

CCNA (200-301) Cisco Certified Network Associate

Prepared by Eng. Taghreed Elsayed

Lecture 9

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

مزيد من التفاصيل

Access Control List (ACL)



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

- ❑ The default behavior of a router is to input a packet on an interface, determine the route, and forward the packet out of the exit interface.
- ❑ His behavior can be changed by implementing ACLs on the router.
- ❑ An ACL is a router configuration script that controls whether a router. permits or denies packets to pass based on criteria found in the packet header.
- ❑ ACLs operate at the network and higher layers.
- ❑ ACLs can be configured on routers to enable the admin to control a user's access to network resources.
- ❑ Can be used to create a simple firewall.

Applying ACL

- In bound Access List: Permit or deny the packets from entering the router before any routing operation.
- Out bound Access List: Permit or deny the packets from exiting the router interface that will done after routing operation.

❑ Types of ACLs :

There are two types of ACLs

❑ Standard ACLs : permit or deny traffic based on **source IP addresses**

❑ Extended ACLs : permit or deny traffic based on:

1. Source information → Source IP address
→ source port number
2. Destination information → Destination IP address
→ Destination port number



3.Protocol type → TCP/UDP/ IP/EIGRP..... .

- ☐ You can configure ACLs on an interface to filter inbound traffic, outbound traffic, or both.
- ☐ You can configure one ACL per protocol, per direction, per interface.

Standard IP ACLs

- Standard ACLs **allow you to permit or deny traffic based solely on the source IP address of each packet.**
- The following ACL will permit access to everyone except users on network 192.168.10.0/24, with the exception of one particular user:

```
access-list 10 permit 192.168.10.1 0.0.0.0
access-list 10 deny 192.168.10.0 0.0.0.255
access-list 10 permit 0.0.0.0 255.255.255.255
```

- ❑ Each line of the ACL is entered at the global config level in the order it is to be applied to the packet.
- ❑ The ID of a Standard IP ACL must be in the range 1-99
- ❑ **Syntax for Standard IP ACLs** ?
 - 1.number - the ID of the ACL in range 1 -99
 - 2.Deny - filter the packet if a match
 - 3.Permit-forward the packet if a match

4.SourceIP - to be compared to the packet source IP address

5.Wildcard - wildcard mask to be used in the comparison

access-list number {deny | permit } source IP [wildcard]

Editing Standard ACLs

- ❖ **Note that the order of the access-lines is important to the meaning of the ACL**
- ❖ **Any new access-line added can only be added to the end of the list**
- ❖ **You are advised to always type ACLs into a text file and paste the ACL into the router .**

- ❖ To remove all access-lines of an ACL with ID n, simply enter:
→ **no access-list n**
- ❖ Then re-paste the modified ACL statements from the text file.

Numbered IP ACLs

- ❖ The ID number of an ACL determines what type it is:
1-99 → **Standard IP ACL**
100-199 → **Extended IP ACL**

Syntax Abbreviations

- A number of abbreviations can be used when writing ACLs, in order to make them more readable.
- **host ip _ address** → replaces ip _ address 0.0.0.0
- **Any** → replace 0.0.0.0 255.255.255.255
- **Omitting a wildcard mask will assume 0.0.0.0**

SO the previous ACL could be written:

```
access-list 10 permit host 192.168.10.1
access-list 10 deny 192.168.10.0 0.0.0.255
access-list 10 permit any
```

A Common Error

- ❑ **Note that every ACL ends with an implicit deny any statement**
- ❑ **To allow everyone except 192.168.10.1 user to access a network, you could write:**

```
access-list 10 deny host 192.168.10.1
```

- **This would be an error. Why?**
- **It should be:**

```
access-list 10 deny host 192.168.10.1  
access-list 10 permit any
```

Note that every ACL must have at least one permit statement

Applying the ACL

- ❑ Having defined the ACL, it must be applied to an interface:

```
int s0/0  
ip access-group 10 in
```

- ❑ Each IP packet entering the s0/0 interface will be permitted or denied by the ACL before the router attempts to route the packet.
- ❑ Alternatively the ACL could be applied to the outward traffic flow:

```
int s0/1  
ip access-group 10 out
```

- ❑ Here, each IP packet routed out of the s0/1 interface will be permitted or denied by the ACL.

