

**CCNA**

640-802

**Security**







Revision no.: PPT/2K804/04

Revision no.: PPT/2K804/04



# Mitigating Security Threats

**Cisco’s IOS Firewall**



* We can mitigate some of the more common security threats using Cisco IOS Firewall features:

**Stateful IOS Firewall inspection engine**

* This is your perimeter protection feature because it gives your internal users secure access control on a per-application basis. People often call it Context-Based Access Control (CBAC).

**Intrusion detection**

* A deep packet inspection tool that lets you monitor, intercept, and
* respond to abuse in real time by referencing 102 of the most common attack and intrusion detection signatures





# Firewall voice traversal



* An application-level feature based on the protocol’s understanding of call flow as well as the relevant open channels.
* It supports both the H.323v2 and Session Initiation Protocol (SIP) voice protocols.





# ICMP inspection



* Basically permits responses to ICMP packets like ping and traceroute that come from inside your firewall while denying other ICMP traffic.

### Authentication proxy

* A feature that makes users authenticate any time they want to access the network’s resources through HTTP, HTTPS, FTP, and Telnet.
* It keeps personal network access profiles for users and automatically gets them for you from a RADIUS or TACACS+ server and applies them as well.





### Destination URL policy management

* A buffet of features that’s commonly referred to as URL Filtering.

### Per-user firewalls

* These are basically personalized, user-specific, downloadable firewalls obtained through service providers. You can also get personalized ACLs and other settings via AAA server profile storage.

### Cisco IOS router and firewall provisioning

* Allows for no-touch router provisioning, version updates, and security policies.





### Denial of service (DoS) detection and prevention

* A feature that checks packet headers and drops any packets it finds suspicious.

### Dynamic port mapping

* A sort of adapter that permits applications supported by firewalls on nonstandard ports.

### Java applet blocking

* Protects you from any strange, unrecognized Java applets.





### Basic and Advanced Traffic Filtering

* You can use standard, extended, even dynamic ACLs like Lock- and-Key traffic filtering with Cisco’s IOS Firewall.
* And you get to apply access controls to any network segment you want.
* You can specify the exact kind of traffic you want to allow to pass through any segment.

### Policy-based, multi-interface support

* Allows you to control user access by IP address and interface

depending on your security policy.





### Network Address Translation (NAT)

* Conceals the internal network from the outside, increasing security.

### Time-based access lists

* Determine security policies based upon the exact time of day and the particular day of the week.

### Peer router authentication

* Guarantees that routers are getting dependable routing information from actual, trusted sources. (For this to work, you need a routing protocol that supports authentication, like RIPv2, EIGRP, or OSPF.)





# Introduction to Access Lists



## An Access List is essentially List of Conditions that are Categorize Packets

* Types Of Access lists
  + Standard IP Access Lists
    - Use source address for applying rules
  + Extended Access Lists
    - Use both source and Destination address and allow for filtering by protocol type

## Names - After IOS 11.1, you can use access list NAMES

* + Name Access Lists are either Standard or Extended and not Actually a New Type





**Introduction to Access Lists (contd.)**



## To use an Access list as a Packet Filter You Need to Apply it to an Interface on the Router Where You Want the Traffic Filtered

* + Inbound Access List
  + Outbound Access List

## You can Assign only one access list per interface per protocol per direction

* Organize your access lists so that the more specific tests are at the top of the access
* Any time a new entry is added to the access list, it will placed at the bottom of the list
* You cannot remove one line from an access list



Revision no.: PPT/2K804/04



**Introduction to Access Lists (contd.)**

## Unless your access list ends with a permit any command, all packets will be discarded



* Create access lists and then apply them to an interface. Any access list applied to an interface without an access list present will not filter traffic.
* Access lists are designed to filter traffic going through the router. They will not filter traffic that has originated from the router
* Place IP Standard access lists as close to the destination as possible
* Place IP Extended access lists as close to the source as possible



Revision no.: PPT/2K804/04



# Standard IP Access Lists

## Standard IP Access Lists filter network traffic by Examining the Source IP Address in a Packet.



* It Uses access-list numbers 1-99 or 1300-1999

Lab\_A(config)#access-list 10 ?

deny Specify packets to reject permit Specify packets to forward

Lab\_A(config)#access-list 10 permit ?

