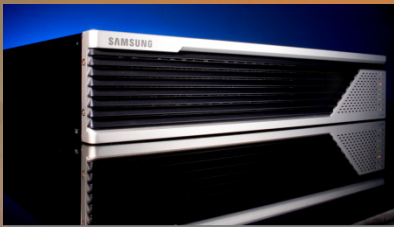


Firewall and ACL Configuration



Contents

- Overview
- Firewall Configuration
- ACL Configuration

Overview

Security Features of Ubigate

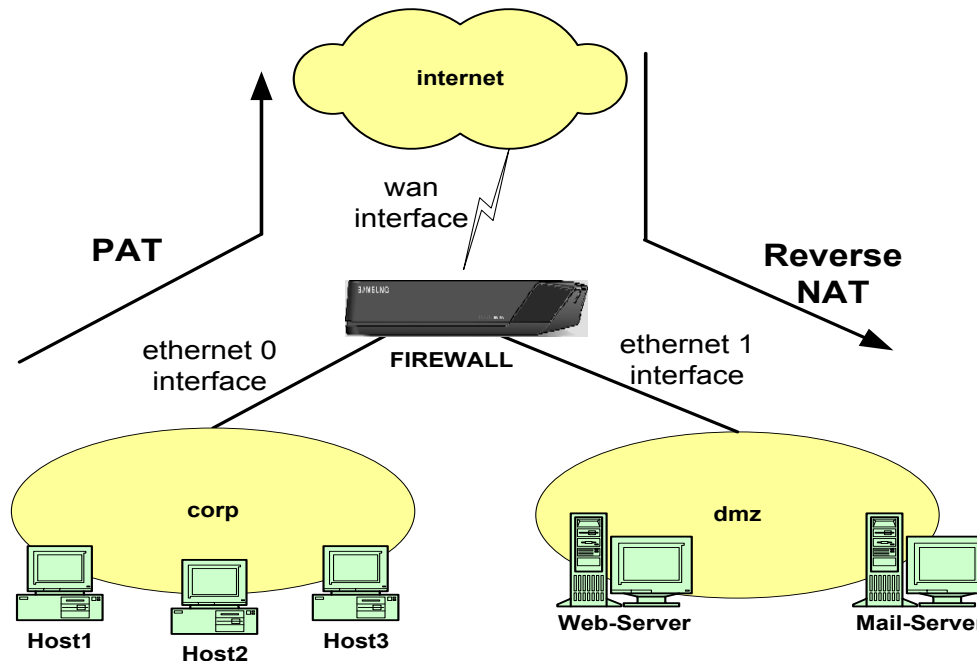
Category	Functionality
Firewall	<ul style="list-style-type: none"> • Stateful packet inspection • ALG • NAT (static, dynamic, PAT) • Application content filtering <ul style="list-style-type: none"> ✓ Block java based on extension ✓ Block URLs ✓ Block activeX ✓ FTP's protocol command ✓ SMTP's protocol command ✓ RPC program number • Attack defense <ul style="list-style-type: none"> ✓ Dos/DDos protection ✓ Fragment attack ✓ IP spoofing ✓ DMZ ✓ Rate limiting <ul style="list-style-type: none"> ✓ Max. connection ✓ Connection creation ✓ Bandwidth
VPN	<ul style="list-style-type: none"> • IPSec for site to site • IPSec for remote access • IPSec <ul style="list-style-type: none"> ✓ Protocols : ESP, AH, ESP with AH ✓ Encryption : DES-CBC, 3DES-CBC, AES-CBC ✓ Hashing : HMAC-SHA1, HMAC-MD5

Category	Functionality
VPN	<ul style="list-style-type: none"> • IKE <ul style="list-style-type: none"> ✓ Authentication : pre-shared, RSA/DSA signature, Xauth ✓ Mode : Main, Aggressive, Quick ✓ Diffie-hellman group : 1, 2, 5 ✓ Encryption : DES, 3DES, AES ✓ Hash : SHA1, MD5 • L2TP, GRE, NAT traversal • PKI support (SCEP, manual, CRL, OCSP)
Internal Security	<ul style="list-style-type: none"> • AAA, 802.1x, ACL, MAC address filtering



Overview

- Stateful inspection firewall for IPv4.
- Packets are allowed or denied to be forwarded through the system, based on pre-defined policies.
- Offers a rich set of features such as protection against DOS (Denial of Service) attacks, Network Address Translation (NAT), etc.





