IPv4 ACLs (Access Control Lists)

Standard ACL Syntax

```
! Legacy syntax
access-list <number> {permit | deny} <source> [log]

! Modern syntax
ip access-list standard {<number> | <name>}
 [<sequence>] {permit | deny} <source> [log]
```

Actions	
permit	Allow matched packets
deny	Deny matched packets
remark	Record a configuration comment
evaluate	Evaluate a reflexive ACL

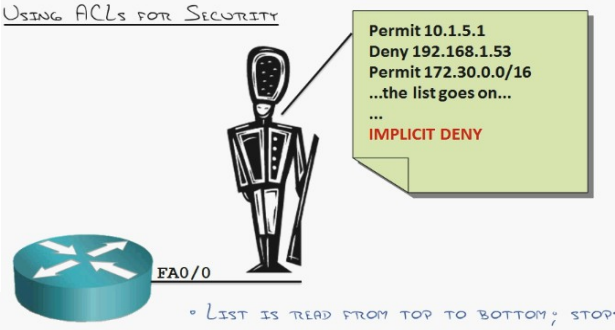
Extended ACL Syntax

```
! Legacy syntax
access-list <number> {permit | deny} <protocol> <source> [<ports>] <destination> [<ports>] [<options>]

! Modern syntax
ip access-list extended {<number> | <name>}
 [<sequence>] {permit | deny} <protocol> <source> [<ports>] <destination> [<ports>] [<options>]
```

- WHAT THEY CAN BE USED FOR:
  - ACCESS CONTROL
  - NAT
  - QUALITY OF SERVICE
  - DEMAND DIAL ROUTING
  - POLICY ROUTING
  - ROUTE FILTERING

USING ACLs FOR SECURITY



STANDARD AND EXTENDED ACCESS LISTS

- STANDARD
  - MATCHES BASED ON SOURCE ADDRESS
  - LOWER PROCESSOR UTILIZATION
  - AFFECT DEPENDS ON APPLICATION
- EXTENDED
  - MATCHES BASED ON SOURCE/DESTINATION ADDRESS, PROTOCOL, SOURCE/DESTINATION PORT NUMBER
  - HIGHER PROCESSOR UTILIZATION
  - SYNTAX TAKES SOME TIME TO LEARN
- REFLEXIVE (ESTABLISHED)
  - ALLOWS RETURN TRAFFIC FOR INTERNAL REQUESTS

- LIST IS READ FROM TOP TO BOTTOM; STOPS AT FIRST MATCH
- INVISIBLE IMPLICIT DENY AT THE BOTTOM
- ACL IS APPLIED TO AN INTERFACE INBOUND OR OUTBOUND
- OTHER POSSIBILITIES: ACL USED FOR NAT, QoS, VPN

Source/Destination Definitions	
any	Any address
host <address>	A single address
<network> <mask>	Any address matched by the wildcard mask

IP Options	
dscp <DSCP>	Match the specified IP DSCP
fragments	Check non-initial fragments
option <option>	Match the specified IP option
precedence {0-7}	Match the specified IP precedence
ttl <count>	Match the specified IP time to live (TTL)

TCP/UDP Port Definitions			
eq <port>	Equal to	**neq <port>**	Not equal to
lt <port>	Less than	**gt <port>**	Greater than
range <port> <port>	Matches a range of port numbers		

Miscellaneous Options	
reflect <name>	Create a reflexive ACL entry
time-range <name>	Enable rule only during the given time range

ACL Numbers	
1-99 **1300-1999**	IP standard
100-199 **2000-2699**	IP extended
200-299	Protocol
300-399	DECnet
400-499	XNS
500-599	Extended XNS
600-699	Appletalk
700-799	Ethernet MAC
800-899	IPX standard
900-999	IPX extended
1000-1099	IPX SAP
1100-1199	MAC extended
1200-1299	IPX summary

TCP Options	
ack	Match ACK flag
fin	Match FIN flag
psh	Match PSH flag
rst	Match RST flag
syn	Match SYN flag
urg	Match URG flag
established	Match packets in an established session