Hostname or A.B.C.D Address to match any Any source host

host A single host address Lab\_A(config)#access-list 10 permit host 172.16.30.2



Revision no.: PPT/2K804/04



# Wildcard Masking

## Wildcard are used with access lists to specify an individual host, a network, or a certain range of a network or networks.



* Block size is used to specify a range addresses, for e.g.. 64, 32, 16, 8, and 4
* To specify a range of addresses, choose the next-largest block size for e.g. if you need 33 networks you need a block size of 64
* To specify a host, the address would look like this :
  + 172.16.30.5 0.0.0.0
  + The four zero represents each octet, it means that octet in the address must match exactly.



Revision no.: PPT/2K804/04



**Wildcard Masking (contd.)**

## To specify the octet can be any value, the value of 255 is used as shown in following example here’s how a /24 subnet is specified with a wildcard



* + 172.16.30.0 0.0.0.255

## If you want to specify only a small range of subnets specify the range value in a block size.

* For e.g. you want to block access to part of network that is in the range from 172.16.8.0 to 172.16.15.0 .
* Your wildcard would be 0.0.7.255 7.255 is used to determine the block size, which tell router to start at

172.16.8.0 and go up a block size of eight address to network 172.6.15.0



Revision no.: PPT/2K804/04



**Wildcard Masking (contd.)**

## The following e.g. tells the router to match the first three octets exactly but fourth octet can be anything:



Lab\_A(config)#access-list 10 deny 172.16.10.0 0.0.0.255

## The following e.g. tells the router to match the first two octet exactly but last two octets can be anything:

Lab\_A(config)#access-list 10 deny 172.16.10.0 0.0.255.255

## The following e.g. tells the router to start at network 172.16.16.0 and use block size of 4. The range would be then 172.16.16.0 through 172.16.19.0

Lab\_A(config)#access-list 10 deny 172.16.16.0 0.0.3.255



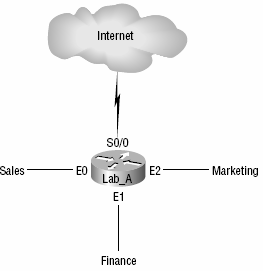
Revision no.: PPT/2K804/04



# Standard Access List Example

Lab\_A#config t



Lab\_A(config)#access-list 10 deny 172.16.40.0

0.0.0.255

Lab\_A(config)#access-list 10 permit any

* The any command is the same thing as saying the following using wildcard

Lab\_A(config)#access-list 10 permit 0.0.0.0

255.255.255.255

Lab\_A(config)#int e1

Lab\_A(config-if)#ip access-group 10 out

* This completely stops traffic from 172.16.4.0 from getting out Ethernet 1



Revision no.: PPT/2K804/04



# Controlling VTY (Telnet) Access

## To perform this function, follow these steps:



* + Create a standard Ip access list that permits only the host or hosts you want to be able to telnet into the router
  + Apply access list to the VTY line with the access-class command
  + Following e.g. only host 172.16.10.3 to telnet into a router

Lab(config )#access-list 50 permit 17.16.10.3

Lab(config )#line vty 0 4 Lab(config )#access-class 50 in



Revision no.: PPT/2K804/04



# Extended Access Lists

## Extended access list filters network traffic by specifying source and destination address as well as the protocol and port number that identify the upper-layer protocol or application



* Example of Extended Access List

|  |  |  |  |
| --- | --- | --- | --- |
| Lab\_A(config)#access-list | 110 | deny tcp any host 172.16.30.5 | eq 21 |
| Lab\_A(config)#access-list | 110 | deny tcp any host 172.16.30.5 | eq 23 |
| Lab\_A(config)#access-list | 110 | permit ip any any |  |

Lab\_A(config)#ip access-group 110 out





# Named Access List



* Named access list are just another way to create standard and extended access list

Lab\_A(config)#ip access-list standard blocksales Lab\_A(config-std-nacl)#

Lab\_A(config-std-nacl)#deny 172.16.40.0 0.0.0.255 Lab\_A(config-std-nacl)#permit any

Lab\_A(config-std-nacl)#exit Lab\_A(config)#^z Lab\_A#show running-config

…

!

ip access-list standard blocksales deny 172.16.40.0 0.0.0.255

permit any

!