- **Stateful Inspection**
- **DOS Attack Protection**
- **Network Address Translation**
- **Application Content Filtering**
- **URL Key-word Filtering**
- **Traffic Scheduling**
- **Rate Limiting**
- **Application Level Gateway**

● Overview

- Stateful packet inspection firewall maintains a table of active sessions/connections.
- Entries are created only for those connections/streams that satisfy a defined security policy; packets associated with these sessions are permitted to pass through the firewall.
- Each connection will have a timeout period associated with it.
- Eg: TCP connection.
 - First packet should be TCP syn.
 - When firewall receives a TCP syn packet, it check the security policies.\
 - If this TCP connection is permitted then an entry is made for this connection.
 - Firewall will keep track of the state of the tcp connection and the tcp sequence number.
 - Firewall will expect the next packet to be TCP SYN ACK.
- Eg: DNS request
 - First packet should be DNS request.
 - Firewall will keep track of the DNS request ID.
 - DNS reply should have the same ID.

● DOS : Denial Of Service

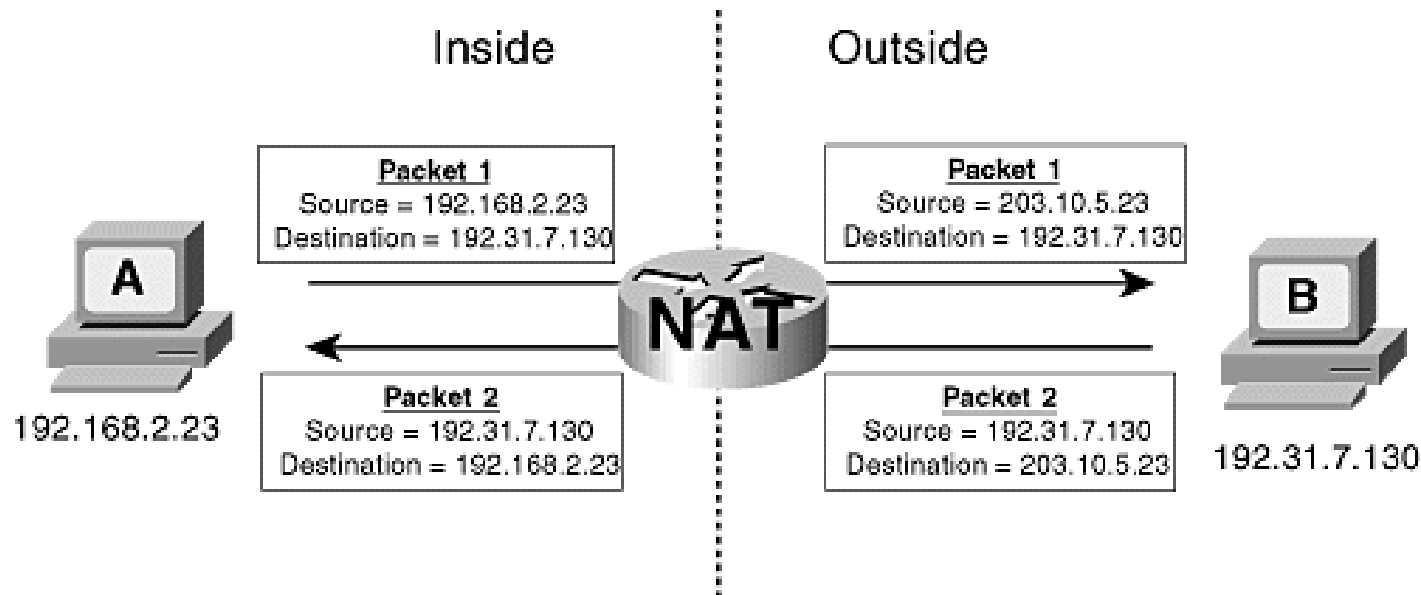
- Type of attack on a network that is designed to bring the network to its knees by flooding it with useless traffic.
- Eg: TCP Syn flooding : generate many half-open TCP connections by sending SYN packets to the target without replying to the following SYN-ACK packet
- Boink Attack, teardrop attack: IP reassembly attack
- Smurf Attack : Attacker sends a large amount of ICMP echo (ping) traffic at IP broadcast addresses, all of it having a spoofed source address of a victim.

