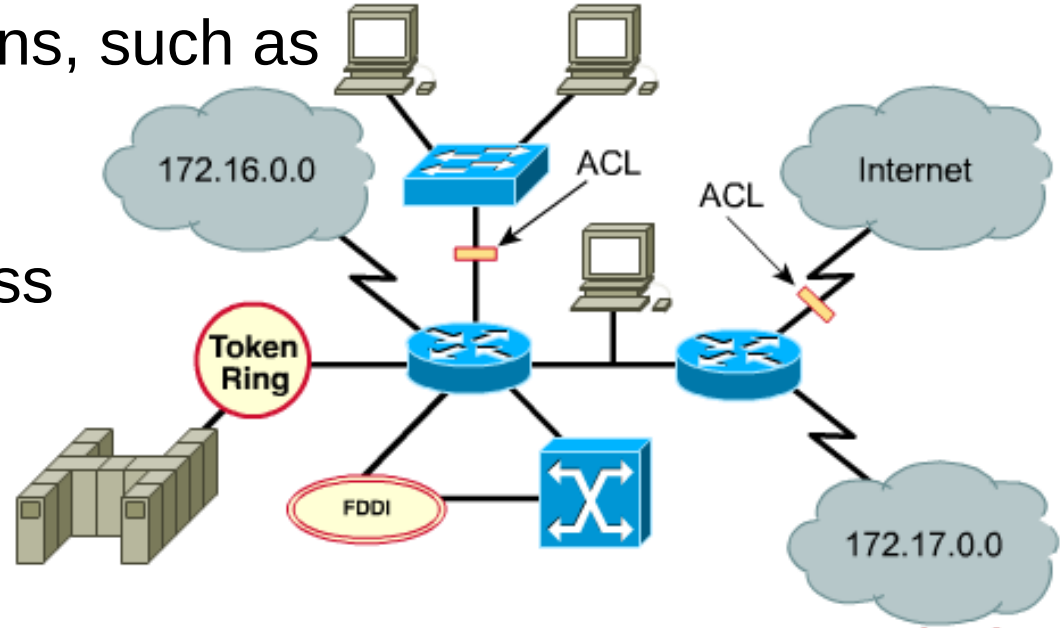
Understanding

Access Control Lists (ACLs)

Access Control Lists

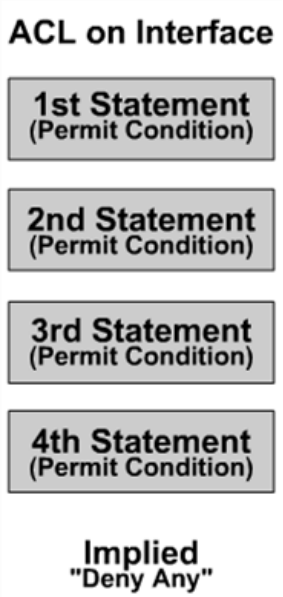
Access control lists (ACLs) are lists of instructions you apply to a router's interface. These lists tell the router what kinds of packets to accept and what kinds of packets to deny. Acceptance and denial can be based on certain specifications, such as

- source address
- destination address
- port number
- protocol



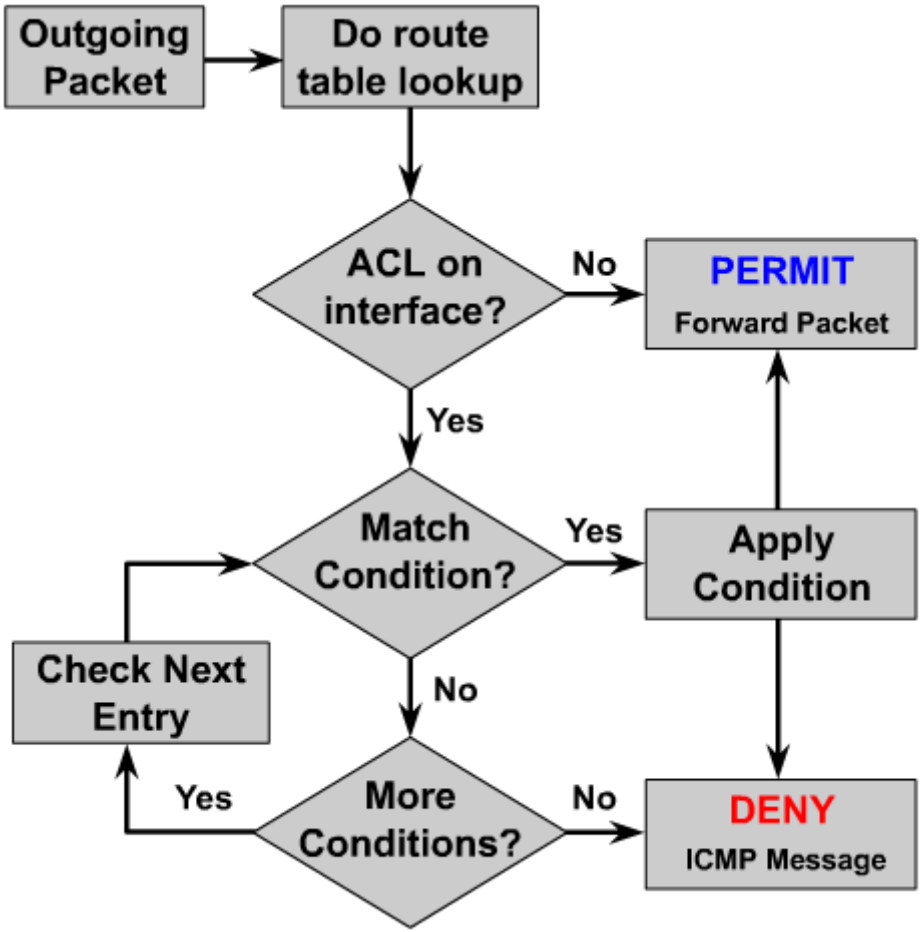
Access Control Lists

- The order in which you place ACL statements is important. When the router is deciding whether to forward or block a packet, the Cisco IOS software tests the packet against each condition statement, in the order in which the statements were created (sequential order).
- After a match is found, no more condition statements are checked.
- If ACL are used and a packet does not match any of the test in the access list, it is discarded. This is because an implicit "deny any" statement is imposed.
- **There will always be a TRUE (matched) statement.**



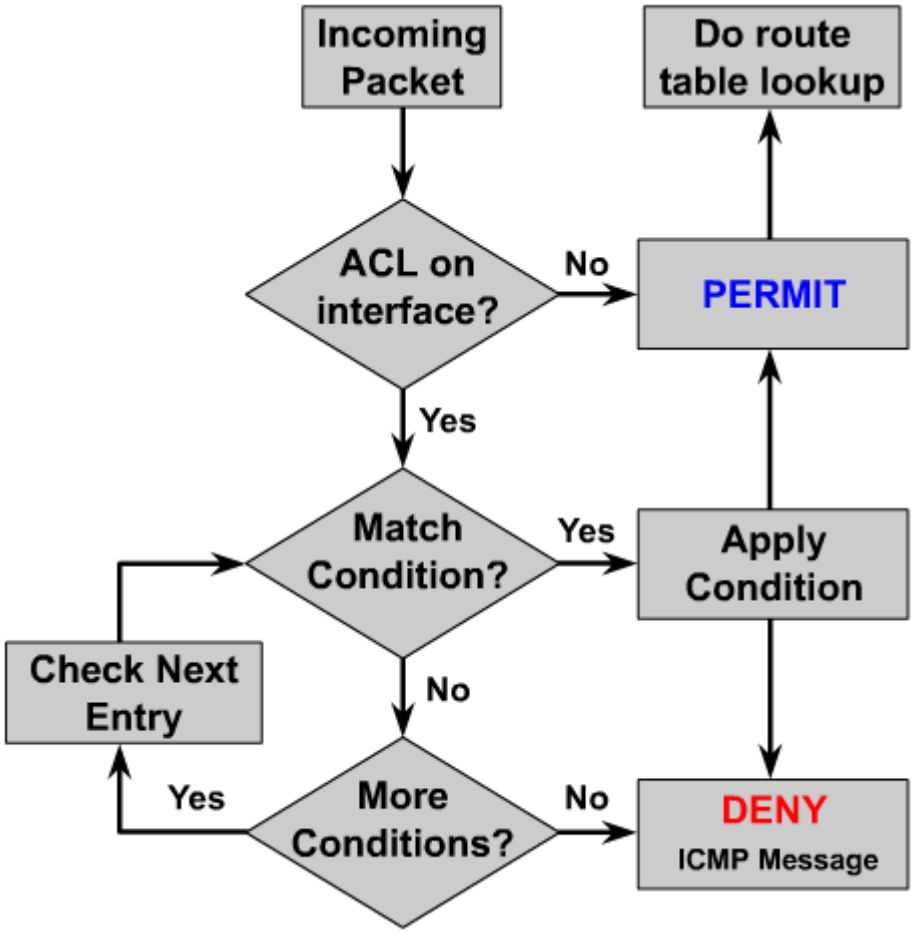
How a Router Uses an ACL (outbound)

- Check to see if packet is routable. If so, look up route in routing table.
- Check for an ACL for the outbound interface
  - If no ACL, switch the packet out the destination interface
  - If an ACL, check the packet against the ACL statements sequentially--denying or permitting based on a matched condition.
- If no statement matches, what happens?