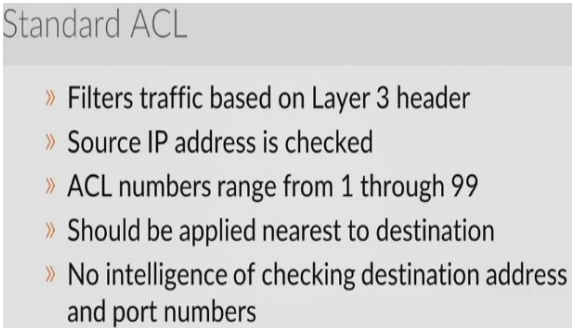
Logging Options	
log	Log ACL entry matches
log-input	Log matches including ingress interface and source MAC address

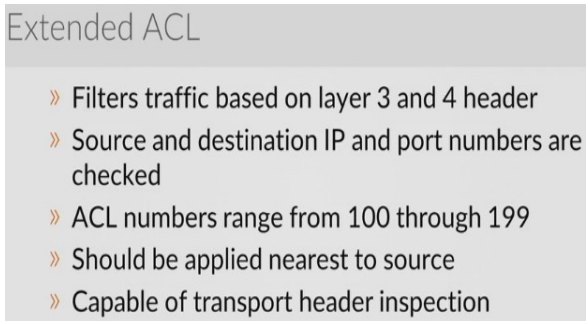
- Packet filtering mechanism
- Can filter packets on the basis of Layer 3 and Layer 4 header
- Should have at least one permit statement
- Works in sequential order; statement with lower sequence is preferred and checked

- Only one ACL can be applied per interface, per direction
- Can be applied inbound and outbound
  - Inbound - before routing
  - Outbound - after routing
- Implicit deny rule applied at the end of the sequence if nothing has been defined

1. There is a deny all at the end of the list
2. Every entry in an ACL is ACE (Access-list Entry)

(randomly generated port numbers on the PCs are called ephemeral ports)





Standard Access List: Standard Access List close to the destination are best.

R1(config)#access-list 1 permit 10.0.0.0 0.255.255.255
R1(config)#access-list 1 deny host 10.0.0.1 log
R1(config)#access-list 1 permit any any !(don't forget this as there is a default deny at the end)

Extended Access list: Extended Access lists closer to the source are best.

R1(config)#access-list 101 permit tcp 10.0.0.0 0.255.255.255 187.100.1.6 0.0.0.0 eq 20
!(187.100.1.6 0.0.0.0 is the same as host 187.100.1.6)
R1(config)#access-list 101 deny tcp any eq 22 host 10.0.0.1 range 22 23
R1(config)#access-list 101 permit ip any any dscp cs2

Apply this ACL to an interface:

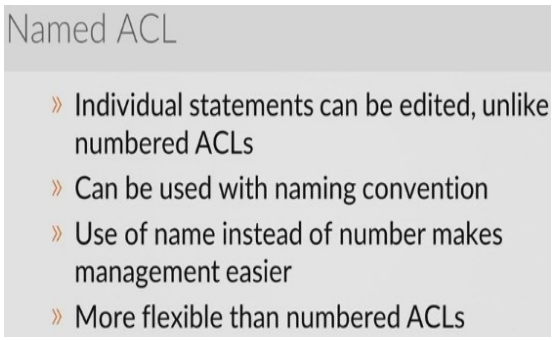
R1(config)#interface Fa0/1
R1(config-if)#ip access-group 1 out
OR
R1(config)#interface Fa0/0
R1(config-if)#ip access-group 1 in

Named Access Lists:

R(config)#ip access-list extended MyACL
R(config-ext-nacl)#100 permit ip host 1.1.1.1 any

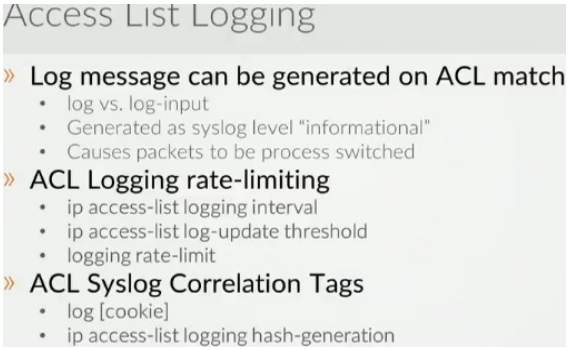
Edit and Insert Lines in Access Lists:

