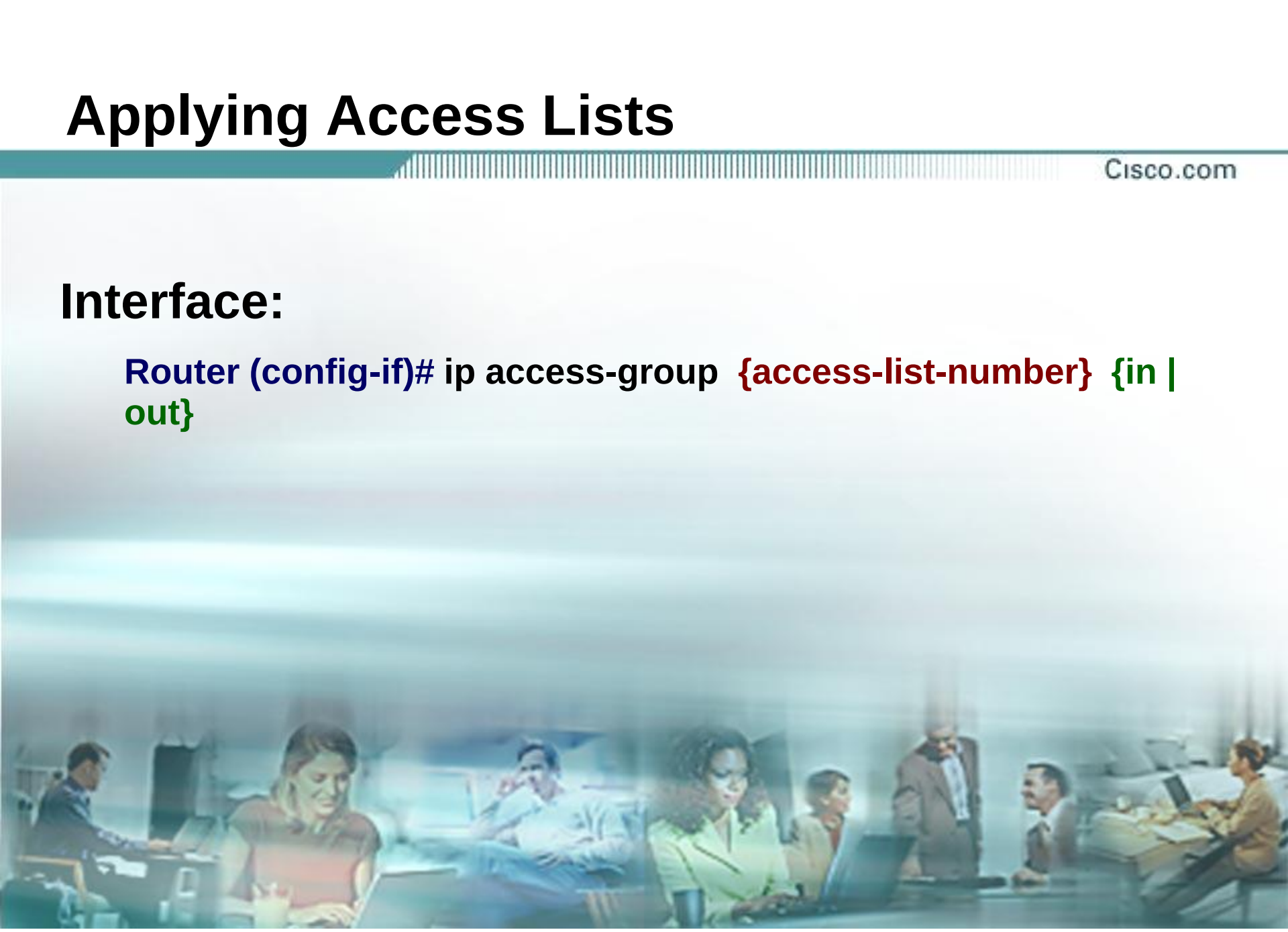
- **Numbered 1 – 99**
- **Only look at the IP *Source Address***
- **Easiest to Configure**
- **Good for blocking traffic close to destination**

Note: You cannot delete lines of a numbered access list. You must first *remove* the entire access list.

