Διαφάνεια 1

Υλοποίηση Δικτυακών Υποδομών και Υπηρεσιών: Χρήση Access - List

Δρ. Απόστολος Γκάμας

Διδάσκων (407/80)

gkamas@uop.gr

Υλοποίηση Δικτυακών Υποδομών και Υπηρεσιών

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ACL Summary

- Access lists perform several functions within a Cisco router, including:
  - Implement security / access procedures
  - Act as a protocol "firewall"
- Extended access lists allow filtering on address, protocol, and applications.
- Access lists are used to limit broadcast traffic.
- Filter the packet flows that flow in or out router interfaces.
- Help protect expanding network resources without impeding the flow of legitimate communication.
- Differentiate packet traffic into categories that permit or deny other features.

ACL Summary

- You can also use access lists to:
  - Identify packets for priority or custom queuing
  - Restrict or reduce the contents of routing updates
- Access lists also process packets for other security features to:
  - Provide IP traffic dynamic access control with enhanced user authentication using the lock-and-key feature
  - Identify packets for encryption
  - Identify Telnet access allowed to the router virtual terminals

What are Access Lists?

- Statements that specify conditions that an administrator sets so the router will handle the traffic covered by the access list in an out-of-the ordinary manner.
- Give added control for processing the specific packets in a unique way.
- Two main types of access lists are:
  - Standard
  - Extended