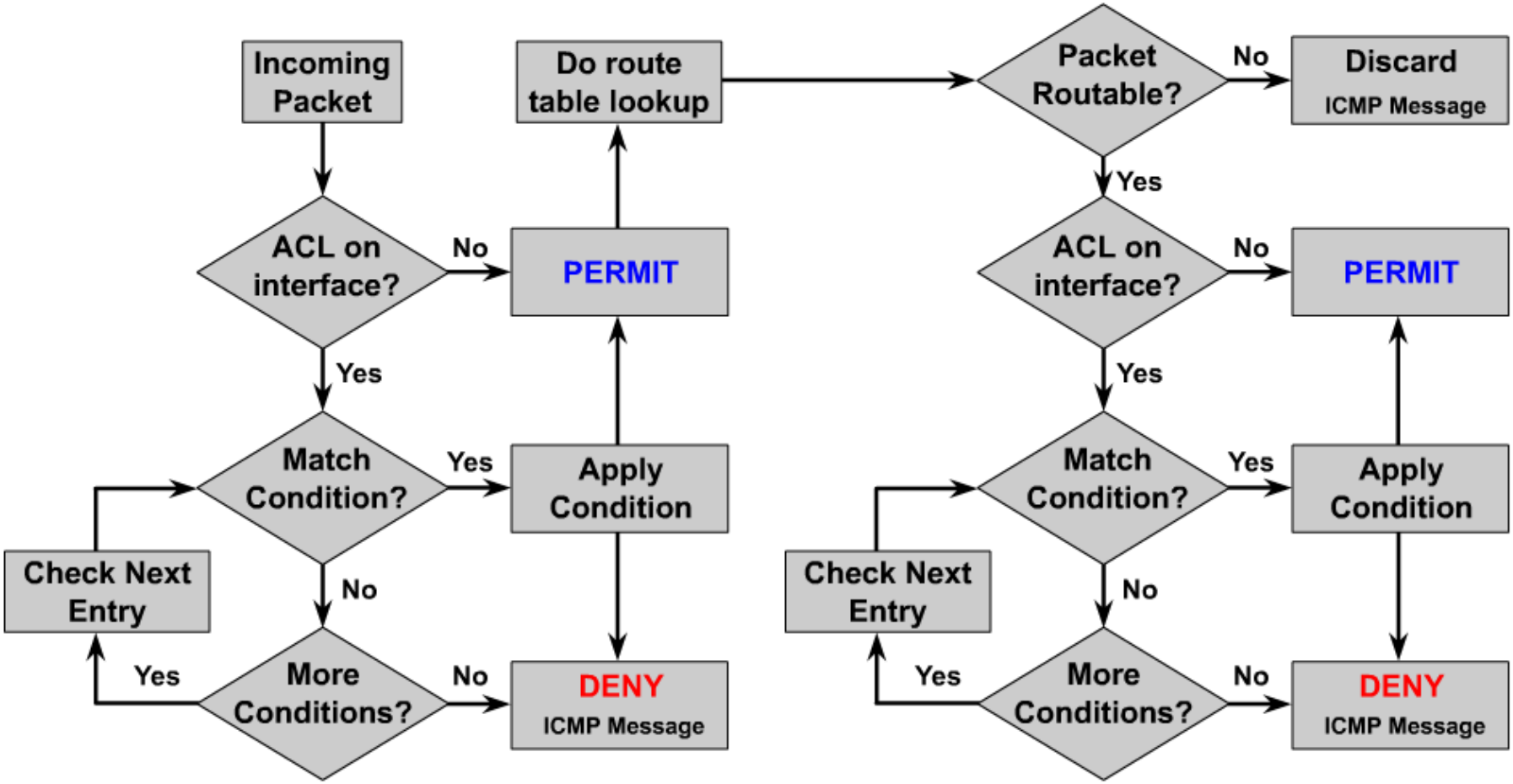


How a Router Uses an ACL (inbound)

- If an ACL is configured to filter inbound traffic, the route table lookup is done only if the packet is permitted.



ACL Processing Inbound and Outbound



Basic ACL Rules

- One ACL per protocol, per interface, per direction. Write your ACLs carefully, since you have to include all traffic that should be filtered inbound or outbound in a single ACL!
- Standard ACLs should be applied closest to the destination. Extended ACLs should be applied closest to the source.
- Statements are processed sequentially until a match is found, if no match is found then the packet is denied (implied “deny any”).
- Specific hosts should be filtered first, and groups or general filters should come last.
- ACLs are assigned to one or more interfaces and can filter inbound traffic or outbound traffic.
  - Outbound ACLs are generally more efficient than inbound, and are therefore preferred.
  - Inbound ACL must check every packet to see whether it matches the ACL condition before switching the packet to an outbound interface.

Basic ACL Rules

- Never work with an access list that is actively applied. Use a text editor first.
- New lines are always added to the end of the access list. It is not possible to selectively add and remove lines.
- An IP access list will send an ICMP host unreachable message to the sender of the rejected packet.
- Outbound filters do not affect traffic originating from the local router.

ACLs

When configuring ACLs on a router, you must identify each ACL uniquely by assigning a number

The number must be within the specific range of numbers that is valid for the protocol.

Protocol	Range
IP	1-99
Extended IP	100-199
AppleTalk	600-699
IPX	800-899
Extended IPX	900-999
IPX Service Advertising Protocol	1000-1099

ACLs

Configuring an ACL is a two step process:

1. Configure the access list to permit or deny packets based upon the test condition established in the access list (define the test conditions)
2. Apply the access list to one or more interfaces

An ACL does not take action on a packet until it has been applied to an interface.

Wildcard Mask

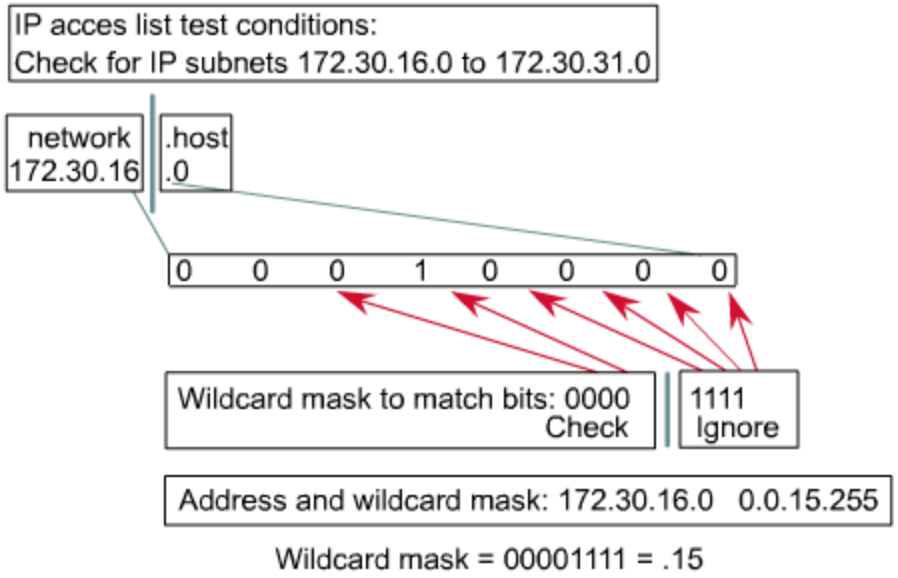
A wildcard mask is a 32-bit quantity that is divided into four octets, with each octet containing 8 bits. The wildcard bits determine which address bits are examined.

- A wildcard mask bit 0 means "check the corresponding bit value”
- A wildcard mask bit 1 means "do not check (ignore) that corresponding bit value"

Wildcard Mask

Test an IP address for subnets that will be permitted or denied.

Assume that the IP address is a Class B address with 8 bits of subnetting. You want to use IP wildcard mask bits to permit all packets from any host in the 172.30.16.0 to 172.30.31.0 subnets.



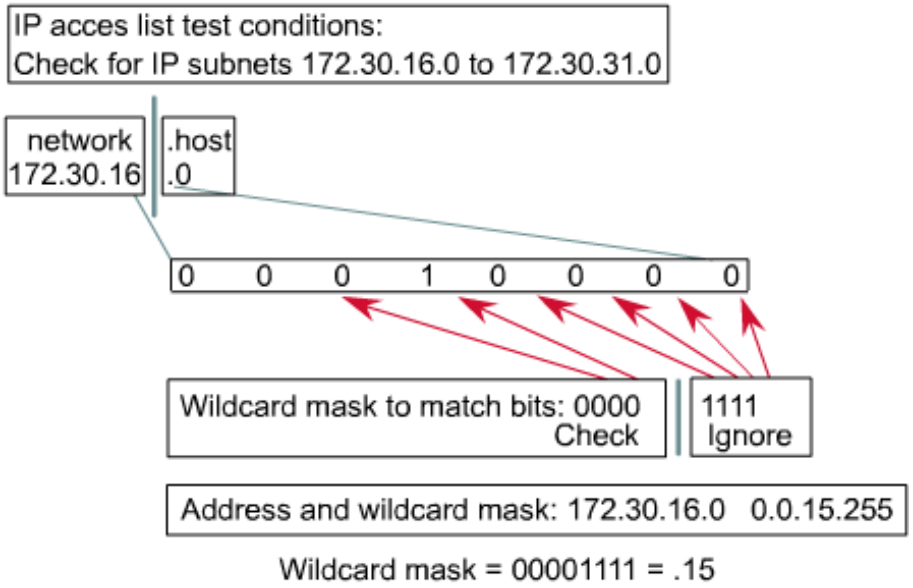
Wildcard Mask

Test an IP addresses for subnets that will be permitted or denied.
(third octet values)
Wildcard Mask = 0.0.15.255

	128	64	32	16	8	4	2	1	Match Y / N
		Match				Ignore			
WC	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	
Test	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	
15	0	0	0	0	1	1	1	1	**N**
18	0	0	0	1	0	0	1	0	**Y**
25	0	0	0	1	1	0	0	1	**Y**
31	0	0	0	1	1	1	1	1	**Y**

Wildcard Mask

In the third octet, the wildcard mask is 15 (00001111), and the IP address is 16 (00010000). The first four zeros in the wildcard mask tell the router to match the first four bits of the IP address (0001).