Applying Access Lists

Interface:

Router (config-if)# ip access-group {access-list-number} {in | out}



Access List Overview

Address	*Wildcard Mask*	*Match Condition*
0.0.0.0	255.255.255.255	All addresses will match ACL condition
131.54.0.0/16	0.0.255.255	Network 131.54.0.0 is permitted/denied
131.22.5.2/16	0.0.0.0	Only host 131.22.5.2 is permitted/denied
131.111.0.8	0.0.0.7	Only subnet 131.111.0.8/29 is permitted/denied
131.111.8.8	0.0.0.7	Only subnet 131.111.8.8/29 is permitted/denied
131.111.8.16	0.0.0.3	Only subnet 131.111.8.16/30 is permitted/denied

- **Access Lists could be:**

  Inbound: Check the filter condition before Routing table lookup

  Outbound: Checks the filter condition after Routing table lookup

- **To Configure Access List, we use Wildcard Masks:**

  Wildcard is the reverse of subnet mask

  Wildcard masks can be discontiguous (subnet masks can't)

  0 bit => must match bits in address

  1 bit => don't care. No need to match.

Wildcard Masks vs. Subnet Masks

Type:	Contiguous or not?:	Zero (0) means…	One (1) means …	Examples:
Wildcard	Not required.	Match, must match address bits.	Ignore	`access-list 9 permit 10.1.2.0 0.0.0.255`
Subnet	Yes, must be.	Ignore	Match	`IP address 10.1.2.0 255.255.255.0`

- This statement…
- `access-list 9 permit 10.1.2.0 0.0.0.255`
- … uses a wildcard mask, which permits any host in the range of 10.1.2.0 network.

Extended IP ACL Syntax

access-list access-list-number {permit|deny} protocol {host | source source-wildcard | any} {host | destination destination-wildcard | any} [precedence *precedence name or #*]

- **Numbered 100 – 199**
- **Looks both the IP *source address* and *destination address***
- **Checks many IP and upper layer header fields**
- **Good for blocking traffic anywhere**

Applying Access Lists

Interfaces:

Router (config-if)# ip access-group {access-list-number} {in | out}

EXAMPLE

```
router(config)#access-list 10 deny 172.16.40.0 0.0.0.255
router(config)#access-list 10 permit any

router(config)#interface fa0/1
Router(config-if)#ip access-group 10 out
```

ACL Guidelines

- Use Standard Access lists when filtering near destination:
  - Use Extended Access lists when filtering near Source, and/or need to specify protocol, ports, etc.
  - Create ACL first, then Apply to interface
  - Invest time to plan your ACL…consider CPU of Routers
  - Carefully place…consider bandwidth, etc.
  - Remember the implicit "deny all" at end of ACL
  - No editing or re-ordering of numbered ACLs (other than *adding* lines at end)

