

Lab 9.2.5 Configure Object Groups and Nested Object Groups using CLI

Objective

In this lab, the students will complete the following tasks:

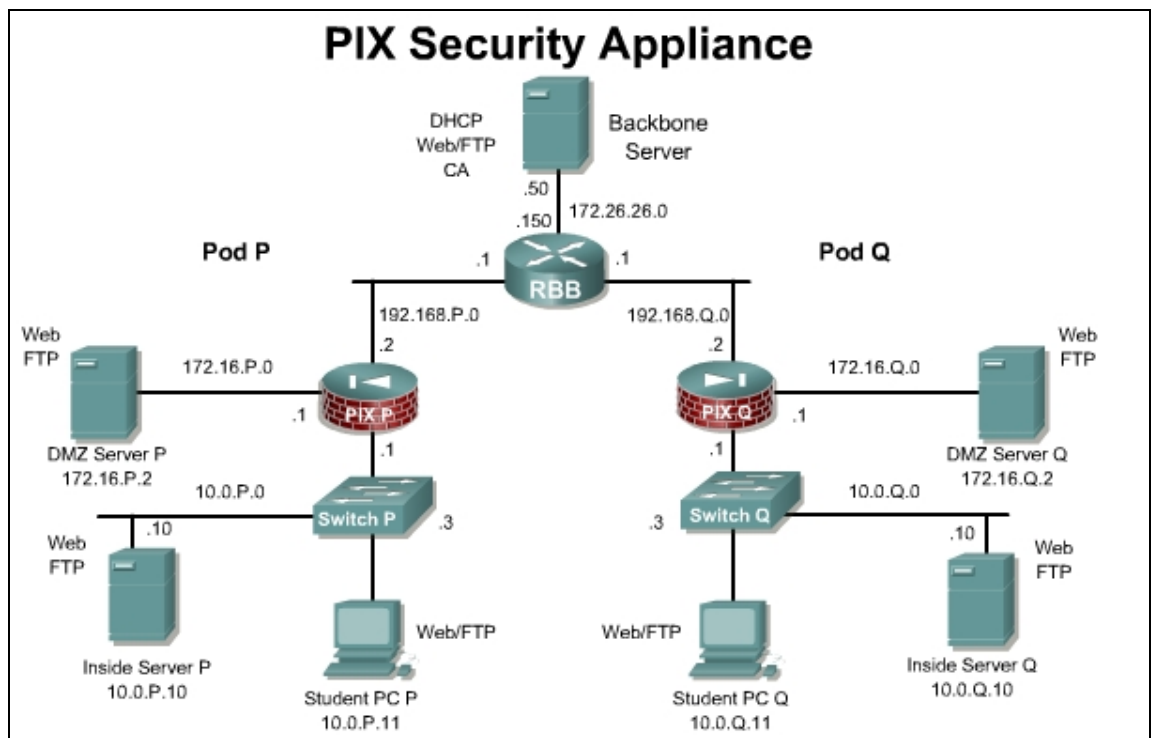
- Configure a service, ICMP-Type, and nested server object group.
- Configure an inbound access control list (ACL) with object groups.
- Configure web and ICMP access to the inside host.
- Test and verify the inbound ACL.

Scenario

In the previous lab, ASDM was used to configure a service object group. ASDM has some limitations when adding, editing, and deleting some object group types. CLI is the preferred method to handle object groups and nested object groups.

Topology

This figure illustrates the lab network environment.



Preparation

Begin with the standard lab topology and verify the starting configuration on the pod PIX Security Appliances. Access the PIX Security Appliance console port using the terminal emulator on the student PC. If desired, save the PIX Security Appliance configuration to a text file for later analysis.

Tools and resources

In order to complete the lab, the following is required:

- Standard PIX Security Appliance lab topology
- Console cable
- HyperTerminal

Additional materials

Further information about the objectives covered in this lab can be found at:

http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_configuration_guide_chapter09186a0080423271.html#wp1053224

Command list

In this lab exercise, the following commands will be used. Refer to this list if assistance or help is needed during the lab exercise.