Because the last four bits are ignored, all numbers in the range of 16 (00010000) to 31 (00011111) will match because they begin in the pattern 0001.

Wildcard Mask

If you wanted to deny traffic only on the 203.49.23.0 network, what would the wildcard mask be?

0.0.0.255

If you wanted to deny traffic only on the 158.64.0.0 network, what would the wildcard mask be?

0.0.255.255

If you wanted to deny traffic only on the 102.0.0.0 network, what would the wildcard mask be?

0.255.255.255

Wildcard Mask

Given a IP address of 199.25.8.0 with a subnet mask of 255.255.255.224, what wildcard mask would be used to deny traffic from subnet #1?

0.0.0.31

Given a IP address of 199.25.32.0 with a subnet mask of 255.255.255.240, what wildcard mask would be used to deny traffic from subnet #2?

0.0.0.15

What about subnet #3?

0.0.0.15

ANY Command

For the most common uses of wildcard masking, you can use abbreviations.

- If you want to specify that any destination address will be permitted in an ACL test, you would enter 0.0.0.0.
- To indicate that the ACL should ignore any value, the corresponding wildcard mask bits for this address would be all ones (that is, 255.255.255.255).
- You can use the abbreviation **any** to communicate this same test condition to Cisco IOS ACL software.

ANY Command

Below is an example an access list with and without the **any** command.

```
Router(config)# access-list 1 permit
0.0.0.0 255.255.255.255
```

```
Router(config)# access-list 1 permit any
```

Both command do the same thing.

HOST Command

If you want to specify that a specific IP host address will be permitted in an ACL test, you would enter the full address (for example, 172.30.16.29).

- Then, to indicate that the ACL should check all the bits in the address, the corresponding wildcard mask bits for this address would be all zeros (that is, 0.0.0.0).
- You can use the abbreviation **host** to communicate this same test condition to Cisco IOS ACL software.

```
Router(config)# access-list 1 permit
172.30.16.29 0.0.0.0
```

OR

```
Router(config)# access-list 1 permit host
172.30.16.29
```

Types of ACLs

There two basic types of ACLs:

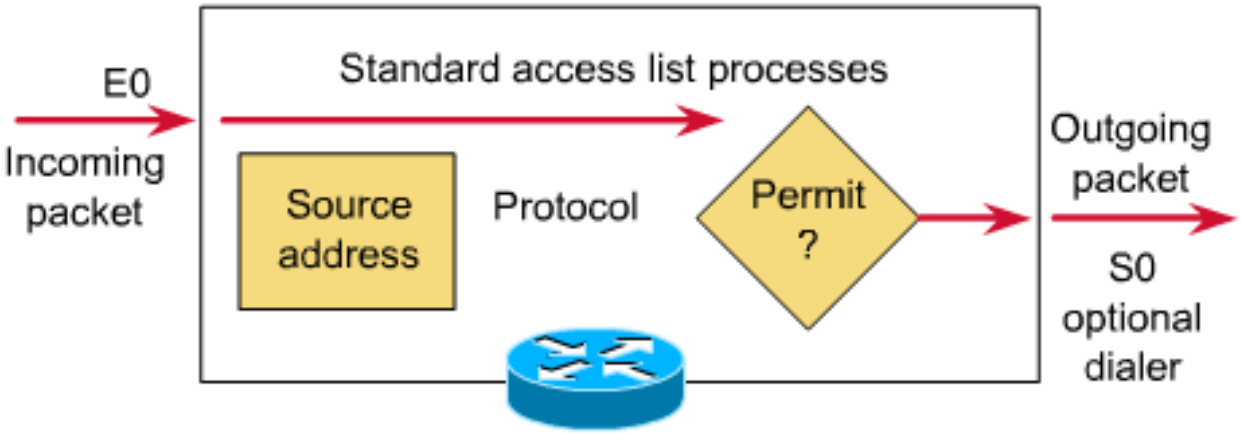
- Standard ACLs
  - -Simpler address specifications
  - -Permits or denies an entire protocol suite
  - -IP standard ACLs are numbered 1 - 99
- Extended ACLs
  - -More complex address specifications
  - -Permits or denies specific parts of a protocol suite
  - -IP extended ACLs are numbered 100 - 199

Standard ACLs

You use standard ACLs when:

- You want to block or allow all traffic from a specific network
- Deny protocol suites (Standard ACLs permit or deny the entire IP suite.)

Standard ACLs check the source address of packets that could be routed.



Standard ACLs Syntax

The full syntax of the command is:

```
Router(config)# access-list access-list-number {deny | permit} source [source-wildcard ]
```

Example-

```
Router(config)# access-list 1 permit 198.3.5.0 0.0.0.255
```

You use the **no** form of this command to remove a standard ACL. This is the syntax:

Example-

```
Router(config)# no access-list 1 permit 198.3.5.0 0.0.0.255
```

Standard ACLs Syntax

The command **log** may be added to the end of the ACL.

```
Router(config)# access-list access-list-number {deny | permit} source [source-wildcard ] [log]
```

Example-

```
Router(config)# access-list 1 permit 198.3.5.0 0.0.0.255 log
```

This command keeps track of how many packets were permitted or denied based on an access list test condition. It includes the ACL number, whether the packet was permitted or denied, the source address, and the number of packets.

Standard ACLs

What do the following ACLs do?

```
Router(config)# access-list 33 permit 172.16.0.0
0.0.255.255
```

```
Router(config)# access-list 44 deny 172.16.13.7
0.0.0.0 log
```

How else could this command be written?

access-list 44 deny host 172.16.13.7

Standard ACLs

Write an access list that denies all traffic from the 196.39.220.0 network.

access-list 1 deny 196.39.220.0 0.0.0.255

access-list 1 permit 0.0.0.0 255.255.255.255

Write an access list that denies all traffic from the host 208.200.55.68.

access-list 2 deny host 208.200.55.68

access-list 2 permit any

Write an access list that permits only traffic from the 139.220.0.0 network.

access-list 3 permit 139.220.0.0 0.0.255.255

Standard ACLs

Write an access list that permits only traffic from the 119.0.0.0 network.

access-list 4 permit 119.0.0.0 0.255.255.255

Write an access list that denies all traffic from the host 22.20.10.6.

access-list 5 deny host 22.20.10.6

access-list 5 permit any

Write an access list that permits all traffic except from the 139.22.0.0 network.

access-list 6 deny 139.22.0.0 0.0.255.255

access-list 6 permit any

Applying an ACL

To apply an ACL, the **`ip access-group`** command is used. This command groups an existing ACL to an interface (applies it to the interface).

Remember that only one ACL per port per protocol per direction is allowed.

The format of the command is:

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

Example-

```
Router(config)# interface e0
Router(config-if)#ip access-group 5 in
```

*These commands are the same for standard or extended ACLS

Applying an ACL

Example-

```
Router(config)# interface e0
Router(config-if)#ip access-group 5 in
```

The **in** / **out** portion of the command is from the perspective of the router. From the router's view point, are you permitting / denying packets that are coming in (**in**) or packets that going out (**out**).

Note: If **in** / **out** is not is not specified, the default setting is **out.**

Applying a Standard ACL

- When applying a standard ACL, you should apply it as close to the destination as possible.
- When applying an extended ACL, you should apply it as close to the source as possible.
- ACLs can be applied to both inbound and outbound ports.