Network Address Translation



Basic NAT Concept

- Inside – Private network
- Outside – Public network



- **Static NAT (one-to-one)**
 - Mapping a private IP address to a public IP address on a one-to-one basis
- **Dynamic NAT (many-to-many)**
 - Maps a private IP address to a public IP address from a group of public IP addresses on a many-to-many basis.
- **PAT (Port Address Translation) (many-to-one)**
 - A form of dynamic NAT that maps multiple private IP addresses to a single public IP address by using different ports. This is a many-to-one mapping.
- **Reverse NAT**
 - Used for accessing internal servers in DMZ from outside world.

● Overview

- Firewall supports command level filtering for certain well known applications
- Firewall has the necessary intelligence to parse the contents of these applications and selectively filter out some commands
- FTP Filter : Filtering FTP commands (put, get etc..)
- HTTP Filter : Filtering based on file extension (*.txt,*.ocx etc..)
- SMTP Filter : Filtering SMTP commands (mail, rcpt etc..)
- RPC Filter : Filtering based on RPC program number.

URL Key-Word Filtering



- Block the access to certain URLs/websites.
- Eg: block the access to mail and sports sites.

Traffic Schedule



- Traffic can be permitted/denied based on the time.
- Eg: Permitting/denying certain Traffic during office hours

Rate Limiting



- Limiting the traffic based on
- Maximum number of connections.
- Maximum connection rate.
- Maximum Bandwidth.

● Overview

- Dynamically opening ports based on application.
- Modifying the content of the application packets.

● eg: FTP

- Control packet is thru port 21.
- Data transfer will be thru a different port, FTP ALG will open this port

● eg: SIP – Session Initiation Protocol

- Control packets will use UDP, TCP ports 5060
- RTP packets will use UDP ports which are opened dynamically by SIP ALG by checking the contents of control packets.

Firewall

How to Create a Firewall Policy?



● Type of Traffic

- Transit : Traffic passing thru Firewall
- Self : Traffic initiated from the Firewall/Traffic destined to Firewall interfaces.