…

Lab\_A(config)#ip access-group blocksales out Lab\_A(config)#^z

Lab\_A#





# Advanced Access Lists



### Switch Port ACLs:

* You can only apply port ACLs to layer 2 interfaces on your switches.
* You can apply them as only inbound lists on your interfaces, and you can use only named lists.
* It uses MAC extended access lists that use source and destination MAC addresses and optional protocol type information.





* Switches scrutinize all inbound ACLs applied to a certain interface and decide to allow traffic through depending on whether the traffic is a good match to the ACL or not.
* ACLs can also be used to control traffic on VLANs.
* To make this happen, you just need to apply a port ACL to a trunk port.





Let’s check out the access list:

S1#config t

S1(config)#mac access-list ? extended Extended Access List S1(config)#mac access-list extended ? WORD access-list name

S1(config)#mac access-list extended Todd\_MAC\_List S1(config-ext-macl)#deny ?

H.H.H 48-bit source MAC address any any source MAC address

host A single source host S1(config-ext-macl)#deny any ?

H.H.H 48-bit destination MAC address any any destination MAC address

host A single destination host





S1(config-ext-macl)#deny any host ?

H.H.H 48-bit destination MAC address S1(config-ext-macl)#deny any host 000d.29bd.4b85 S1(config-ext-macl)#permit ?

H.H.H 48-bit source MAC address any any source MAC address

host A single source host S1(config-ext-macl)#permit any any S1(config-ext-macl)#do show access-list

Extended MAC access list Todd\_MAC\_List

deny any host 000d.29bd.4b85

permit any any S1(config-ext-macl)#

Here is how you would apply the list to a switch port: S1(config-ext-macl)#int f0/6

S1(config-if)#mac access-group Todd\_MAC\_List in





**Lock and Key (Dynamic ACLs)**

* This flavor of ACL depends on either remote or local Telnet authentication in combination with extended ACLs.
* Before you can configure a dynamic ACL, you need to apply an extended ACL on your router to stop the flow of traffic through it.
* The only way anyone can get through the blockade is if they telnet to the router and gain authentication.

**It works like this:**

* The Telnet connection the user initiated gets dropped and is replaced with a single-entry dynamic ACL that’s appended to the extended ACL already in place. This causes traffic to be allowed through for a specific amount of time.



### Reflexive ACLs

* These ACLs filter IP packets depending upon upper-layer session information, and they often permit outbound traffic to pass but place limitations on inbound traffic.
* They are only defined with extended named IP ACL.





### Time-Based ACLs

* Time-based ACLs work a lot like extended ACLs do, but their type of access control is totally time oriented.
* Basically, you specify a certain time of day and week and then identify that particular period by giving it a name referenced by a task.
* So, by necessity, the reference function will fall under whatever time constraints you’ve dictated.
* The time period is based upon the router’s clock, but highly recommend using it in conjunction with Network Time Protocol (NTP) synchronization





Here’s an example:

* Corp#config t
* Corp(config)#time-range no-http
* Corp(config-time-range)#periodic we?
* Wednesday weekdays weekend
* Corp(config-time-range)#periodic weekend ?
* hh:mm Starting time
* Corp(config-time-range)#periodic weekend 06:00 to 12:00
* Corp(config-time-range)#exit
* Corp(config)#time-range tcp-yes
* Corp(config-time-range)#periodic weekend 06:00 to 12:00
* Corp(config-time-range)#exit





* Corp(config)#ip access-list extended Time
* Corp(config-ext-nacl)#deny tcp any any eq www time-range no-http
* Corp(config-ext-nacl)#permit tcp any any time-range tcp-yes
* Corp(config-ext-nacl)#interface f0/0
* Corp(config-if)#ip access-group Time in
* Corp(config-if)#do show time-range
* time-range entry: no-http (inactive)
* periodic weekdays 8:00 to 15:00
* used in: IP ACL entry
* time-range entry: tcp-yes (inactive)
* periodic weekend 8:00 to 13:00
* used in: IP ACL entry
* Corp(config-if)#





### Remarks

* This is the tool you grab to use the remark keyword, and it’s really important because it arms you with the ability to include comments, or rather remarks, regarding the entries you’ve made in both your IP standard and extended ACLs.
* Remarks are very cool because they efficiently increase your ability to examine and understand your ACLs to the super-hero level.