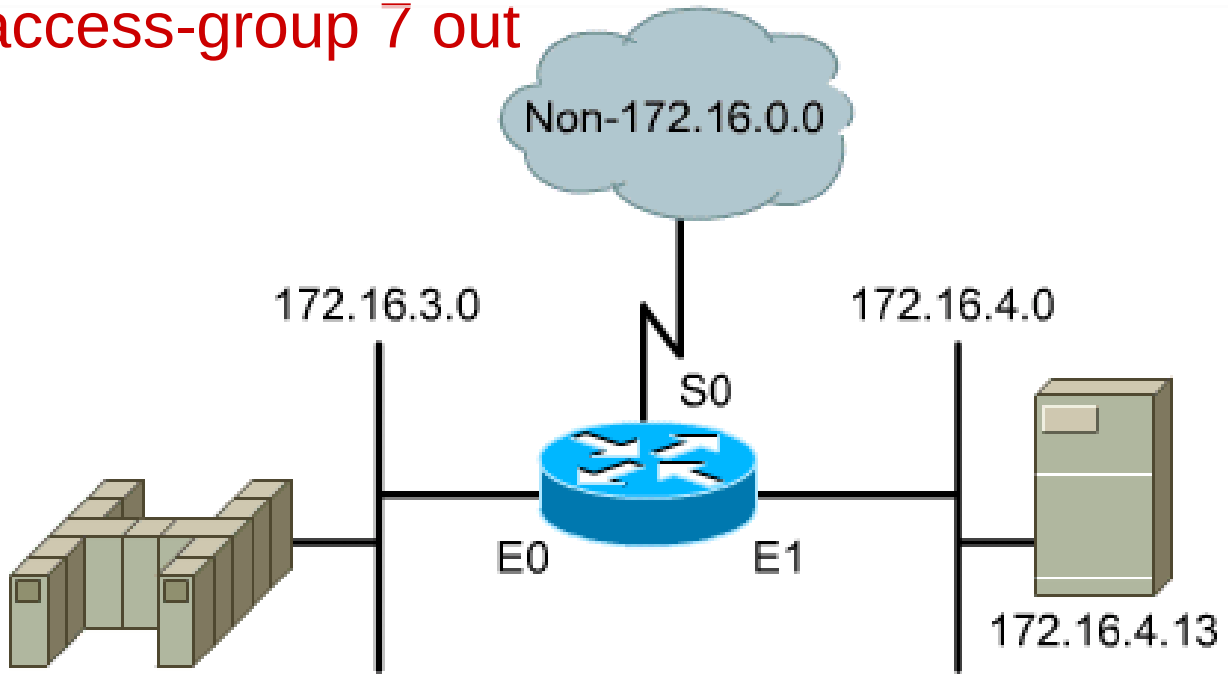
Applying a Standard ACL

Write an ACL that permits only traffic from the 172.16.0.0 network to pass through this router and apply the ACL.

access-list 7 permit 172.16.0.0 0.0.255.255

int s0

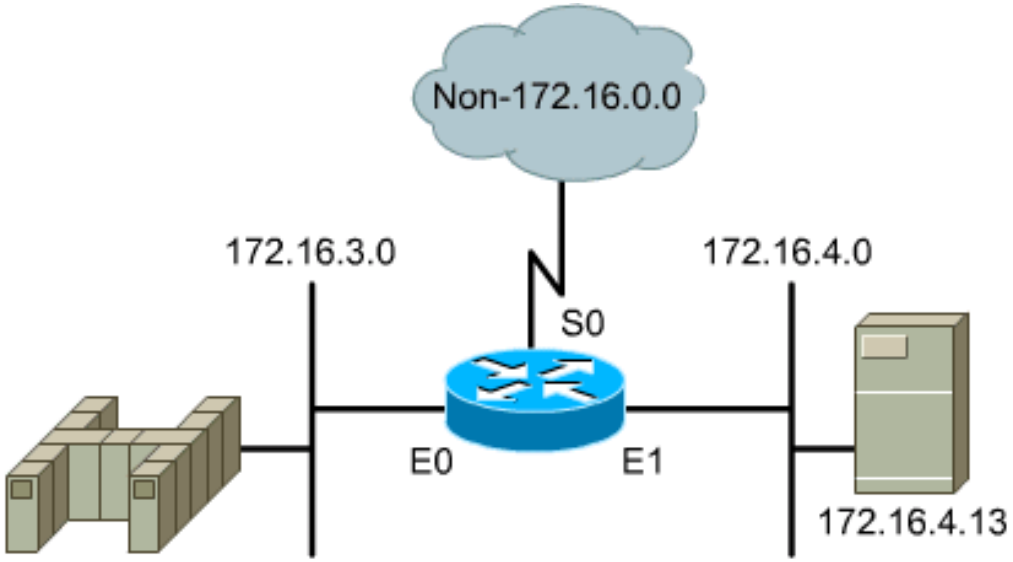
ip access-group 7 out



Applying an ACL

Using the topology below and given the following ACL, what traffic will be allowed to pass through the router?

```
Router(config)# access-list 5 deny host
172.16.3.12
Router(config)# interface e0
Router(config-if)#ip access-group 5 in
```



Extended ACLs

Extended ACLs are used most often to test conditions because they provide a greater range of control than standard ACLs.

- Check for both source and destination packet addresses
- Check for specific protocols, port numbers, and other parameters.
- Extended IP ACLs use a number in the range 100 to 199.

Extended ACL Syntax

The complete form of the extended `access-list` command is:

```
Router(config)# access-list access-list-number {permit | deny}
protocol source [source-mask] destination [destination-mask] [operator operand]
```

Example-

```
Router(config)# access-list 100 deny tcp 199.2.4.0 0.0.0.255 202.34.5.0 0.0.0.255 eq 23
```

OR

```
Router(config)# access-list 100 deny tcp 199.2.4.0 0.0.0.255 202.34.5.0 0.0.0.255 eq telnet
```

Extended ACL Syntax

Common protocols used in extended ACLs:

- ip
- tcp
- udp
- icmp

Common operators used in extended ACLs:

- eq (equals)
- lt (less than)
- gt(greater than)
- neq (not equal)

The `established` option

- By adding the keyword **established** to the ACL statement, you are requiring that the TCP or UDP session must be established.
- For example, allowing hosts behind your firewall to establish connections with outside hosts.
  - Outside hosts can send packets back to the source only if the source initiated the session.

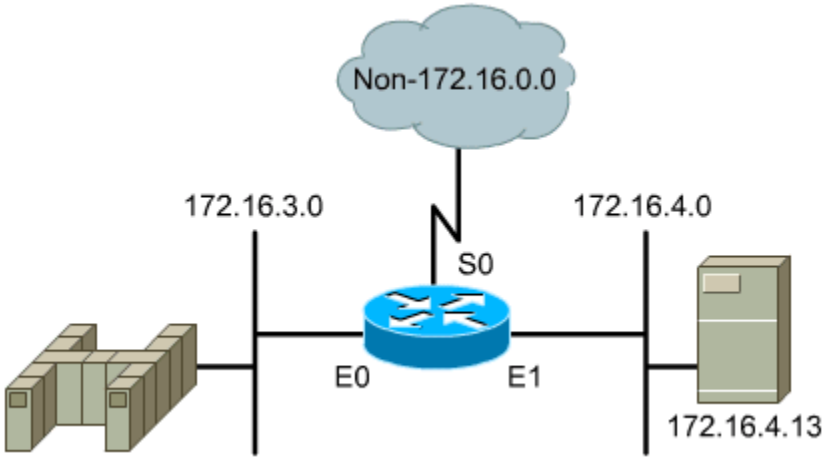
Extended ACLs

What does the following ACL do?

access-list 100 deny tcp 172.16.3.0 0.0.0.255 172.16.4.0 0.0.0.255 eq 53
access-list 100 permit ip any any

Where would you apply this ACL?

interface e0
ip access-group 100 in



Extended ACL

Network

Write an ACL that will block only WWW traffic from the network 174.32.0.0 into any other network.

access-list 101 deny tcp 174.32.0.0 0.0.255.255 any eq 80

access-list 101 permit ip any any

Write an ACL that will allow only WWW traffic into the network 174.32.0.0 from any other network.

access-list 102 permit tcp any 174.32.0.0 0.0.255.255 eq 80

Extended ACL

Write an ACL that will permit only Telnet traffic from the network 14.4.0.0/15 into any other network.

access-list 103 permit tcp 14.4.0.0 0.1.255.255 any eq 23

Write an ACL that will deny only FTP and TFTP traffic into the network 174.32.0.0 from any other network.

access-list 104 deny tcp any 174.32.0.0 0.0.255.255 eq 21

access-list 104 deny udp any 174.32.0.0 0.0.255.255 eq 69

access-list 104 permit ip any any

Extended ACL

Subnet

Write an ACL that will block only telnet traffic from the subnet 174.32.5.32 / 27 to the 182.47.0.0 network.

access-list 105 deny tcp 174.32.5.32 0.0.0.31 182.47.0.0 0.0.255.255 eq 23
access-list 105 permit ip any any

Extended ACL

Subnet

Write an ACL that will permit only telnet traffic from the subnet 174.32.5.32/28 to the subnet 102.47.64.0/18 network.

access-list 106 permit tcp 174.32.5.32 0.0.0.15 102.47.64.0 0.0.63.255 eq 23

Extended ACL

Subnet - High / Low Range

Write an ACL that will deny only WWW traffic from the the lower half of subnet 174.32.5.32 / 27 to the 182.47.0.0 network.

access-list 101 deny tcp 174.32.5.32 0.0.0.15 182.47.0.0 0.0.255.255 eq 80
access-list 101 permit ip any any

	128	64	32	16	8	4	2	1	Match Y / N
WC	0	0	0	0	1	1	1	1	
Test	0	0	1	0	0	0	0	0	
33	0	0	1	0	0	0	0	1	Y
41	0	0	1	0	1	0	0	1	Y
47	0	0	1	0	1	1	1	1	Y
48	0	0	1	1	0	0	0	0	N
50	0	0	1	1	0	0	1	0	N
63	0	0	1	1	1	1	1	1	N

Extended ACL

Subnet - High / Low Range

Write an ACL that will deny only WWW traffic from the the upper half of subnet 174.32.5.32 / 27 to the 182.47.0.0 network.

access-list 101 deny tcp 174.32.5.48 0.0.0.15 182.47.0.0 0.0.255.255 eq 80
access-list 101 permit ip any any

