Access Control Lists

- An ACL identifies traffic based on characteristics of the packet such as source IP address, destination IP address, port number
- The router or switch can take an action based on the result of the ACL
- ACL's are supported on both routers and switches. I will refer to 'routers' throughout this section

Security

- The original use of ACLs was as a security feature to decide if traffic should be allowed to pass through the router
- By default a router will allow all traffic to pass between its interfaces
- When ACLs are applied the router identifies traffic and then decides if it will be allowed or not

- ACL's are also used in other software policies when traffic has to be identified, for example:
    - Identify traffic to give better service to in a QoS Quality of Service policy
    - Identify traffic to translate to a different IP address in a NAT Network Address Translation policy

- Access Control Lists are made up of Access Control Entries which are a series of permit or deny rules
- Each ACE is written in a separate line

```
R2(config)#
access-list 100 deny tcp 10.10.30.0 0.0.0.255 gt 49151 10.10.20.1 0.0.0.0 eq 23
            No.  Action Protocol  Source: IP  Wildcard  Qual. Port  Destination: IP  Wildcard  Qual. Port
```

R1(config)# access-list 100 deny tcp 10.10.10.10 0.0.0.0 gt 49151 10.10.50.10 0.0.0.0 eq 23
R1(config)# access-list 100 permit tcp 10.10.10.0 0.0.0.255 gt 49151 10.10.50.10 0.0.0.0 eq 23
R1(config)# access-list 100 deny tcp 10.10.20.10 0.0.0.0 gt 49151 10.10.50.10 0.0.0.0 eq 23
R1(config)# access-list 100 permit tcp 10.20.10.0 0.0.0.255 gt 49151 10.10.50.10 0.0.0.0 eq 23

Standard vs Extended ACLs

R1(config)#access-list ?
<1-99> IP standard access list
<100-199> IP extended access list
<1300-1999> IP standard access list (expanded range)
<2000-2699> IP extended access list (expanded range)

- Standard ACLs reference the source address only
- Extended ACLs check based on the protocol, source address, destination address, and port number
- Standard ACL Range: 1 – 99
- Extended ACL Range: 100 - 199

- Cisco expanded the original ACL Ranges
- Standard: 1-99, 1300-1999
- Extended: 100-199, 2000-2699

R1(config)# access-list 1 deny 10.10.10.10 0.0.0.0
R1(config)# access-list 1 permit 10.10.10.0 0.0.0.255

- The default wildcard mask for a Standard ACL is 0.0.0.0, meaning an individual host address.
- *R1(config)# access-list 1 deny 10.10.10.10*

- Do not forget to enter the wildcard when specifying an IP subnet
- *R1(config)# access-list 1 deny 10.10.10.0*

Extended
R1(config)# access-list 100 deny tcp 10.10.10.10 0.0.0.0 gt 49151 10.10.50.10 0.0.0.0 eq 23
R1(config)# access-list 100 permit tcp 10.10.10.0 0.0.0.255 gt 49151 10.10.50.10 0.0.0.0 eq telnet

There is no default wildcard mask for Extended ACLs

Named ACLs

- You can now reference ACLs by number or by a name
- Named ACLs begin with the command ‘ip access-list’ instead of ‘access-list

R1(config)#ip access-list standard Flackbox-Demo
R1(config-std-nacl)#deny 10.10.10.10 0.0.0.0
R1(config-std-nacl)#permit 10.10.10.0 0.0.0.255

ACL Syntax

```
R1(config)#access-list 100 ?
  deny     Specify packets to reject
  permit   Specify packets to forward
  remark   Access list entry comment
! Truncated
```

```
R1(config)#access-list 100 permit ?
  <0-255>  An IP protocol number
  ahp      Authentication Header Protocol
  eigrp    Cisco's EIGRP routing protocol
  esp      Encapsulation Security Payload
  gre      Cisco's GRE tunneling
  icmp     Internet Control Message Protocol
  ip       Any Internet Protocol
  ospf     OSPF routing protocol
  tcp      Transmission Control Protocol
  udp      User Datagram Protocol
! truncated
```

- Use TCP or UDP if you want the ACE to apply to traffic for a particular application between a source and destination address

```
R1(config)#access-list 100 deny tcp 10.10.10.0 0.0.0.255 10.10.50.0
0.0.0.255 eq 80
```

- Use IP if you want the ACE to apply to all traffic between a source and destination address

```
R1(config)#access-list 100 deny ip 10.10.10.0 0.0.0.255 10.10.50.0
0.0.0.255
```

```
R1(config)#access-list 100 permit tcp ?
  A.B.C.D  Source address
  any      Any source host
  host     A single source host
```

- Wildcards save you typing out the wildcard mask
- These examples mean the same thing:

```
R1(config)#access-list 100 permit tcp 10.10.10.10 0.0.0.0
R1(config)#access-list 100 permit tcp host 10.10.10.10

R1(config)#access-list 100 permit tcp 0.0.0.0 255.255.255.255
R1(config)#access-list 100 permit tcp any
```

- Specifying the source port number is optional, it defaults to any port

```
R1(config)#access-list 100 permit tcp 10.10.10.0 0.0.0.255 ?
  A.B.C.D  Destination address
  any      Any destination host
  eq       Match only packets on a given port number
  gt       Match only packets with a greater port number
  host     A single destination host
  lt       Match only packets with a lower port number
  neq      Match only packets not on a given port number
  range    Match only packets in the range of port numbers
```