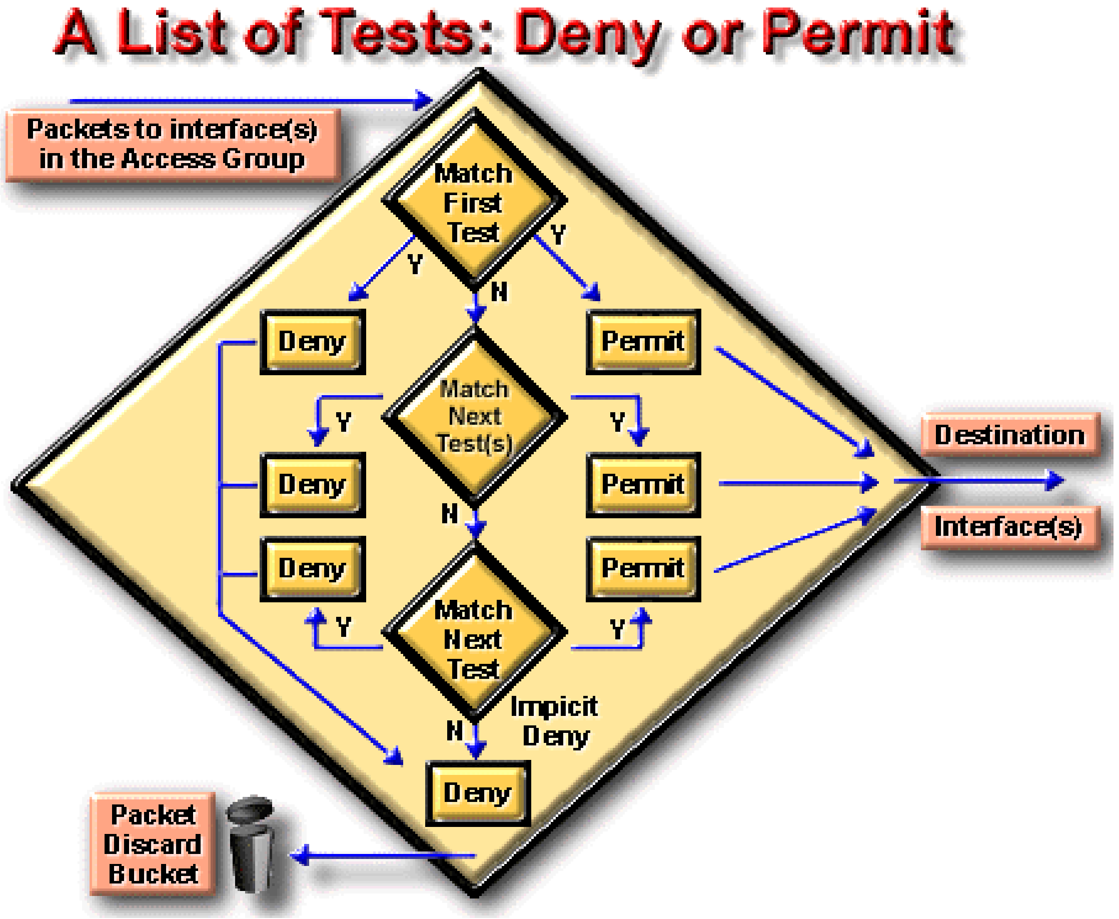
Standard Access Lists

- Standard access lists for IP check the source address of packets that could be routed.
- The result permits or denies output for an entire protocol suite, based on the network/subnet/host address.
- If the packets are denied by the standard access list, all these packets for the given category are dropped.

Extended Access Lists

- Check for both source and destination packet addresses.
- Also can check for specific protocols, port numbers, and other parameters.
- Also permits or denies with more granularity.
- Check for specific protocols, port numbers, and other parameters.
- This allows administrators more flexibility to describe what checking the access list will do. Packets can be permitted or denied output based on where the packet originated and on its destination.
- For example, it can allow electronic mail traffic from E0 to specific S0 destinations, while denying remote logins or file transfers

A list of tests: Deny or Permit



A List of Tests: Deny or Permit

- Access list statements operate in sequential, logical order.
- Evaluate packets from the top down.
- If a packet header and access list statement match, the packet skips the rest of the statements.
- If a condition match is true, the packet is permitted or denied. There can be only one access list per protocol per interface.