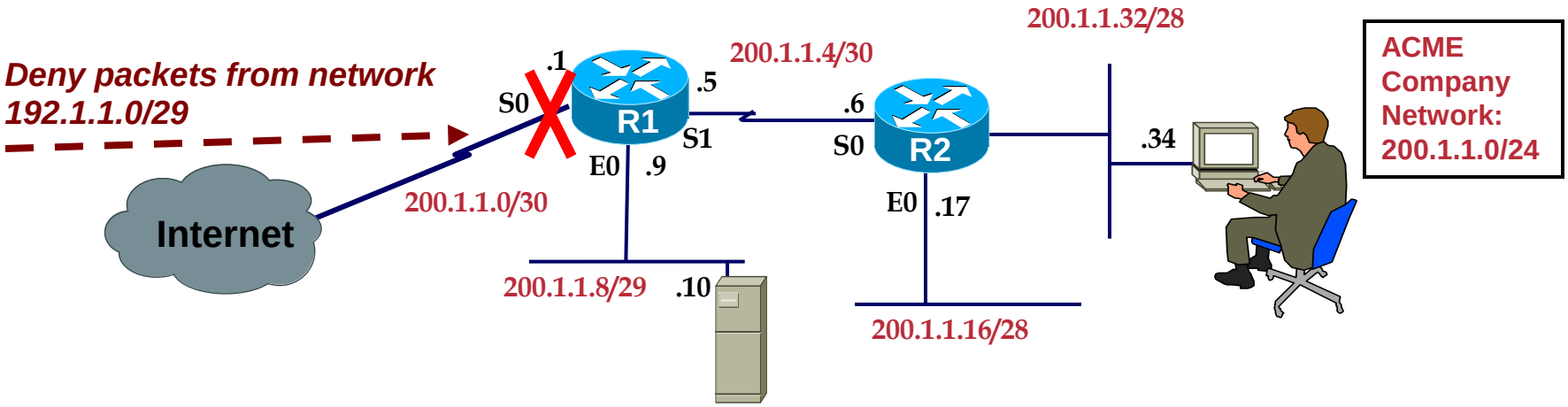
Which IP Protocols Are Supported?


```
Router(config)#access-list 111 permit ?
  <0-255>  An IP protocol number
  ahp      Authentication Header Protocol
  eigrp    Cisco's EIGRP routing protocol
  esp      Encapsulation Security Payload
  gre      Cisco's GRE tunneling
→ icmp     Internet Control Message Protocol
  igmp     Internet Gateway Message Protocol
  igrp     Cisco's IGRP routing protocol
→ ip       Any Internet Protocol
  ipinip   IP in IP tunneling
  nos      KA9Q NOS compatible IP over IP tunneling
→ ospf     OSPF routing protocol
  pcp      Payload Compression Protocol
→ tcp      Transmission Control Protocol
→ udp      User Datagram Protocol
```

Arrows indicate those protocols ACLs are used for most often

Implementing Security with ACLs

Access Lists Overview



- Used to block or permit only certain traffic
- Standard Access List for IP (1-99) & (1300-1999)

  Blocks only Source Addresses
- Extended Access for IP (100-199) & (2000-2699)

  Source and Destination Address, ICMP, TCP, UDP, Ports, etc.
- IPX Standard Access List (800-899)
- IPX Extended Access List (900-999)

Know these for the exam…

Two Basic Steps

- **Define** the Access Control List, then…

```
Router(config)# access-list 8 permit 131.108.7.0 0.0.0.3
Router(config)# access-list 8 permit 131.108.2.0 0.0.0.255
                (access-list 8 deny any)
```

- **Apply it** to an interface

```
Router(config)# interface s0
Router(config-if)# ip access-group 8 in
```