	128	64	32	16	8	4	2	1	Match Y / N
WC	0	0	0	0	1	1	1	1	
Test	0	0	1	1	0	0	0	0	
33	0	0	1	0	0	0	0	1	N
41	0	0	1	0	1	0	0	1	N
47	0	0	1	0	1	1	1	1	N
48	0	0	1	1	0	0	0	0	Y
50	0	0	1	1	0	0	1	0	Y
63	0	0	1	1	1	1	1	1	Y

Extended ACL

Subnet - Even / Odd

Write an ACL that will deny only WWW traffic from the odd numbered addresses in the subnet 174.32.5.32 / 27 to the 182.47.0.0 network.

access-list 121 deny tcp 174.32.5.33 0.0.0.254 182.47.0.0 0.0.255.255 eq 80
access-list 121 permit ip any any

	128	64	32	16	8	4	2	1	Match Y / N
WC	1	1	1	1	1	1	1	0	
Test	0	0	0	0	0	0	0	1	
1	0	0	0	0	0	0	0	1	Y
2	0	0	0	0	0	0	1	0	N
8	0	0	0	0	1	0	0	0	N
11	0	0	0	0	1	0	1	1	Y
50	0	0	1	1	0	0	1	0	N
63	0	0	1	1	1	1	1	1	Y

Extended ACL

Subnet - Even / Odd

Write an ACL that will deny only WWW traffic from the even numbered addresses in the subnet 174.32.5.32 / 27 to the 182.47.0.0 network.

access-list 121 deny tcp 174.32.5.32 0.0.0.254 182.47.0.0 0.0.255.255 eq 80
access-list 121 permit ip any any

	128	64	32	16	8	4	2	1	Match Y / N
WC	1	1	1	1	1	1	1	0	
Test	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	1	N
2	0	0	0	0	0	0	1	0	Y
8	0	0	0	0	1	0	0	0	Y
11	0	0	0	0	1	0	1	1	N
50	0	0	1	1	0	0	1	0	Y
63	0	0	1	1	1	1	1	1	N

Naming ACLs

- One nice feature in IOS versions 11.2 and later is the ability to name ACLs. Advantages include:
  - Intuitive names that can possibly identify the purpose of the ACL.
  - Unlimited ACLs; numbered ACLs have a limited range.
  - Ability to delete entries without having to rewrite the entire ACL; additions are added to the end of the list, just like numbered ACLs
  - Reduced amount of typing; no need to type access-list and *access-list-number* for each statement.

Syntax for Naming an ACL

```
Router(config)#ip access-list standard|extended} name
```

- Router prompt changes to ACL configuration mode.
  - Now you can simply start each statement with the `permit` or `deny` argument.

```
RTB(config)#ip access-list standard MY_ACL1
RTB(config-std-nacl)#deny host 192.168.3.25
RTB(config-std-nacl)#permit 192.168.3.0 0.0.0.255
RTB(config-std-nacl)#exit
RTB(config)#interface e0
RTB(config-if)#ip access-group MY_ACL1 in
!
RTB(config)#ip access-list extended MY_ACL2
RTB(config-ext-nacl)#permit tcp any host 192.168.3.39 eq www
RTB(config-ext-nacl)#permit tcp any host 192.168.3.39 eq ftp
RTB(config-ext-nacl)#permit icmp any any
RTB(config-ext-nacl)#exit
RTB(config)#interface e0
RTB(config-if)#ip access-group MY_ACL2 out
```

Verifying ACLs

- **`show access-lists`**
  - shows all access-lists configured on the router
- **`show [protocol] access-lists {name|number}`**
  - shows the identified access list
- **`show ip interface`**
  - shows the access-lists applied to the interface--both inbound and outbound.
- **`show running-config`**
  - shows all access lists and what interfaces they are applied on

Adding Remarks to ACLs

```
Router(config)#access-list access-list number
                    remark remarks
Router(config-std-nacl)#remark remarks
Router(config-ext-nacl)#remark remarks
```

- The remark command allows you to make comments within your ACL configuration to document the active configuration.
- Remarks are displayed when you issue the show run command. However, you will NOT see remarks displayed with the show access-list command.

Example Using the `remark` Command


```
RTA(config)#access-list 100 remark Deny Charlie web access
RTA(config)#access-list 100 deny tcp host 192.168.1.5 any eq 80
RTA(config)#access-list 100 permit ip any any

RTA#show access-list
Extended IP access list 100
    deny tcp host 192.168.1.35 any eq www
    permit ip any any

!--The "show access-list" command does NOT display remarks
!--Use "show run" to see remarks.

RTA#show run
<output omitted>
```

ACLs in a Nutshell

- Order of the ACL statements is critical
- Generally place the more specific statement first, then move to the more general statement
- There is an implicit deny any as the last statement in all ACLs
- If the last statement written is a deny, the next statement is generally a permit any
- If the last statement written is a permit, the next statement is generally a deny any
- After the ACL is written, it must be applied before it will take affect

ACLs in a Nutshell

- Apply a standard ACL as close to the destination as possible (standar**d**estination)
- Apply an extended ACL as close to the destination as possible (sourc**e**xtended)
- To remove a single ACL, type **no access-list [#]**
- To remove all ACLs, type **no access-lists**
- To monitor ACLs
  - **show ip interface**
  - **show run**
  - **show access-lists**
  - **show access-list [#]**

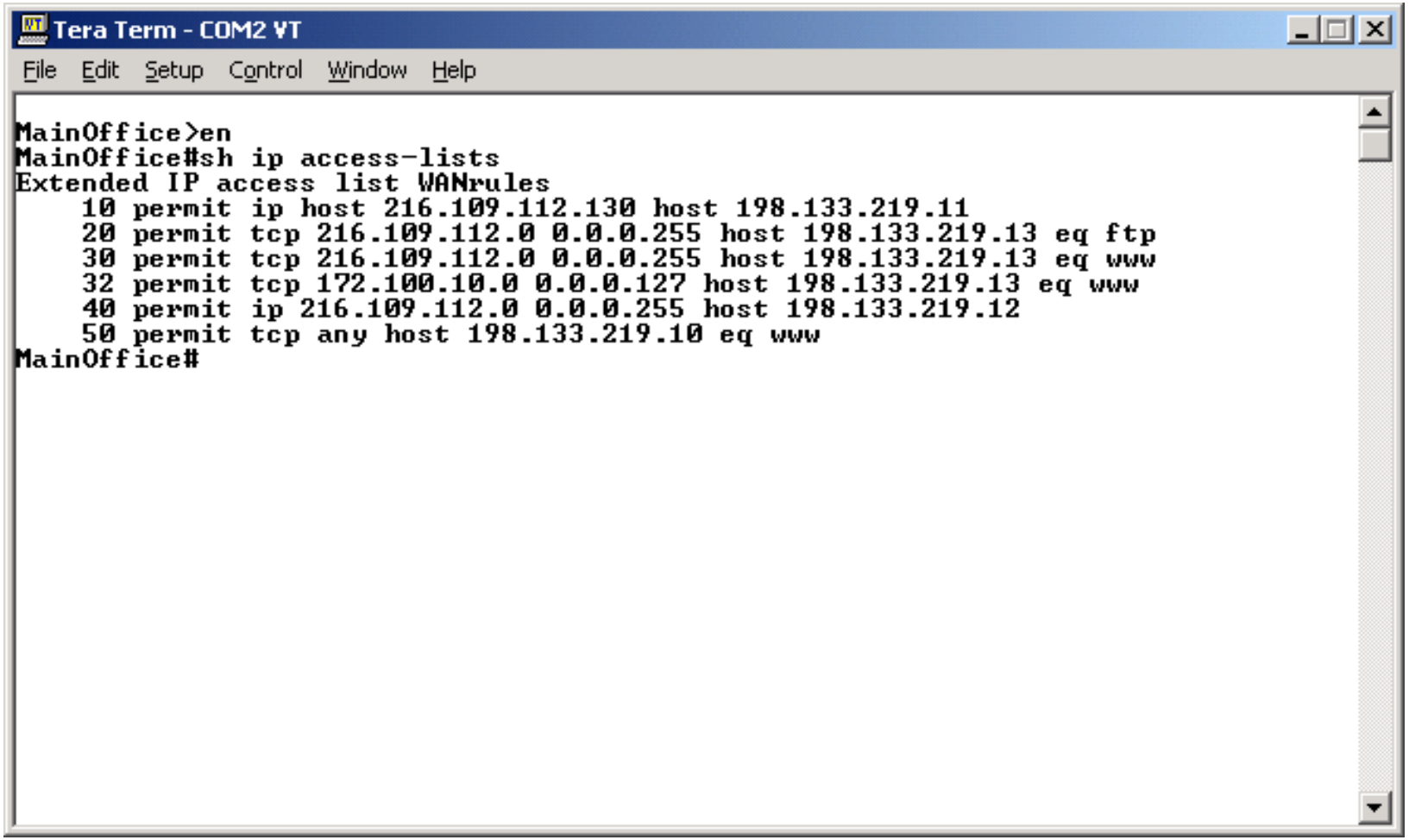
IP Access Control List (ACL)

Sequence Numbering

Named ACLs with Sequence Numbering

- When first introduced, named ACLs allowed only for statements to be inserted at the end of a list.
- However, the addition of Entry Sequence Numbering allows for insertion and reordering of already created and named ACLs.
- In other words, permit or deny statements can be added at any point within an existing ACL. Also allows selected deletion of an entry by referencing the entry sequence number.
- The ability to apply sequence numbers to IP access list entries makes it possible to define the exact position of a new entry in an already created named ACL.
- Prior to the IP Access List Entry Sequence Numbering feature, new entries were always added to the end of the configured statements.