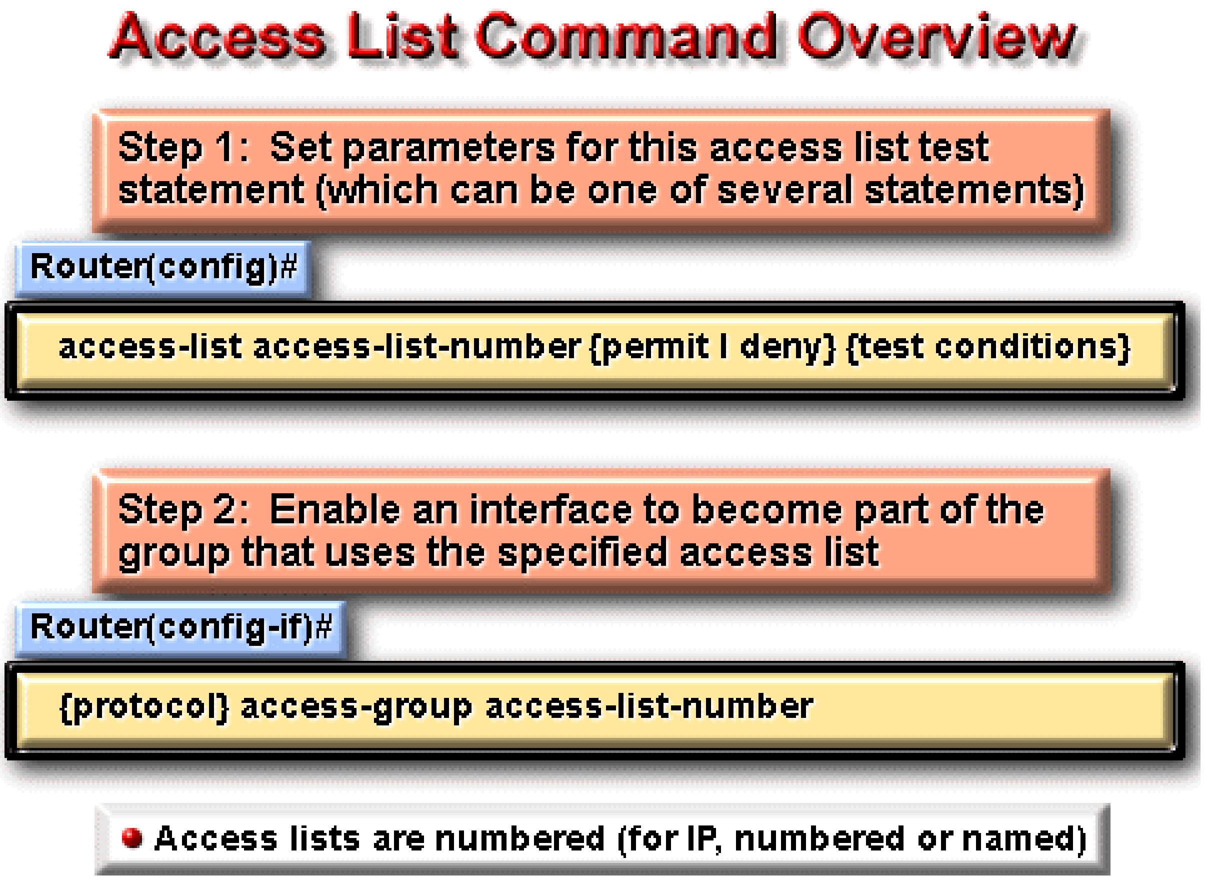
Deny Any Statement

- For logical completeness, an access list must have conditions that test true for all packets using the access list.
- A final implied statement (DENY ANY) covers all packets for which conditions did not test true.
- This final test condition matches all other packets. It results in a deny.
- Instead of proceeding in or out an interface, all these remaining packets are dropped.

Access List Command Overview

- In practice, access list commands can be lengthy character strings.
- Access lists can be complicated to enter or interpret.
- However, you can simplify understanding the general access list configuration commands by reducing the commands to two general elements

Access List Command Overview



Access List Command Overview

Step 1: Set parameters for this access list test statement (which can be one of several statements)

```
Router(config)#
access-list access-list-number {permit l deny} {test conditions}
```

Step 2: Enable an interface to become part of the group that uses the specified access list

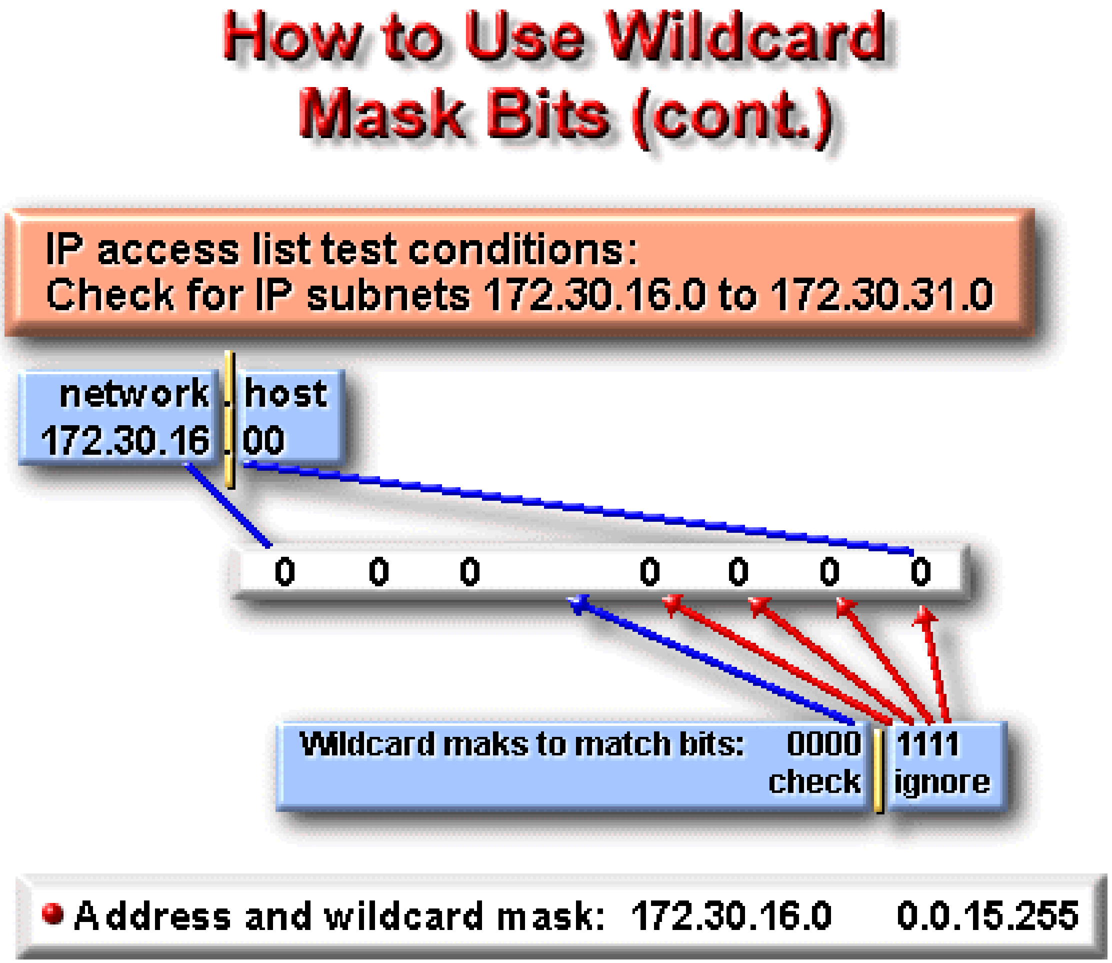
```
Router(config-if)#
{protocol} access-group access-list-number
```