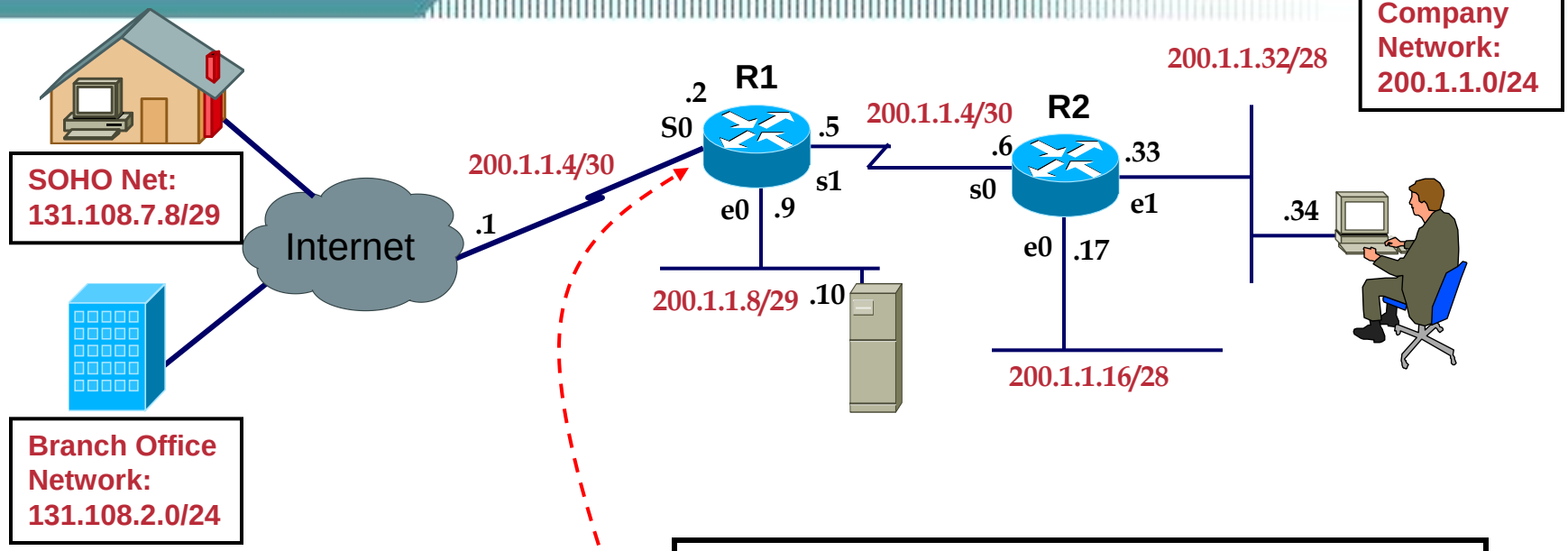
Access List Rules

Access Lists do nothing until applied

- **ACLs are processed "top-down"**
  - **First match for a given packet used, no further processing (until another packet arrives)**
- **Only one ACL can be applied…**
  - **Per protocol (IP, IPX, etc.)**
  - **Per direction**
  - **Per interface**

Question: ***How many access-lists could be applied if you had both IP and IPX configured on a single Ethernet interface?*** **Four**

Creating and Applying ACLs



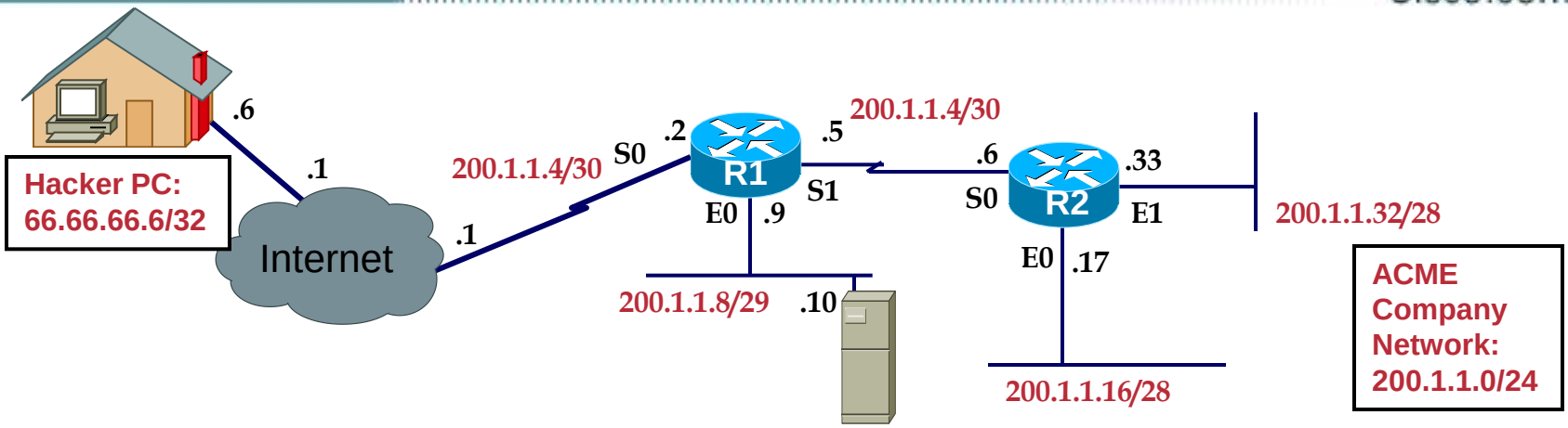

```
access-list 1 permit 131.108.7.8 0.0.0.7
access-list 1 permit 131.108.2.0 0.0.0.255
(access-list 1 deny any)

interface s0
  ip access-group 1 in
```

Things to keep in mind

- Processed "Top Down"
- Implicit "Deny Any" at end

Exercise: Hacker Attack!



ACME has *big* problem. A devious hacker has been detected trying to reach the ACME network. He needs to be denied access of any kind.