What does a sequence number look like?



Commands and Configuration

Viewing Sequence Numbers

- To ensure backward compatibility, ACL sequence numbers are not saved and cannot be viewed using the `show run` or `show start` commands.
- To view the current sequence numbers for ACLs, use one of the following commands:

```
router# show access-lists
router# show access-lists list name
router# show ip access-lists
router# show ip access-lists list name
```

- Since the sequence numbers are not saved, a system reload will cause all ACLs to be renumbered using the defaults: first line will start at 10 and each additional line will be incremented by 10.

Default Sequence Numbering

- When an entry with no sequence number is entered, by default it has a sequence number of 10 more than the last entry in the access list.

```
Router# show access-lists 150
Extended IP access list 150
    10 permit icmp any any
    20 permit tcp any host 10.3.3.3
    30 permit ip host 10.4.4.4 any
    40 permit ip host 10.3.3.3 any log
    50 permit ip any any
```

Adding Entries with Sequence Numbers

- View current sequence numbers

```
Router# show ip access-lists
Standard IP access list tryit
    10 permit 10.10.10.0, wildcard bits 0.0.0.255
    20 permit 10.10.20.0, wildcard bits 0.0.0.255
```

- Enter ACL configuration mode

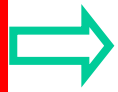
```
Router(config)# ip access-list standard tryit
```

- Add a numbered entry.

```
Router(config-std-nacl)# 15 permit 10.5.5.0 0.0.0.255
```

- View new entry.

```
Router# show ip access-list
Standard IP access list tryit
    10 permit 10.10.10.0, wildcard bits 0.0.0.255
    15 permit 10.5.5.0, wildcard bits 0.0.0.255
    20 permit 10.10.20.0, wildcard bits 0.0.0.255
```



Adding Entries without Sequence Numbers

- View current sequence numbers

```
Router# show ip access-lists
Standard IP access list tryit
    10 permit 10.10.10.0, wildcard bits 0.0.0.255
    20 permit 10.10.20.0, wildcard bits 0.0.0.255
```

- Enter ACL configuration mode

```
Router(config)# ip access-list standard tryit
```

- Add an unnumbered entry.

```
Router(config-std-nacl)# permit 10.5.5.0 0.0.0.255
```

- View new entry added at the end.

```
Router# show ip access-list
Standard IP access list tryit
    10 permit 10.10.10.0, wildcard bits 0.0.0.255
    20 permit 10.10.20.0, wildcard bits 0.0.0.255
    30 permit 10.5.5.0, wildcard bits 0.0.0.255
```

Removing Entries

- ACL entries can be removed by identifying the sequence number.
- View the ACL to identify the line number to remove.

```
Router# show access-lists 150
Extended IP access list 150
    10 permit icmp any any
    20 permit tcp any host 10.3.3.3
    30 permit ip host 10.4.4.4 any
    40 permit ip any any
```

- From ACL configuration mode, use the no command followed by the sequence number.

```
Router(config-std-nacl)# no 30
```

- View the ACL to confirm that the line was removed.

```
Router# show access-lists 150
Extended IP access list 150
    10 permit icmp any any
    20 permit tcp any host 10.3.3.3
    40 permit ip any any
```

Resequencing

- This example shows resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values that users provide.

```
Router(config)# ip access-list resequence
  george 1 2
Router# show access-list george
Extended IP access list george
     1 permit icmp any any
     3 permit tcp any host 10.3.3.3
     5 permit ip host 10.4.4.4 any
     7 permit ip host 10.3.3.3 any log
     9 permit ip any any
```

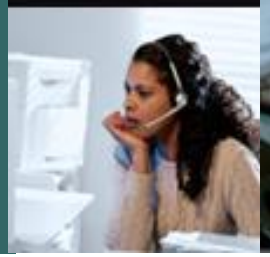
IOS Support for ACL Sequence Numbering

Which images support Sequence Numbering?

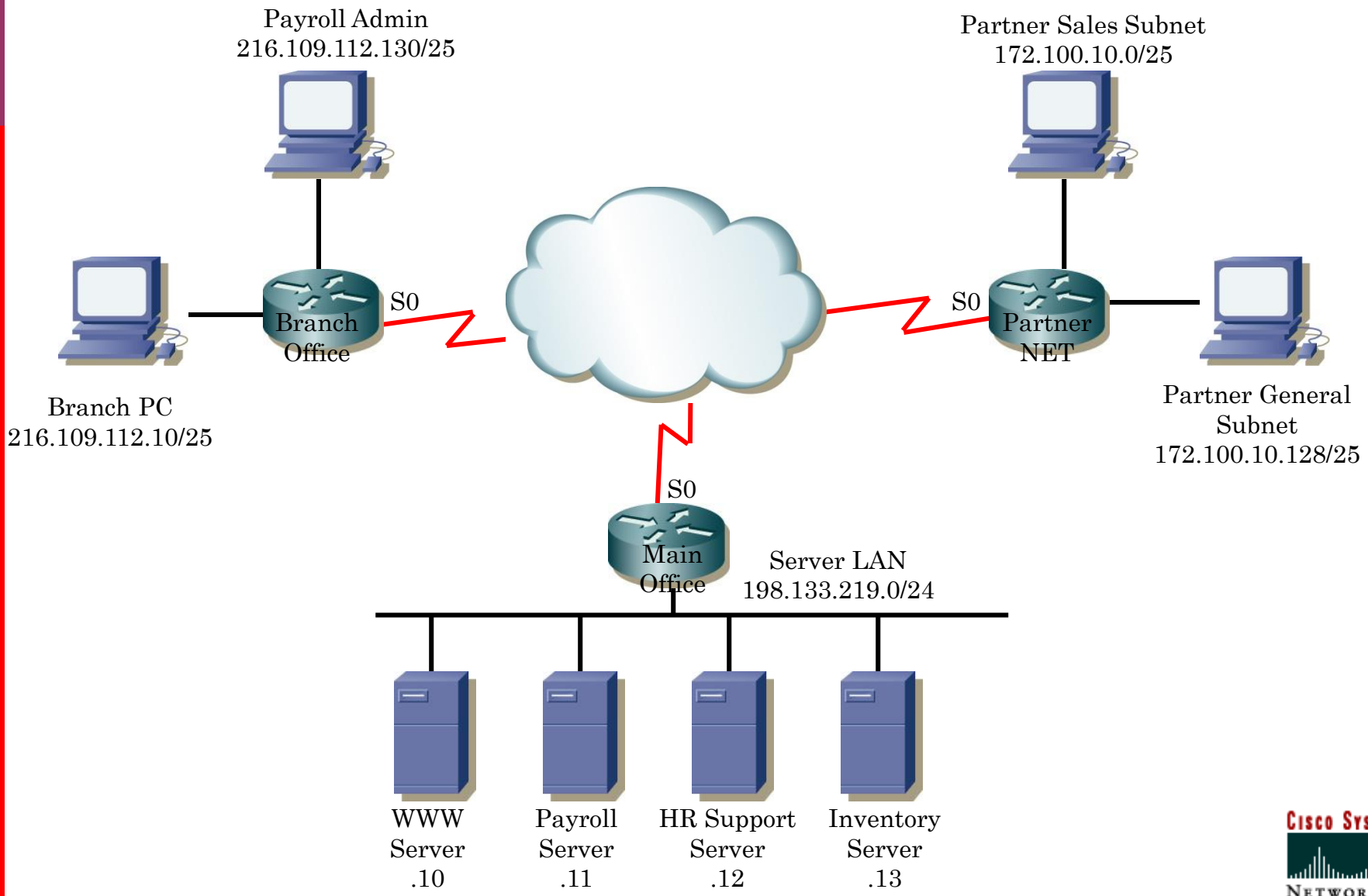
- Cisco IOS Software Release 12.2(14)S and higher.
- To determine if your image supports sequence numbering, enter global configuration mode and type "?" to see if the `resequence` command is listed.

```
Router(config)# ip access-list ?
  extended     Extended Access List
  log-update   Control access list log
  updates
  logging      Control access list logging
  resequence   Resequence Access List
  standard     Standard Access List
```