- Access lists are numbered (for IP, numbered or named)

Wildcard Mask Bits

- IP access lists use wildcard masking.
- Wildcard Masking for IP address bits uses the number 1 and the number 0 to identify how to treat the corresponding IP address bits.
  - 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.”

How to use wildcard mask bits



How to Use Wildcard Mask Bits (cont.)

IP access list test conditions:
Check for IP subnets 172.30.16.0 to 172.30.31.0

- Address and wildcard mask: 172.30.16.0 0.0.15.255

Standard ACL (1-99)

Access-list list# {permit/deny} source IP [wildcard mask]

interface [router port]

ip access-group [list#] in|out (out is the default)

- If a match is made, the action defined in this access list statement is performed.
- If no match is made with an entry in the access list, the deny action is performed (implicit deny)
- Should be put close to the destination address because you can not specify the destination address.

Wildcard Mask

- 32 bit long
- Mask bits of 0 imply that the same bit positions must be compared
- Mask bits of 1 imply that the same bit positions are considered to match

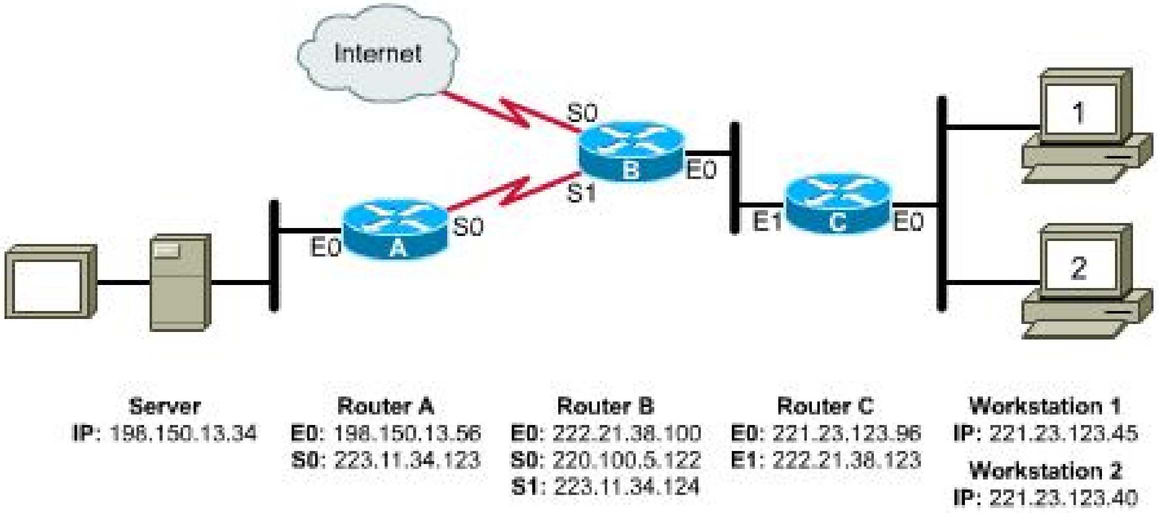
Extended ACL (100-199)