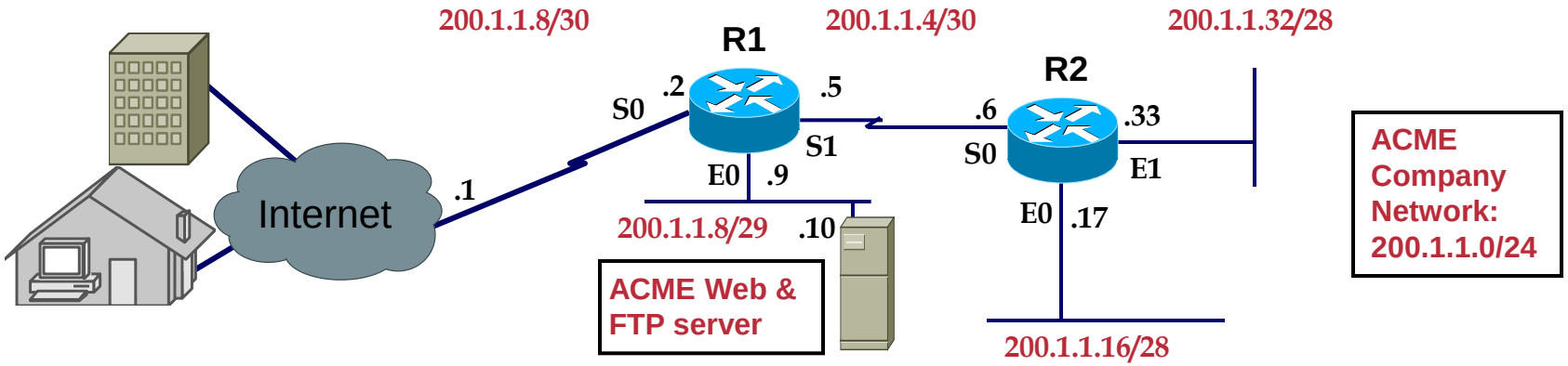
What type of Access List would you use? Why?

What will you deny? What will you permit?

Where will you place it?

- **Which Router?**
- **Which interface?**
- **Which direction?**

Exercise: No FTP for Them!



Various Internet users are attempting to open FTP sessions with ACME's Web server. It's creating too much load on the server. How would you resolve this problem, while still enabling Internet users access only to ACME's Web service?

What type of Access List would you use? Why?

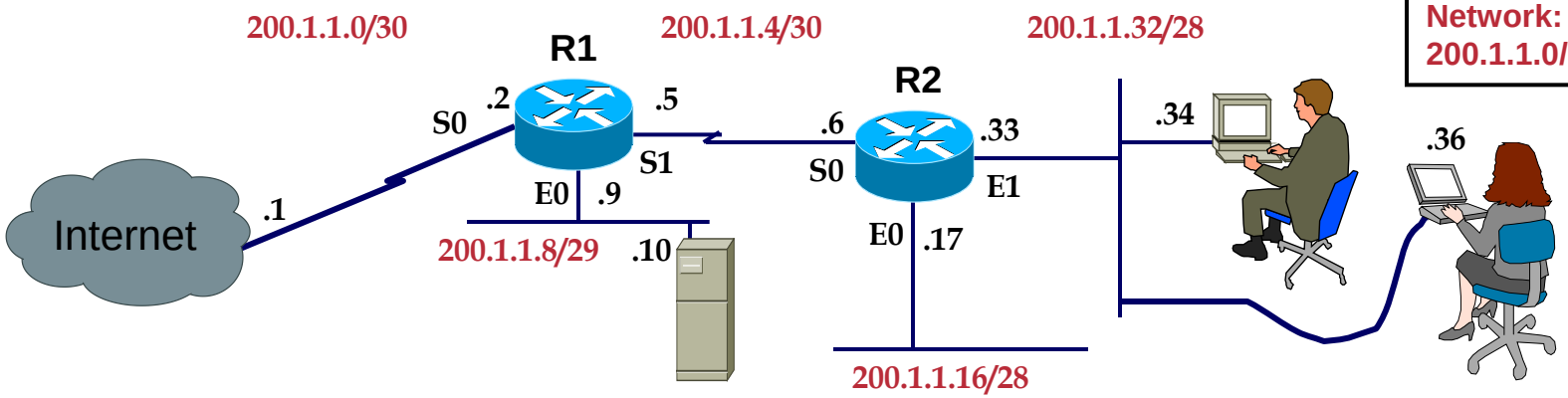
What will you deny? What will you permit?