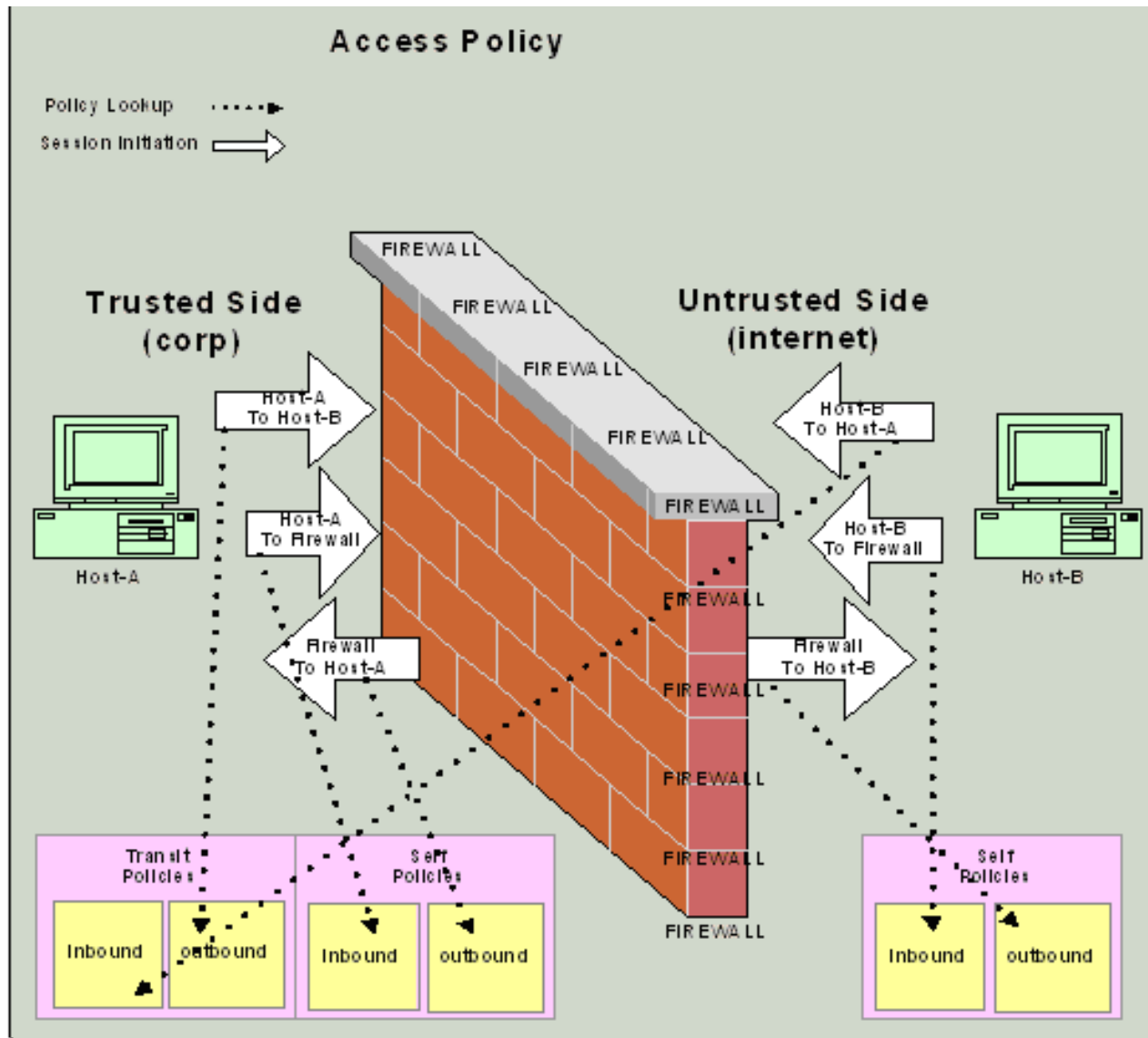
● Direction of Traffic

- Inbound: Traffic coming towards a map.
- OutBound : Traffic going out from a map.

● Other Parameters

- Source Address
- Destination address
- Protocols-UDP, TCP,ICMP etc..
- Port Number
- Service – FTP,Telnet,HTTP etc..

How to Create a Firewall Policy?





● Enabling Firewall

- Corp : trusted
- Internet : un-trusted

```
ibg01/configure#firewall corp
ibg01/configure/firewall corp# end

ibg01/configure#firewall internet
ibg01/configure#firewall internet#

ibg01/configure#firewall dmz
ibg01/configure#firewall dmz#
```

● Adding Ethernet 0/0 to corp map

```
ibg01/configure#firewall corp
ibg01/configure/firewall corp# interface ethernet0/0
```

● Verify the firewall interface

```
ibg01# show firewall interface corp
```

Interface	Map Name
-----	-----
ethernet0/0	corp

```
ibg01#
```

● Verify the default policy

```
ibg01# show firewall policy corp
ibg01# show firewall policy internet
ibg01# show firewall policy dmz
```

- Default policy on corp and dmz maps
 - Permit all outgoing traffic and deny all in incoming traffic.
 - Permit both incoming and outgoing self traffic.
- Default policy on internet map
 - Internet map will have only self polices.
 - Permit all outgoing self traffic and deny all incoming self traffic.