Access-list list# {permit / deny} protocol source [source mask] destination [destination mask] operator [port]

- Should be put close to the source

Correct Placement of Extended ACLs

- Since extended ACLs have destination information, you want to place it as close to the source as possible.
- Place an extended ACL on the first router interface the packet enters and specify inbound in the access-group command.



Server
IP: 198.150.13.34

Router A
E0: 198.150.13.56
S0: 223.11.34.123

Router B
E0: 222.21.38.100
S0: 220.100.5.122
S1: 223.11.34.124

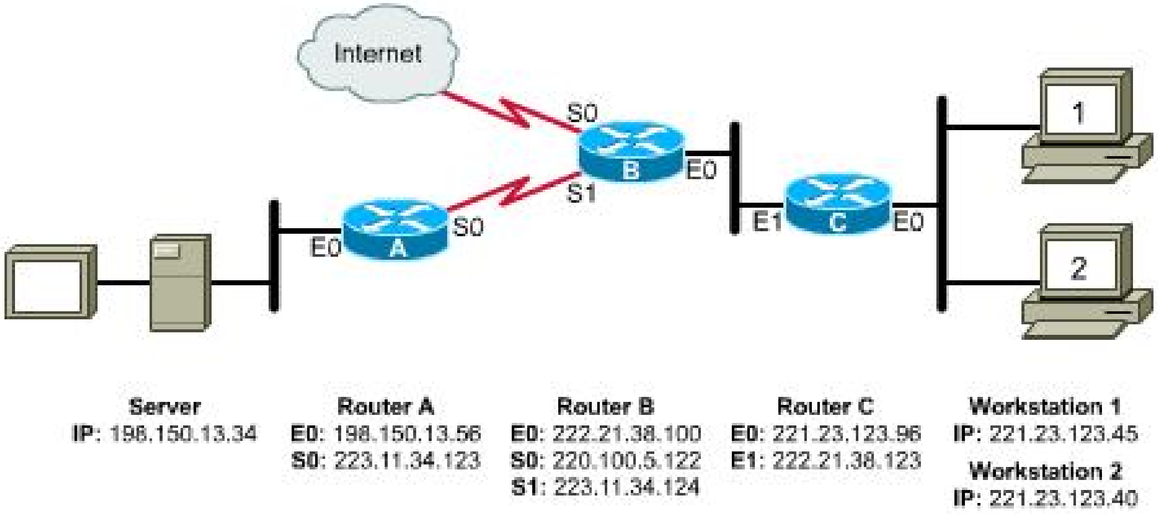
Router C
E0: 221.23.123.96
E1: 222.21.38.123

Workstation 1
IP: 221.23.123.45

Workstation 2
IP: 221.23.123.40

Correct Placement of Extended ACLs

- In the graphic below, we want to deny network 221.23.123.0 from accessing the server 198.150.13.34.
- What router and interface should the access list be applied to?
  - Write the access list on Router C, apply it to the E0, and specify in
  - This will keep the network free of traffic from 221.23.123.0 destined for 198.150.13.34 but still allow 221.23.123.0 access to the Internet



Server
IP: 198.150.13.34

Router A
E0: 198.150.13.56
S0: 223.11.34.123

Router B
E0: 222.21.38.100
S0: 220.100.5.122
S1: 223.11.34.124

Router C
E0: 221.23.123.96
E1: 222.21.38.123

Workstation 1
IP: 221.23.123.45

Workstation 2
IP: 221.23.123.40

Example

- Configure an access list that blocks network 210.93.105.0 from exiting serial port s0 on some router. Allow all other to pass.

```
access-list 4 deny 210.93.105.0 0.0.0.255
access-list 4 permit any
interface s0
ip access-group 4
```