An Example

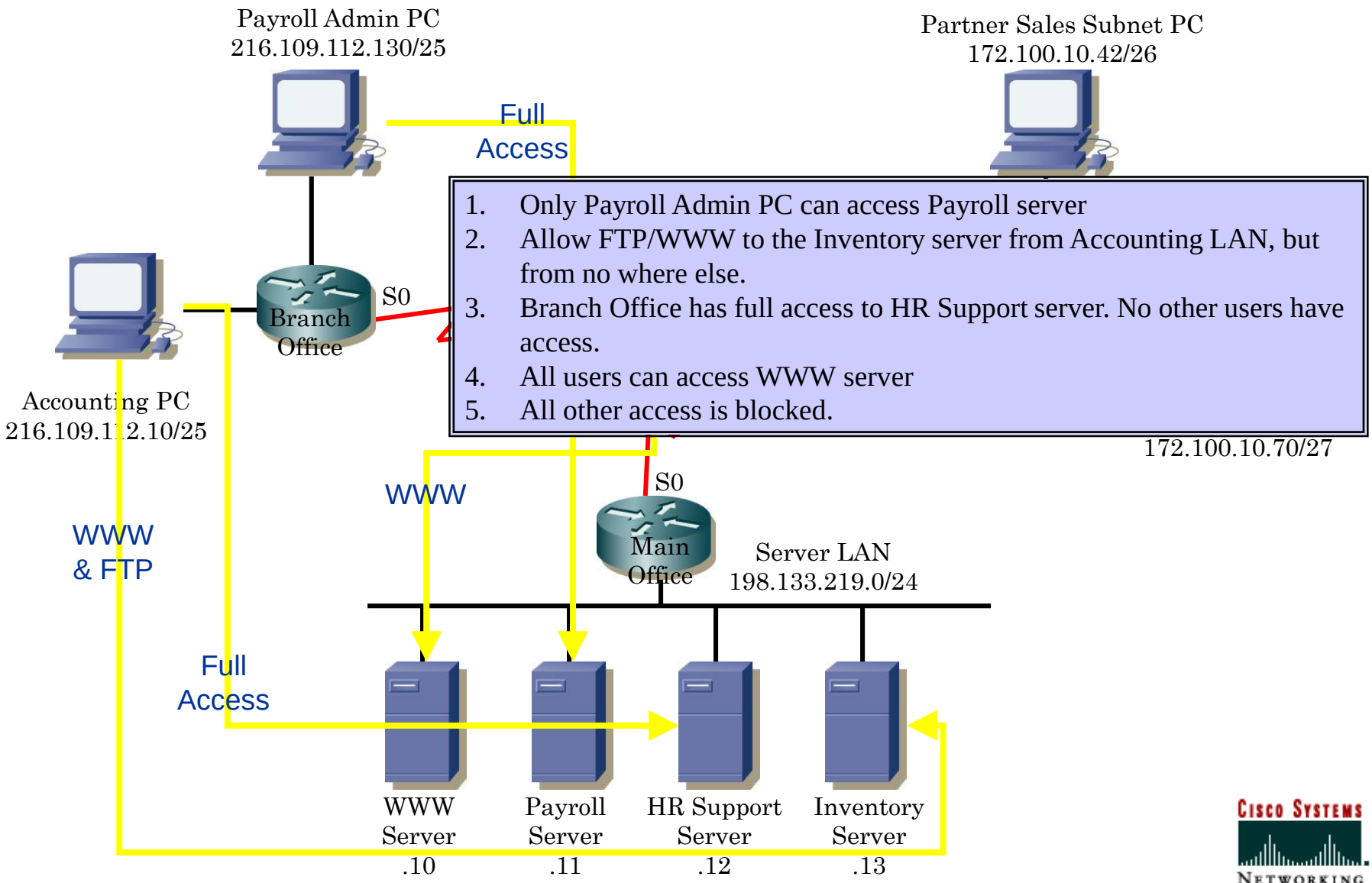


Topology



Use an extended access list named WANRULES
Sequence starting at 1 and incrementing in 1

Business Rules



Plan the ACL statements

- Only Payroll Admin PC can access Payroll server.

```
1 permit ip host 216.109.112.130 host 198.133.219.11
```

- Allow Web/FTP to the Inventory server from Accounting LAN, but from no where else.

```
2 permit tcp 216.109.112.0 0.0.0.127 host
  198.133.219.13 eq 21
3 permit tcp 216.109.112.0 0.0.0.127 host
  198.133.219.13 eq www
```

- Branch Office has full access to HR Support server. No other users have access.

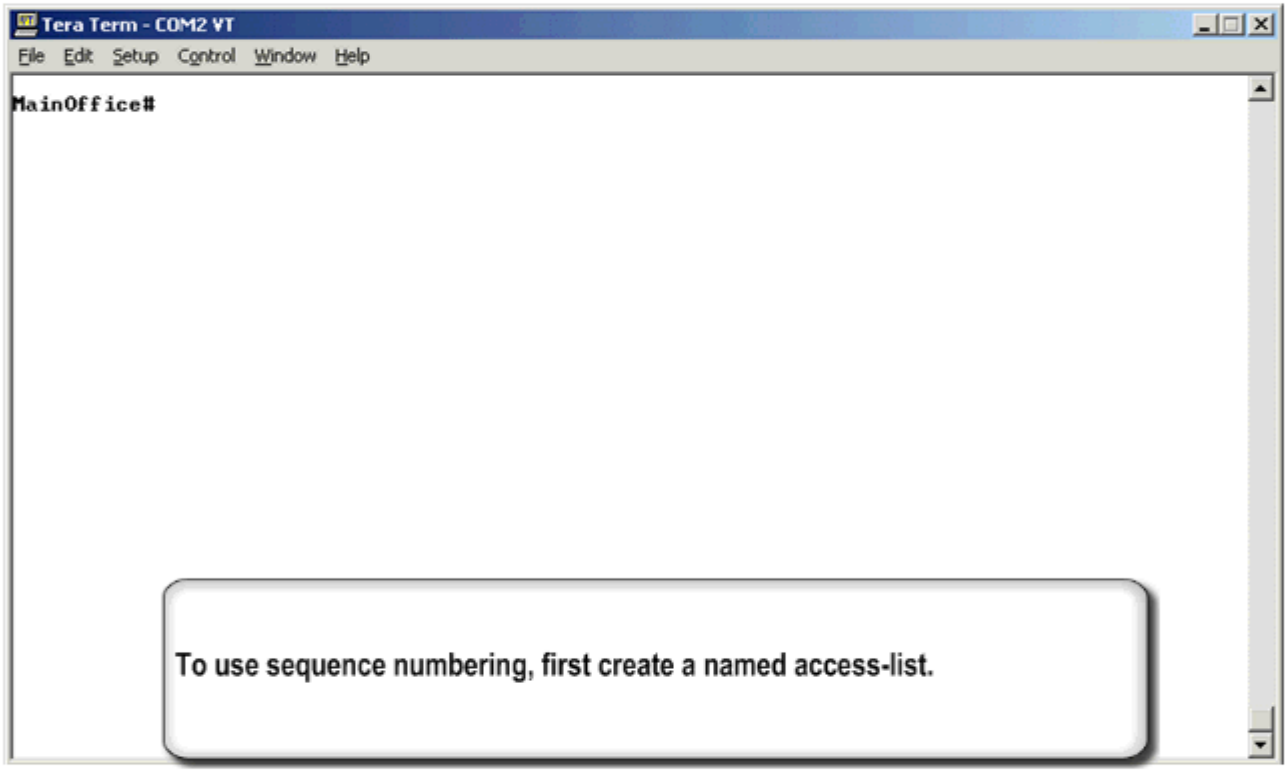
```
4 permit ip 216.109.112.0 0.0.0.255 host
198.133.219.12
```

- All users can access WWW server

```
5 permit tcp any host 198.133.219.10 eq 80
```

- All other access is blocked: Implicit `deny any any`

Configure named ACL



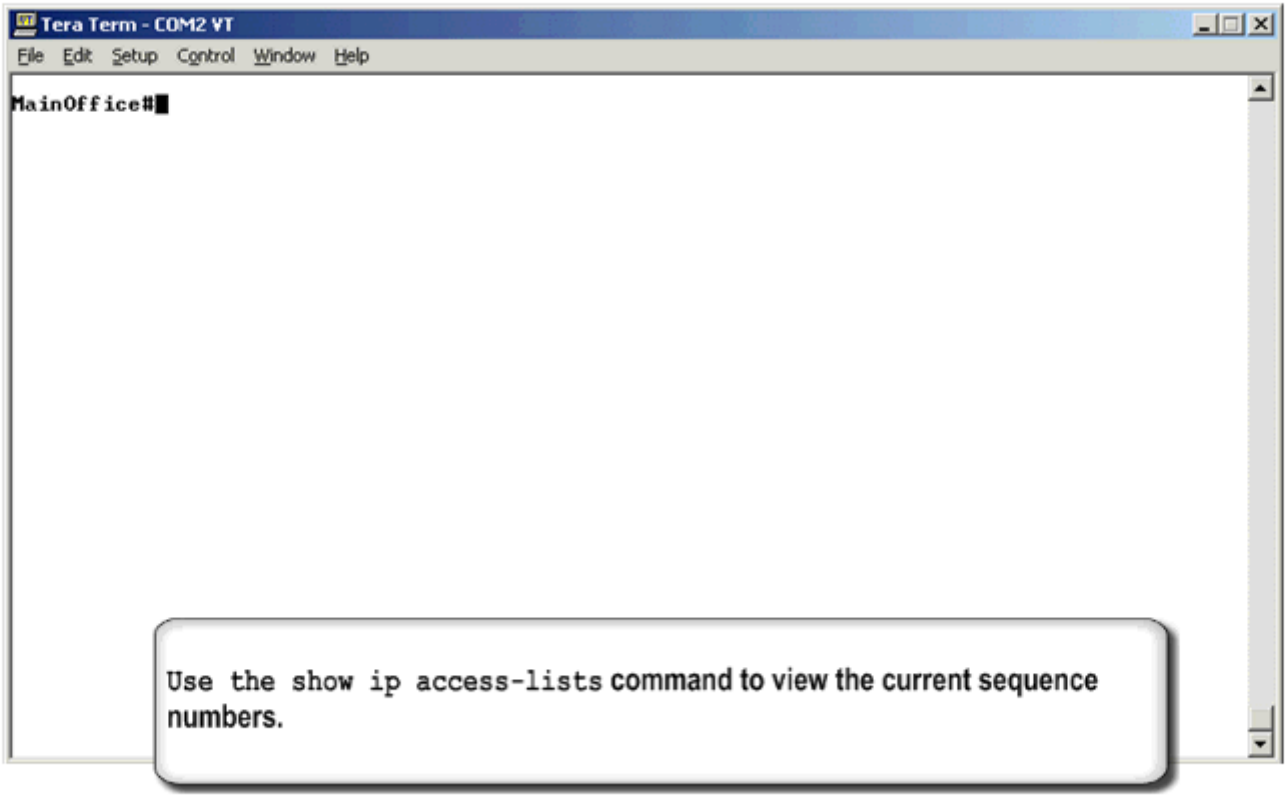
New Business Rule

- After configuring, applying and testing the ACL, management decides to allow the Partner Sales subnet at the PartnerNET site to access the Inventory server for FTP and WWW services. I want these statements to be the 5th & 6th statements in the list.
- This will require editing the ACL to add additional permit statements.
- Prior to sequence numbering, the only way to add lines to the middle of a configured ACL was to remove the ACL and reconfigure it.
- Thanks to sequence numbering, adding lines to the ACL will be easy.