● Permit traffic from corp to internet

- Create an outbound policy on corp map for permitting telnet from PC#1 (10.10.10.10) to PC#2 (30.30.30.30)

```
Ibg01/configure/firewall corp# policy 100 out address  
10.10.10.10 32 30.30.30.30 32 service telnet
```

- Verify the corp policy

```
Ibg01# show firewall policy corp
```

- Verify the connections on the iBG, 1 TCP connection should be displayed.

```
Ibg01# show firewall connections all
```

● Deny traffic from corp to internet

- Create an outbound policy on corp map for denying FTP from PC#1 (10.10.10.10) to PC#2 (30.30.30.30)

```
Ibg01/configure/firewall corp# policy 99 out address  
10.10.10.10 32 30.30.30.30 32 deny service ftp
```

- Verify the corp policy

```
Ibg01# show firewall policy corp
```

- Verify the connections on the iBG, FTP connection should NOT be displayed.

```
Ibg01# show firewall connections all
```



● Permit traffic from internet to corp

- Create an inbound policy on corp map for permitting traffic from PC#2 (30.30.30.30) to PC#1 (10.10.10.10).

```
Ibg01/configure/firewall corp# policy 50 in address  
30.30.30.30 32 10.10.10.10 32
```

- Verify the corp policy

```
Ibg01# show firewall policy corp
```

- Verify the connections on the iBG, 1 TCP connection should be displayed.

```
Ibg01# show firewall connections all
```



● Firewall NAT

- NAT can be enabled for traffic from corp to internet.
- Create an outbound policy on corp map for applying NAT with source IP address 20.20.20.1

```
Ibg01/configure/firewall corp# policy 10 out nat-ip  
20.20.20.1
```

- Verify the corp policy

```
Ibg01# show firewall policy corp
```

- Verify the NAT translation, IP address, 10.10.10.10, should get translated to 20.20.20.1

```
Ibg01# show firewall nat-translations all
```

● Firewall static NAT

- Create a static NAT object with NAT IP 20.20.20.1

```
Ibg01/configure# firewall global
Ibg01/configure/firewall global# object
Ibg01/configure/firewall global/object# nat-pool stat_nat
static 20.20.20.1
```

- Global map is used for defining global parameters and global configuration objects for the firewall policies.
- Configuring an outbound policy on corp map and attach the static nat objects.

```
Ibg01/configure/firewall corp# policy 5 out address
10.10.10.10 32 any
Ibg01/configure/firewall corp/policy 5 out# apply-object nat-
pool stat_nat
```

Basic Configuration



- Verify the static nat object

```
Ibg01/configure# show firewall object nat-pool global
```

- Verify the corp policy

```
Ibg01/configure# show firewall policy corp  
Ibg01/configure# show firewall policy corp detail priority 5
```

- Verify the NAT translation, IP address 150.1.1.2 should always get translated to 200.1.1.50

```
Ibg01/configure# show firewall nat-translations all
```


● Firewall dynamic NAT

- Create a dynamic nat object with NAT IP range 20.20.20.1 to 20.20.20.20
- Create an outbound policy on corp map and attach the dynamic nat object.

```
Ibg01/configure/firewall global/object# nat-pool dyn_nat  
dynamic 20.20.20.1 20.20.20.20
```

- Create an outbound policy on corp map and attach the dynamic nat object.

```
Ibg01/configure/firewall corp# policy 4 out address 10.10.10.0  
24 any  
Ibg01/configure/firewall corp/policy 4 out# apply-object nat-  
pool dyn_nat
```

Basic Configuration



- Verify the dynamic nat object

```
Ibg01/configure# show firewall object nat-pool global
```

- Verify the corp policy

```
Ibg01/configure# show firewall policy corp  
Ibg01/configure# show firewall policy corp detail priority 4
```