Where will you place it?

- **Which Router?**
- **Which interface?**
- **Which direction?**

Exercise: New Kid on the Job

ACME Company Network: 200.1.1.0/24



A Junior network administrator has responsibility for R2, which is considered low-risk if he screws things up. Set up an ACL to allow him (.34) and his manager (.36) telnet access to R2.

What type of Access List would you use? Why?

What will you deny? What will you permit?

What is the least number of lines with which you could do this?

Where will you place it?

Which Router?

Which "interface"?

Which direction?

Access List Review

- **Access Lists could be**

  Inbound: Checks the filter condition before Routing table lookup

  Outbound: Checks the filter condition after Routing table lookup

- **To Configure Access List, we use Wildcard Masks**

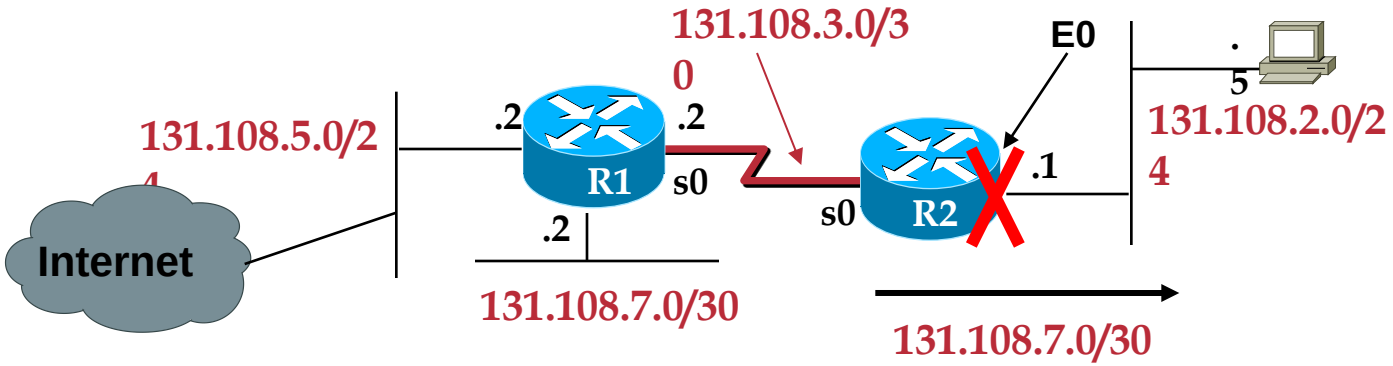
  Wildcard is the reverse of Netmask

  0 bit => must match bits in address

  1 bit => don't care. No need to match

Address	*Wildcard Mask*	*Match Condition*
0.0.0.0	255.255.255.255	All addresses will match ACL condition
131.54.0.0/16	0.0.255.255	Network 131.54.0.0
131.22.5.2/16	0.0.0.0	Only host 131.22.5.2 is permitted
131.111.8.0	0.0.0.7	Only subnet 131.111.8.0/29 is permitted
131.111.8.8	0.0.0.7	Only subnet 131.111.8.8/29 is permitted
131.111.8.15	0.0.0.3	Only subnet 131.111.8.15/30 is permitted

Access-List Example
Permit Telnet, FTP, to a Specific Host



ACL Objective for R2:

1. Deny all outbound traffic from network 131.108.7.0/30 from leaving interface Ethernet 0 on R2

2. Allow a specific host on the Internet (131.101.2.5) to have only FTP and Telnet access to an internal server located at 131.108.2.5.

Config:

```
access-list 101 deny ip 131.108.7.0 0.0.0.3 any

access-list 101 permit tcp host 131.101.2.5 host 131.108.2.5 eq ftp

access-list 101 permit tcp host 131.101.2.5 host 131.108.2.5 eq telnet
```

Apply:

```
interface e0

ip access-group 101 out
```

Monitoring Access Lists

- Show access list
- Show access list 110
- Show ip access list
- Show ip interface
- Show running config
- Show mac access-group