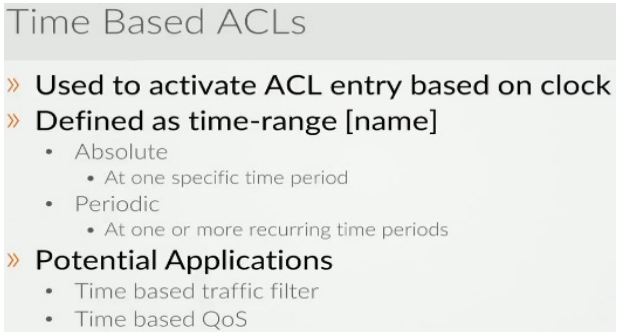
R(config)#ip access-list extended MyACL
R(config-ext-nacl)#no 500
R(config-ext-nacl)#500 permit ip any host 5.5.5.5
R(config-ext-nacl)#510 permit ip any host 6.6.6.6



Time-based ACLs:

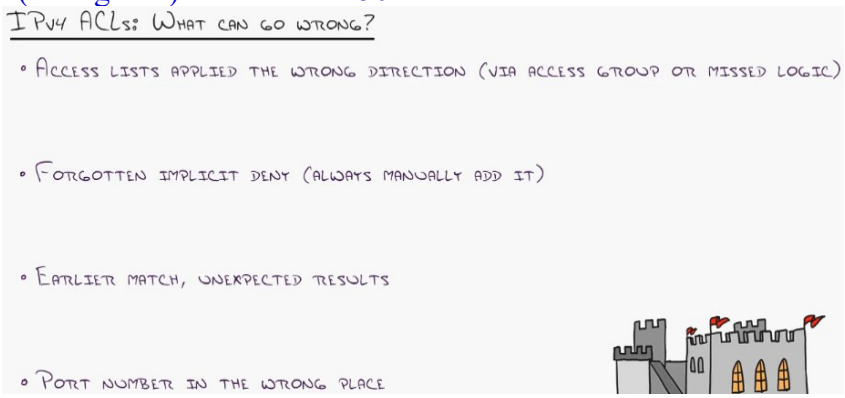
R(config)#time-range TR_WORKDAYS
R(config-time-range)#periodic weekdays 08:00 to 19:00
!(Don't configure NTP unless mentioned in the LAB)
R(config)#ip access-list extended 100
R(config-ext-nacl)#27 permit tcp any any eq www time-range TR_WORKDAYS



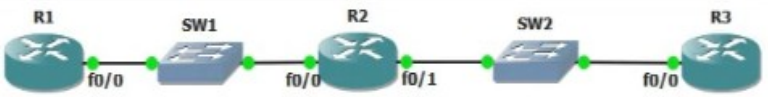


Restricting Telnet access with an Access-list:

R(config)#access-list 50 permit host 10.20.2.100
R(config)#line vty 0 15
R(config-line)#access-class 50 in



Verification and TSHOOT Commands:



(In the lab they could simply say to permit echo requests between two router (e.g. R3 and R1), but you need to be careful if you are running any routing protocols (eigrp or rip) to be permitted too, even if they haven't explicitly mentioned. Always verify by using log or log-input at the end of the ACLs and with debug commands).
R2(config)#access-list 100 permit icmp host 3.3.3.3 host 1.1.1.1 echo
R2(config)#access-list 100 permit eigrp any any
R2(config)#access-list 100 deny ip any any log
R2(config)#access-list 100 deny ip any any log-input

R(config)#ip access-list logging interval 500 !(in ms. This is the interval time it takes to send it to syslog)
R(config)#ip icmp rate-limit unreachable 100 !(in ms)
!(problem with this is that router will have to generate a huge amount of unreachable messages if someone pings hugely)

sh access-lists
sh ip access-lists
sh access-lists 1
sh ip access-lists 1
sh access-lists ACL_NAME1
sh ip access-lists ACL_NAME1
sh ip access-lists interface fa0/0
sh ip int fa0/0
sh time-range
sh time-range ACL_NAME1
sh run | inc access-list