- Verify the NAT translation. IP address 10.10.10.0 should always get translated to any of the IP address in the range of 20.20.20.1 to 20.20.20.20, depending on the availability of the free NAT IP.

```
Ibg01/configure# show firewall nat-translations all
```



● Firewall reverse NAT

- Create an inbound policy on DMZ map for permitting traffic from internet to DMZ with reverse NAT. All packets destined to 200.1.1.14 will be forwarded to 160.1.1.2.

```
Ibg01/configure/firewall DMZ# policy 100 in address any  
200.1.1.15 32 nat-ip 160.1.1.2
```

- Verify the DMZ policy

```
Ibg01/configure# show firewall policy DMZ  
Ibg01/configure# show firewall policy DMZ detail priority 100
```

- Verify the connections on the iBG, one ICMP connection should be displayed.

```
Ibg01/configure# show firewall connections all
```

Basic Configuration



- Verify the NAT translation, IP address 200.1.1.15 should get translated to 160.1.1.2 and traffic to 200.1.1.15 should reach 160.1.1.2

```
Ibg01/configure# show firewall nat-translations all
```

- **Self traffic to the iBG can be permitted or denied.**
 - Create a self inbound policy on internet map for permitting traffic from PC to iBG.

```
Ibg01/configure/firewall internet# policy 1 in self
```

- Verify the self policy on internet map

```
Ibg01# show firewall policy internet
```

- Verify the connections on the iBG, 1 TCP connection should be displayed.

```
Ibg01# show firewall connections all
```



- **DOS (Denial Of Service Attack) protection can be enabled.**

- Enabling DOS protect

```
Ibg01/configure/firewall global# dos-protect  
Ibg01/configure/firewall global/dos-protect# enable-all
```

- Verify the DOS protect

```
Ibg01# show firewall dos-protect
```

- Verify the firewall statistics

```
Ibg01# show firewall statistics
```


- **Traffic can be controlled using schedule objects.**

- Create a schedule object named “test” for Monday 9am to 5pm.

```
Ibg01/configure/firewall global/object# schedule test week-day  
mon fri start-time 9 00 end-time 17 00
```

- Create an outbound policy on corp map and attach the schedule object.

```
Ibg01/configure/firewall corp# policy 1 out  
Ibg01/configure/firewall corp/policy 1 out# apply-object  
schedule test
```

- Now set the date and time to Monday 8 50 am.

```
Ibg01/configure# date 5 29 2006  
Ibg01/configure# time 8 50
```

- telnet/ping from PC#1 to PC#2

● Verify

- Verify the schedule object

```
Ibg01# show firewall object schedule global
```

- Verify the corp policy, schedule object should get applied to the policy.

```
Ibg01# show firewall policy corp  
Ibg01# show firewall policy corp detail priority 1
```

- Verify the date and time

```
Ibg01# show date
```

- telnet/ping traffic from PC#1 to PC#2 before 9 am should fail.

Access Control List

● Introduction

- An access control list is a sequential list of instructions to either permit or deny access through a router interface based on IP address or other criteria.
- ACLs can be used to provide a basic level of security for accessing the network.
- ACL is not stateful.

● Types

● IP ACL

- Filtering based on IP & Layer 4 (TCP/UDP) header information
- When it is applied for network module based Ethernet Interface Card (ESG, LMP, LMF, LMG), it just supports inbound filtering
- Header fields used for matching are
 - Source IP
 - Destination IP
 - Protocol : TCP/UDP/IGMP/ICMP etc.
 - TCP/UDP ports
 - TCP flags : fin, rst, psh, syn, urg, ack



- IP precedence
- Type of service
- ICMP type / code
- Differentiated Services Code Point
- Non initial IP fragments

● MAC ACL

- Filtering based on layer 2 header information
- It is working only for network module based Ethernet Interface Card (ESG, LMP, LMF, LMG card) and on 3026/2016 models
- It just supports inbound filtering
- MAC header fields used for filtering are
 - Source MAC
 - Destination MAC
 - Ethernet type : ARP/IP
 - VLAN ID
 - COS (Class of service)

- Create an IP ACL, add a rule to it and attach the ACL to an interface.