Example (continued)

- Same example but would like to block only the first half IP of the network.

```
access-list 4 deny 210.93.105.0 0.0.0.127
   access-list 4 permit any
        interface s0
        ip access-group 4
```

Example (continued)

- Same example but would like to block only the even numbered IP of the network.

```
access-list 4 deny 210.93.105.0 0.0.0.254
  access-list 4 permit any
     interface s0
     ip access-group 4
```

Example (continued)

- Same example but would like to block only the odd numbered IP of the network.

```
access-list 4 deny 210.93.105.1 0.0.0.254
   access-list 4 permit any
        interface s0
        ip access-group 4
```

Time Savers: the any command

- Since ACLs have an implicit “deny any” statement at the end, you must write statements to permit others through.
- Using our previous example, if the students are denied access and all others are allowed, you would write two statements:
  - **Lab-A(config)#access-list 1 deny 192.5.5.0 0.0.0.127**
  - **Lab-A(config)#access-list 1 permit 0.0.0.0 255.255.255.255**
- Since the last statement is commonly used to override the “deny any,” Cisco gives you an option--the any command:
  - **Lab-A(config)#access-list 1 permit any**

Time Savers: the host command

- Many times, a network administrator will need to write an ACL to permit a particular host (or deny a host). The statement can be written in two ways. Either...

```
Lab-A(config)#access-list 1 permit 192.5.5.10 0.0.0.0
```

- or...

```
Lab-A(config)#access-list 1 permit host 192.5.5.10
```

Ext. ACL Misc

- Port accounting
- access-list 106 permit udp any any
  - eq Match only packets on a given port number
  - fragments Check non-initial fragments
  - gt Match only packets with a greater port number
  - log Log matches against this entry
  - log-input Log matches against this entry, incl. input interface
  - lt Match only packets with a lower port number
  - neq Match only packets not on a given port number
  - precedence Match packets with given precedence value
  - range Match only packets in the range of port numbers
  - tos Match packets with given TOS value

Ext. ACL Misc. cnt.

- TCP header fields
- access-list 106 permit tcp any any

ack	Match on the ACK bit
eq	Match only packets on a given port number
established	Match established connections
fin	Match on the FIN bit
fragments	Check non-initial fragments
gt	Match only packets with a greater port number
log	Log matches against this entry
log-input	Log matches against this entry, incl. input interface
lt	Match only packets with a lower port number
neq	Match only packets not on a given port number
precedence	Match packets with given precedence value
psh	Match on the PSH bit
range	Match only packets in the range of port numbers
rst	Match on the RST bit
syn	Match on the SYN bit
tos	Match packets with given TOS value
urg	Match on the URG bit

Verifying ACLs

- Show commands:
  - show access-lists
    - shows all access-lists configured on the router
  - show 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

Enhanced Access Lists

- Time-Based—Access lists whose statements become active based upon the time of day and/or day of the week.
- Reflexive—Create dynamic openings on the untrusted side of a router based on sessions originating from a trusted side of the router.
- Dynamic (Lock and Key)—Create dynamic entries.
- Context-Based Access Control (CBAC)—Allows for secure handling of multi-channel connections based on upper layer information.

BGP route filtering

- Filtering incoming/outgoing routes
  - Network filter

```
router bgp <AS>
neighbor <ip-address> remote-as <his-AS>
neighbor <ip-address> distribute-list <ACL> [in/out]
!
access-list <ACL> [permit/deny] <network> <mask>
```

BGP route filtering 2

- Filtering with AS-path

```
router bgp <AS>
neighbor <ip-address> remote-as <his-AS>
neighbor <ip-address> filter-list <AS-ACL> [in/out]
!
ip as-path access-list  <AS-ACL> [permit/deny]  <regexp>
```

- regexp examples:

```
^$      - network originated in local AS
.*      - matches anything
_123_  - networks reachable through AS 123
```