- The destination address uses the same format as the source address

```
R1(config)#access-list 100 permit tcp host 10.10.10.10 10.10.20.0
0.0.0.255
```

- Additional options are available after entering the destination address such as destination port, TCP flags and logging.

```
R1(config)#access-list 100 permit tcp host 10.10.10.10 10.10.20.0 0.0.0.255 ?
  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
  gt           Match only packets with a greater port number
  log          Log matches against this entry
  log-input    Log matches against this entry, including input interface
  lt           Match only packets with a lower port number
  neq          Match only packets not on a given port number
  range        Match only packets in the range of port numbers
  rst          Match on the RST bit
  syn          Match on the SYN bit
  urg          Match on the URG bit
```

Complete ACE Example

```
R1(config)#access-list 100 deny tcp host 10.10.10.10 10.10.20.0
0.0.0.255 eq www log
```

```
R2#sh access-lists 100
Extended IP access list 100
permit tcp host 10.10.30.10 host 10.10.20.1 eq telnet (13 match(es))
deny tcp 10.10.30.0 0.0.0.255 host 10.10.20.1 eq telnet (4 match(es))
```

- The 'log' keyword is not required to log hit counts. It is used to log to the console or an external monitoring server

ACL Operations/Groups

- ACLs are applied at the interface level with the Access-Group command
- ACLs can be applied in the inbound or outbound direction
- You can have a maximum of one ACL per interface per direction
- You can have both an inbound and an outbound ACL on the same interface, but not 2 inbound or outbound ACLs
- An interface can have no ACL applied, an inbound ACL only, an outbound ACL only, or ACLs in both directions

Access-Group Configuration
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ip access-group 100 out
R1(config-if)# ip access-group 101 in

R3#show ip interface f1/0 | include access list
Outgoing access list is 100
Inbound access list is 101
('not set' if ACL is not applied)

- The ACL is read by the router from top to bottom
- As soon as a rule matches the packet, the permit or deny action is applied and the ACL is not processed any further
- The order of rules is important

This will deny 10.10.10.10 but permit the rest of the 10.10.10.0/24 subnet
R1(config)# access-list 1 deny host 10.10.10.10
R1(config)# access-list 1 permit 10.10.10.0 0.0.0.255
This will permit all of the 10.10.10.0/24 subnet including 10.10.10.10
R1(config)# access-list 1 permit 10.10.10.0 0.0.0.255
R1(config)# access-list 1 deny host 10.10.10.10

ACEs are automatically numbered in increments of 10
R1#sh access-lists 110
Extended IP access list 110
10 deny tcp host 10.10.10.10 host 10.10.50.10 eq telnet
20 permit tcp 10.10.10.0 0.0.0.255 host 10.10.50.10 eq telnet
30 deny tcp host 10.10.20.10 host 10.10.50.10 eq telnet
40 permit tcp 10.20.10.0 0.0.0.255 host 10.10.50.10 eq telnet

Injecting ACEs in an Existing ACL

Support for injecting ACEs in an existing ACL started in Named ACLs but is also supported in Numbered ACLs now

R1(config)#ip access-list extended 110
R1(config-ext-nacl)#15 deny tcp host 10.10.10.11 host 10.10.50.10 eq telnet

R1#sh access-lists 110
Extended IP access list 110
10 deny tcp host 10.10.10.10 host 10.10.50.10 eq telnet
15 deny tcp host 10.10.10.11 host 10.10.50.10 eq telnet
20 permit tcp 10.10.10.0 0.0.0.255 host 10.10.50.10 eq telnet
30 deny tcp host 10.10.20.10 host 10.10.50.10 eq telnet
40 permit tcp 10.20.10.0 0.0.0.255 host 10.10.50.10 eq telnet

Implicit Deny All

- There is an implicit ‘deny any any’ rule at the bottom of ACLs
- If an ACL is not applied to an interface, all traffic is allowed
- If an ACL is applied, all traffic is denied except what is explicitly allowed
- Traffic from 10.10.10.0/24 will be permitted, everything else is denied

R1(config)# access-list 1 permit 10.10.10.0 0.0.0.255

Many organisations include an explicit deny all at the end of ACLs to log illegal traffic

R1(config)# access-list 1 permit 10.10.10.0 0.0.0.255
R1(config)# access-list 1 deny any log

Explicit Permit All

- If an ACL is applied, all traffic is denied except what is explicitly allowed
- If you want to reverse this so that all traffic is permitted except what is explicitly denied, add a permit all statement to the end of the ACL
- Traffic from 10.10.10.0/24 is denied, everything else is permitted

R1(config)# access-list 1 deny 10.10.10.0 0.0.0.255
R1(config)# access-list 1 permit any

Traffic Sourced from Router

- ACL’s applied to an interface do not apply to traffic which originates from the router itself
- The hosts in the 10.1.1.0/24 subnet cannot Telnet to R2
- An administrator can Telnet to R2 from the CLI on R1

R1(config)# access-list 100 deny tcp any any eq 23
R1(config)# interface f1/0
R1(config)# ip access-group 100 out