- Create an IP ACL named "test".

```
Ibg01/configure# ip access-list test
```

- Add a rule in the ACL.

```
Ibg01/configure/ip/access-list test# add permit ip any any
```

- Attach the ACL to the ethernet0/0 interface in the inbound direction.

```
Ibg01/configure# access-group ethernet0/0 in ip test
```

● Verify

- Verify that ACL named test created.

```
Ibg01# show ip access-list test
```

- Verify the ACL attached to the interface.

```
Ibg01# show ip access-list-rule ethernet0/0
```

Insert rule to an existing ACL



- **Additional rules can be inserted into an already existing ACL.**

- Insert a rule at the beginning of the ACL rule list.

```
Ibg01/configure# ip access-list test  
Ibg01/configur/ip/access-list test# insert 1 deny icmp any any
```

- **Verify**

- Verify that new rule is inserted into the ACL.

```
Ibg01# show ip access-lists test
```


Delete rule from existing ACL



- Rules can be deleted from an already existing ACL.

- Delete the rule at the beginning of the ACL rule list.

```
Ibg01/configure# ip access-list test  
Ibg01/configure/ip/access-list test# delete 1
```

- Verify

- Verify that rule is deleted from the ACL.

```
Ibg01# show ip access-list test
```



● Remove an ACL

- Remove the ACL

```
Ibg01/configure# no ip access-list test
```

● Verify

- Verify that ACL is removed

```
Ibg01# show ip access-lists test  
Ibg01# show ip access-lists all  
Ibg01# show ip access-lists-rules ethernet0/0
```

Permit inbound telnet traffic



- Telnet traffic can be permitted using ACL.

- Create an IP ACL named "test".

```
Ibg01/configure# ip access-list test
```

- Add a rule in the ACL for permitting tcp traffic from PC#1 to PC#2 to destination port 23 (telnet)

```
Ibg01/configure/ip/access-list test# add permit tcp  
10.10.10.10 30.30.30.30 dport =23
```

- Attach the policy to the ethernet3/0 interface in inbound direction.

```
Ibg01/configure# access-group ethernet0/0 in ip test
```

- Telnet from PC#1 to PC#2.
- Ping from PC#1 to PC#2.

Permit inbound telnet traffic



● Verify

- Verify the ACL

```
Ibg01# show ip access-lists test
```

- Verify the ACL attached to the interface

```
Ibg01# show ip access-lists-rules ethernet0/0
```

- Telnet from PC#1 to PC#2 should be successful, ACL should permit telnet traffic.
- Ping should fail; ACL should deny any traffic other than telnet.

Deny outbound telnet traffic

- Telnet traffic can be denied using ACL.

- Create an IP ACL named "test".

```
Ibg01/configure# ip access-list test
```

- Add rule in the ACL for permitting all traffic.

```
Ibg01/configure/ip/access-list test# add permit ip any any
```

- Insert a rule at the beginning of ACL for denying telnet from PC#1 to PC#2.

```
Ibg01/configure/ip/access-list test# insert 1 deny tcp  
10.10.10.10 30.30.30.30 dport =23
```

- Attach the policy to the ethernet0/2 interface in outbound direction.

```
Ibg01/configure# access-group ethernet0/0 out ip test
```

- Telnet from PC#1 to PC#2.
- Ping from PC#1 to PC#2.

Deny outbound telnet traffic

● Verify

- Verify the ACL.

```
Ibg01# show ip access-lists test
```

- Verify the ACL attached to the interface

```
Ibg01# show ip access-lists-rules ethernet0/0
```

- Telnet from PC#1 to PC#2 should fail, ACL should deny telnet traffic.
- Ping should be successful; ACL should permit any traffic other than telnet.

Thank you!