Even though you have the option of placing your remarks either before or after a permit or deny statement. Let’s take a look at an example of how to use the remark command:

R2#config t

R2(config)#access-list 110 remark Permit Bob from Sales Only To Finance R2(config)#access-list 110 permit ip host 172.16.10.1 172.16.20.0 0.0.0.255

R2(config)#access-list 110 deny ip 172.16.10.0 0.0.0.255

172.16.20.0 0.0.0.255

R2(config)#ip access-list extended No\_Telnet

R2(config-ext -nacl)#remark Deny all of Sales from Telnetting to Marketing

R2(config-ext -nacl)#deny tcp 172.16.30.0 0.0.0.255

172.16.40.0 0.0.0.255 eq 23

R2(config-ext -nacl)#permit ip any any R2(config-ext -nacl)#do show run [output cut]

!

ip access-list extended No\_Telnet

remark Stop all of Sales from Telnetting to Marketing

deny tcp 172.16.30.0 0.0.0.255 172.16.40.0 0.0.0.255 eq telnet

permit ip any any





**Context-Based Access Control (Cisco IOS Firewall)**

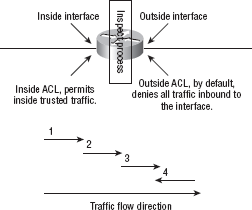
.

* You’ve got to have the Cisco IOS Firewall set in the IOS to make use of Context-Based Access Control (CBAC).
* The CBAC’s job is to crutinize any and all traffic that’s attempting to come through the firewall so it can find out about and control the state information for TCP and UDP sessions.
* And it uses that very information it’s gathered to determine whether to create a temporary pathway into the firewall’s access lists.
* To make this happen, just configure ip inspect lists in the same direction the traffic is flowing.
* If you don’t do this, any return traffic won’t be able to get back through, which will negatively impact any session connections originating from inside the internal network in a big way.





* Take a look at Figure which illustrates in a very simple way how the Cisco IOS Firewall (CBAC) works.
* Cisco IOS Firewall (CBACe) example





A router that’s configured with the Cisco IOS Firewall will process traffic in the following manner:

1. First, if the inside ACL approves, the router will get all inside packets sent to it.
2. Next, the approved traffic is subjected to the firewall’s ip inspect process, which adds the approved connection’s state information into the state table.
3. Finally, the traffic passes through the IP inspect process, which then creates a dynamic ACL entry and puts it into the outside ACL so that the return traffic will be allowed to pass back through the router.





### Authentication Proxy

* You must have this set on all routers, but to be able to do that you must also have the Cisco IOS Firewall feature set up.
* It authenticates inbound users, outbound users, or both.
* Those who would normally be blocked by an ACL can just bring up a browser to get through the firewall and then authenticate on a TACACS+ or RADIUS server.

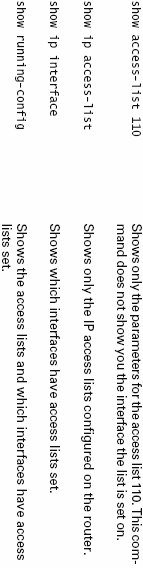
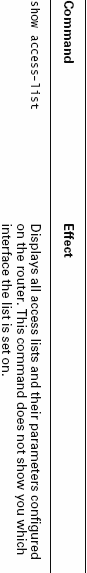
**Note:** You can configure Access Lists by using SDM. You can configure router to work as Firewall by using SDM.







Revision no.: PPT/2K804/04

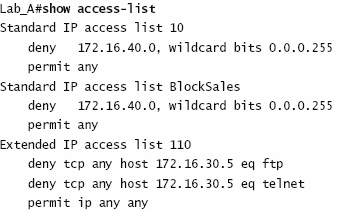


**Monitoring Access Lists**



**Monitoring Access Lists (contd.)**



Lab\_A#show ip interface e1

Ethernet1 is ip, line protocol is up

!

Outgoing access list is BlockSales Inbound access list is not set

!

Lab\_A#



Revision no.: PPT/2K804/04

permission.

prior the

without

emailed or

stored d,

reproduce be

may

material

this of

part No

reserved.

rights All

.

**2004**

**INSTITUTE, CMS**

**©**



**Design & Published by:**