Named ACLs

- Named IP ACLs allow you to delete individual entries in a specific ACL.
- You can use sequence numbers to insert statements anywhere in the named ACL.

```
ip access-list standard NO_ACCESS
deny host 192.168.30.128
permit any
!
int fa0/0
ip access-group NO_ACCESS in
```

Verify Access-Lists

```
R1#show access-lists
Standard IP access list 10
    deny 192.168.10.0 0.0.0.255
    permit any
R1#

R1#show ip int fa0/1
FastEthernet0/1 is up, line protocol is up
  Internet address is 192.168.11.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is 10
  Inbound access list is not set
  Proxy ARP is enabled
```



Extended ACLs

- ❑ **access-list number {deny | permit } protocol source IP [source-wildcard] [operator op] [port port-number] destination IP [destination-wildcard] [operator op] [port port-number]**

- Example:

```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 23
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 21
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 20
```

Port Numbers

- `access-list 101 permit tcp any eq ?`

- 20 ftp-data

- 21 ftp

- 23 telnet

- 25 smtp

- 80 www

- 110 pop3

- Operators Eq= equal



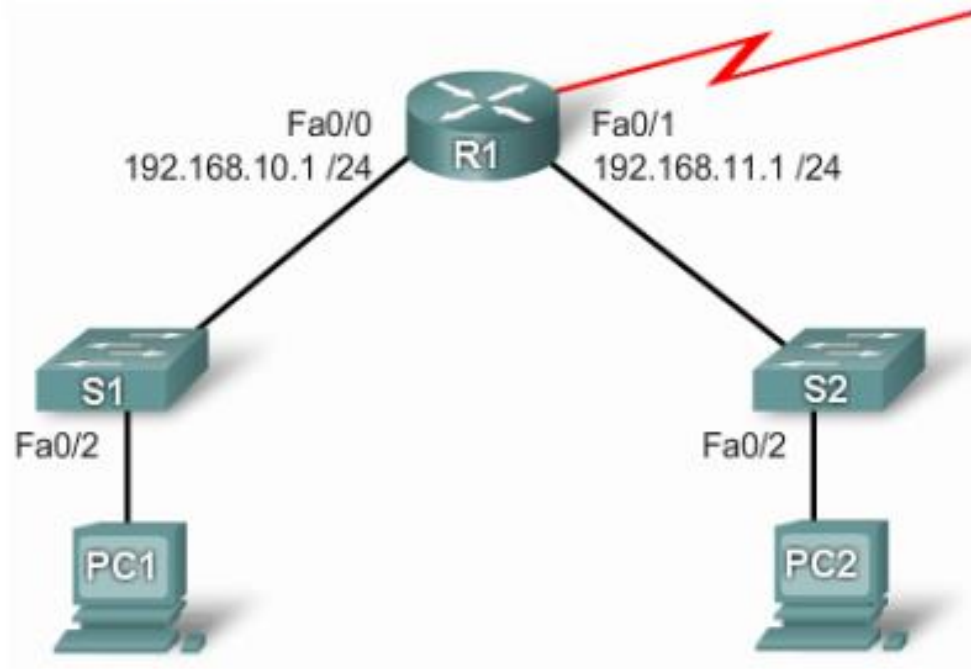
Extended ACL Example

- Allow users to browse both insecure and secure websites

```
access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
!
access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
!
int s0/0
 ip access-group 103 out
 ip access-group 104 in
```

- Port 80 = http:
- Port 443 = https:
- established – allow response on established TCP only

Examples



Example: Deny FTP traffic

- Deny FTP traffic from subnet 192.168.11.0 going to subnet 192.168.10.0, but permitting all other traffic.

```
access-list 101 deny tcp
    192.168.11.0 0.0.0.255
    192.168.10.0 0.0.0.255 eq ftp

access-list 101 deny tcp
    192.168.11.0 0.0.0.255
    192.168.10.0 0.0.0.255 eq ftp-data

access-list 101 permit ip any any

int fa0/1
ip access-group 101 in
```

Example:

- Deny all access to 192.168.10.0 subnet from hosts on 192.168.11.0 subnet, but allow Internet access.

```
access-list 99 deny 192.168.11.0 0.0.0.255
```

```
access-list 99 permit any
```

```
int fa0/0  
ip access-group 99 in
```



Named extended ACLs

```
ip access-list [standard | extended] name
```

```
ip access-group name [in|out]
```

```
ip access-list extended SURFING  
  permit tcp 192.168.10.0 0.0.0.255 any eq 80  
  permit tcp 192.168.10.0 0.0.0.255 any eq 443  
ip access-list extended BROWSING  
  permit tcp any 192.168.10.0 0.0.0.255
```

```
int s0/0  
  ip access-group SURFING out  
  ip access-group BROWSING in
```



Using an ACL to Control VTY Access

- You can control which administrative workstation or network manages your router with an ACL and an access-class statement to your VTY lines.
- You can also use this technique with SSH to further improve administrative access security.
- Only numbered access lists can be applied to VTYS.

```
access-list 21 permit 192.168.10.0 0.0.0.255
!
access-list 21 deny any
line vty 0 4
 login
 password class
 access-class 21 in
```





جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