Prepare to edit: Resequence

- If necessary, we will use the `resequence` command to make room for the new statements.
- Remember, to view the current sequence numbers, use the `show ip access-lists` command from privileged mode.
- To resequence, use the `ip access-list resequence` *`access-list-name starting-sequence-number increment`* command from global config mode.

Resequence


Tera Term - COM2 VT
File Edit Setup Control Window Help
MainOffice#
Use the show ip access-lists command to view the current sequence numbers.

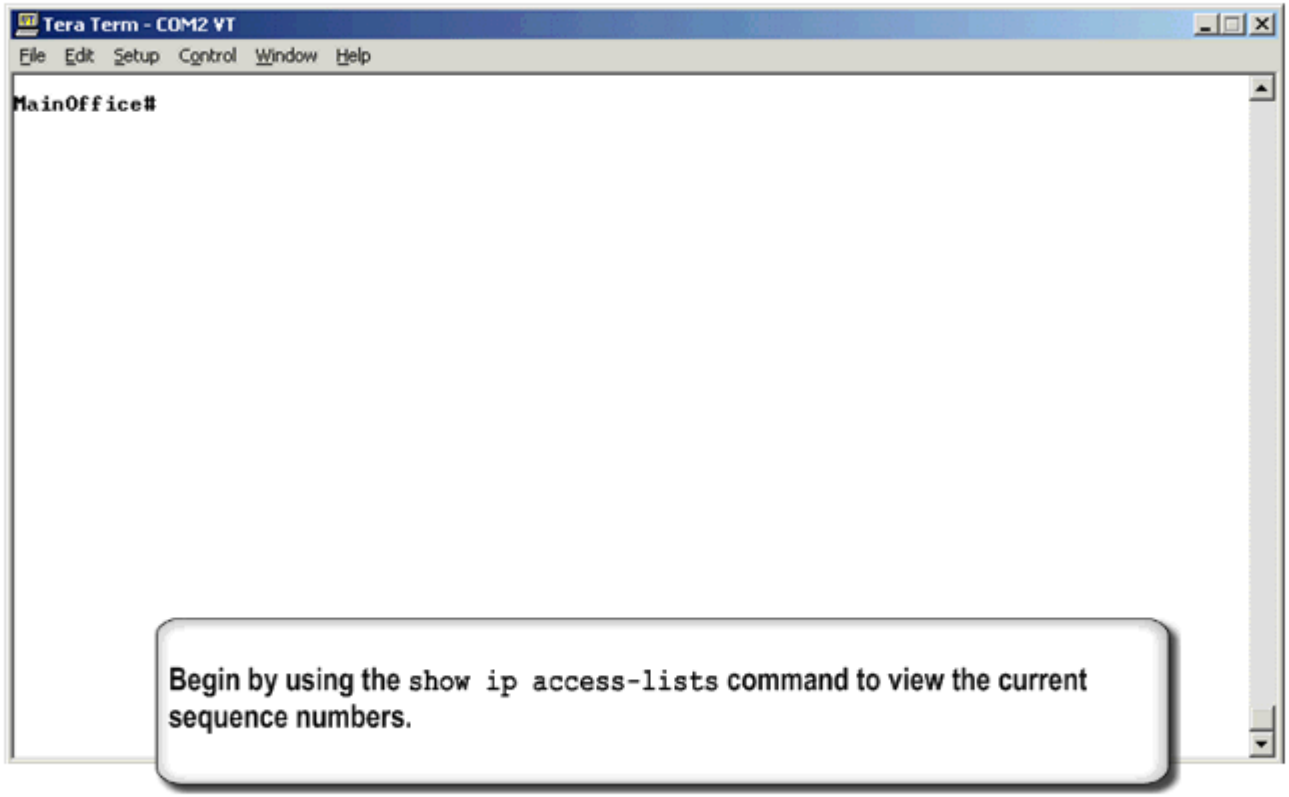
Prepare new statements

- Before editing the ACL, be sure to work out the statements and where you want them placed.
- In our example we want to add these statements:

```
21 permit tcp 172.100.10.0 0.0.0.127 host
  198.133.219.13 eq 21
22 permit tcp 172.100.10.0 0.0.0.127 host
  198.133.219.13 eq 80
```

- These 2 statements will allow the hosts in the sales subnet to access the Inventory server for FTP and WWW services.
- We want to add these statements right after the permit statement that allows the same access from the Branch Office LAN.

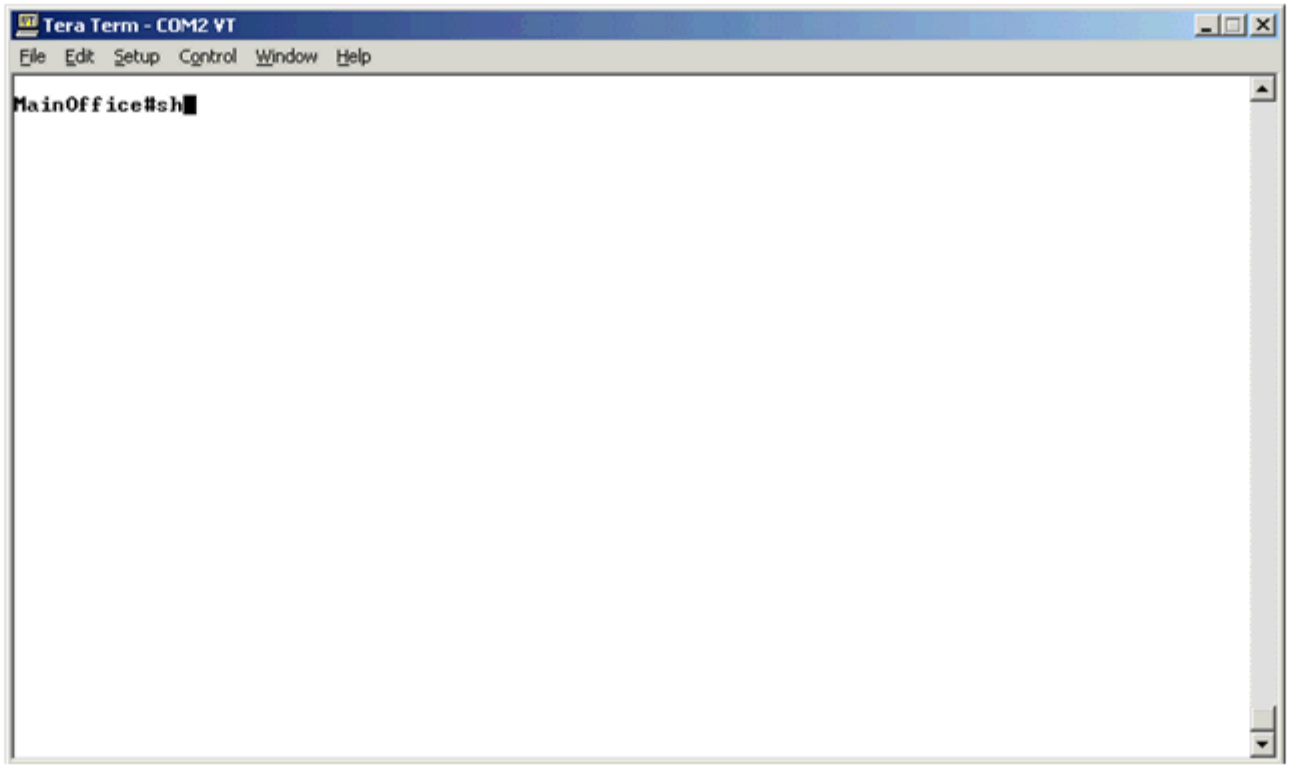
Insert Statements



Remove a line

- After careful consideration, management determines that FTP access to the Inventory server is not required by the PartnerNET Sales hosts.
- You are directed to remove FTP access. WWW access to the Inventory server should remain.
- Use the `show ip access-lists` command to determine which line number to remove.
- From named access list configuration mode use the `no` *`sequence-number`* command to remove the identified line.

Remove ACL sequence numbered line




CISCO SYSTEMS