Access Control Lists

Accessing the WAN – Chapter 5

Objectives

- Explain how ACLs are used to secure a medium-size Enterprise branch office network.
- Configure standard ACLs in a medium-size Enterprise branch office network.
- Configure extended ACLs in a medium-size Enterprise branch office network.
- Describe complex ACLs in a medium-size Enterprise branch office network.
- Implement, verify and troubleshoot ACLs in an enterprise network environment.

Basic OSPF Configuration

```
hostname R1
!
interface Serial0/0
 ip address 192.168.10.1 255.255.255.252
!
interface Serial0/1
 ip address 192.168.10.5 255.255.255.252
!
router ospf 1
 network 192.168.10.4 0.0.0.3 area 0
```

Review of Wildcard Masks

- When a wildcard mask is used to compare two IP addresses for a match, the wildcard mask determines which bits must be tested and which bits should be ignored.
  - 0 bit means the pair of bits must match
  - 1 bit means the bits are ignored

Wildcard octet 0 (00000000) means all 8 bits must match

Wildcard octet 255 (11111111) means all 8 bits are ignored

Wildcard octet 63 (00111111) only the first two need match

Working Out Wildcard Masks

- Use 0.0.0.0 if both addresses must match exactly
- Use 255.255.255.255 if any two addresses should result in a match
- Use 0.0.0.255 if only addresses on the same class C network should match
- Usually we use Wildcard masks to determine if a particular IP is on a particular subnet. In this case, the Wildcard mask is the inverse of the subnet mask.
- What is the wildcard mask used to test if an IP address is in the subnet 172.16.0.0/20
- Subtract the subnet mask from 255.255.255.255
  - Subnet mask is: 255.255.240.0
  - Wildcard mask is: 0.0.15.255

Access Control Lists

- 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.
- This 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

Standard IP ACLs

- Standard IP 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 or 1300-1999; in this case it is 10.

The Action of the ACL

- The ACL would be applied to a packet line by line as follows:
- The source address of the packet is compared to the address in the access-line using the wildcard mask
- If a match:

  The specified action permit or deny is applied. In this case, the packet is permitted to pass through the router.

  The remaining access-lines of the ACL are ignored
- If no match:

  The procedure is repeated with the next access-line
- If no access-lines match:

  The action deny is applied, i.e. the packet is filtered

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.

Syntax for Standard IP ACLs

```
access-list number {deny|permit|remark}
              sourceIP [wildcard] [log]
```

number	- the ID of the ACL in range 0-99, 1300-1999
deny	- filter the packet if a match
permit	- forward the packet if a match
remark	- documentation line
sourceIP	- to be compared to the packet source IP address
wildcard	- wildcard mask to be used in the comparison
log	- generate logging messages

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	replaces 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
```

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

  Source and/or Destination IP address

  Protocol type

  TCP/UDP source port (optionally)

  TCP/UDP destination port (optionally)

- 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

Numbered IP ACLs

- The ID number of an ACL determines what type it is:
- 1-99, 1300-1999

  Standard IP ACL

- 100-199, 2000-2699

  Extended IP ACL

- Other ranges are used for other protocol ACLs

  e.g. IPX/SPX, Appletalk, etc.

- Only IP ACLs are considered on CCNA

Extended ACLs

```
access-list number {deny | permit | remark}
 protocol sourceIP [source-wildcard]
 [operator op] [port port-number]
 destinationIP [destination-wildcard]
 [operator op] [port port-number]
 [established]
```

- 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

  equal (eq), not equal (neq), greater than (gt), and less than (lt)

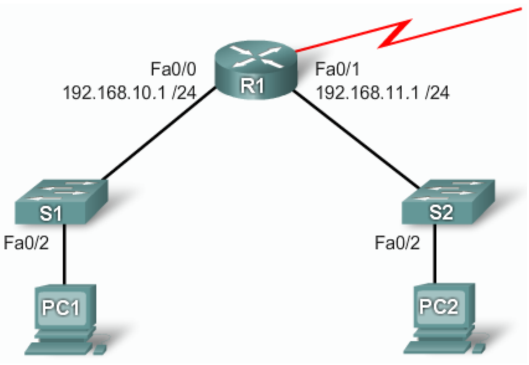
Extended ACL Example

- 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
```

Placing the ACLs

- Standard ACLs
- Because standard ACLs do not specify destination addresses, place them as close to the destination as possible. Otherwise the ACL may be more restrictive than intended

- Extended ACLs
- Locate extended ACLs as close as possible to the source of the traffic denied. This way, undesirable traffic is filtered without crossing the network infrastructure.

```
int s0/0
 ip access-group {number|name} {in|out}
```

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 establishe

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
```

Complex ACLs

- Config. of these ACLs is outside the scope of CCNA
- Dynamic ACLs

  Users are blocked until they use Telnet and are authenticated with the router.

  Dependent on Telnet connectivity, authentication, and extended ACLs.
- Reflexive ACLs

  Provides a truer form of session filtering than an extended ACL that uses the `established` parameter
- Time-based ACLs

  Similar to extended ACLs in function, but they allow for access control based on time

Summary

- An Access List (ACL) is:

  A series of permit and deny statements that are used to filter traffic
- Standard ACL
  - –Identified by numbers 1 - 99 and 1300 - 1999
  - –Filter traffic based on source IP address
- Extended ACL
  - –Identified by number 100 -199 & 2000 - 2699
  - –Filter traffic based on
    - •Source IP and/or Destination IP address
    - •Protocol
    - •Port number

Summary

- Named ACL
  - –Used with IOS 11.2 and above
  - –Can be used for either standard or extended ACL
- ACL’s use Wildcard Masks (WCM)
  - –Described as the inverse of a subnet mask
    - •Reason
      - –0 → check the bit
      - –1 → ignore the bit
- Complex ACL
  - –Dynamic ACL
  - –Reflexive ACL
  - –Time based ACL

Summary

- Implementing ACLs
  - –1st create the ACL
  - –2nd place the ACL on an interface
    - •Standard ACL are placed nearest the destination
    - •Extended ACL are placed nearest the source
- Use the following commands for verifying & troubleshooting an ACL
  - –Show access-list
  - –Show ip interfaces
  - –Show run