Command	Description
access-group <i>access-list</i> { in out } interface <i>interface_name</i> [<i>per-user-override</i>]	Binds the access list to an interface. Configuration mode.
access-list <i>id</i> [line <i>line-number</i>] [extended] { deny permit } { protocol object-group <i>protocol_obj_grp_id</i> } { host <i>source-ip</i> <i>source-ip mask</i> interface <i>ifc_name</i> object-group <i>network_obj_grp_id</i> any } { host <i>destination-ip</i> <i>destination-ip mask</i> interface <i>ifc_name</i> object-group <i>network_obj_grp_id</i> any } [log [[<i>level</i>] [<i>interval secs</i>] disable default]] [inactive time-range <i>time_range_name</i>]	Create an access list.
object-group { protocol network icmp-type } <i>obj_grp_id</i>	The object-group command is used to define a protocol, network, or icmp-type object group.
object-group service <i>obj_grp_id</i> { tcp udp tcp-udp }	Defines a group of TCP/UDP port specifications such as eq smtp and range 2000 2010 .
show running-config object-group [<i>id grp_id</i> <i>grp_type</i>]	Displays the current object groups in the configuration.

Step 1 Configure a Service Group Containing HTTP and FTP

To configure a service group containing HTTP and FTP, complete the following steps:

- a. Delete the ACL configured in the previous lab.

```
PixP(config)# clear configure access-list outside_access_in
```

- b. Verify that the ACL has been removed:

```
PixP(config)# show running-config access-list
```

- c. Delete the service group configured in the previous lab.

```
PixP(config)# no object-group service DMZ_SERVICES tcp
```

- d. Create a TCP service group named **MYSERVICES**. This step assigns a name to the group and enables the service object subcommand mode:

```
PixP(config)# object-group service MYSERVICES tcp
```

- b. Add HTTP and FTP to the service object group:

```
PixP(config-service)# port-object eq http
```

```
PixP(config-service)# port-object eq ftp
```

1. What is the command to group consecutive services?
-

- c. Return to configuration mode:

```
PixP(config-service)# exit
```

- d. Verify that the object group has been configured successfully:

```
PixP(config)# show running-config object-group
```

```
object-group service MYSERVICES tcp
```

```
port-object eq www
```

```
port-object eq ftp
```

Step 2 Configure an ICMP-Type Group

To configure an ICMP-Type group, complete the following steps:

- a. To assign a name to the group and enable the ICMP-Type subcommand mode, create an ICMP-Type object group named PING:

```
PixP(config)# object-group icmp-type PING
```

- b. Add ICMP echo to the ICMP-Type object group:

```
PixP(config-icmp-type)# icmp-object echo
```

- c. Add ICMP echo-reply to the ICMP-Type object group:

```
PixP(config-icmp-type)# icmp-object echo-reply
```

- d. Add ICMP unreachable messages to the ICMP-Type object group:

```
PixP(config-icmp-type)# icmp-object unreachable
```

- e. Return to configuration mode:

```
PixP(config-icmp-type)# exit
```

- f. Verify that the object group has been configured successfully:

```
PixP(config)# show running-config object-group  
object-group service MYSERVICES tcp  
port-object eq www  
port-object eq ftp  
object-group icmp-type PING  
icmp-object echo  
icmp-object echo-reply  
icmp-object unreachable
```

Step 3 Nest an Object Group Within Another Object Group

To nest an object group within another object group, complete the following steps:

- a. Create a network object group named FTPSERVERS:

```
PixP(config)# object-group network FTPSERVERS
```

- b. Add the bastion host to the object group:

```
PixP(config-network)# network-object host 192.168.P.11  
(where P = pod number)
```

- c. Return to configuration mode:

```
PixP(config-network)# exit
```

- d. Create a network object group named ALLSERVERS:

```
PixP(config)# object-group network ALLSERVERS
```

- e. Nest the FTPSERVERS group within the ALLSERVERS group:

```
PixP(config-network)# group-object FTPSERVERS
```

- f. Add the following servers to the ALLSERVERS group:

- 192.168.P.10
- 192.168.P.6
- 192.168.P.7

```
PixP(config-network)# network-object host 192.168.P.10
```

```
PixP(config-network)# network-object host 192.168.P.6
```

```
PixP(config-network)# network-object host 192.168.P.7
```

(where P = pod number)

- g. Verify that the object group has been configured successfully:

```
PixP(config-network)# show running-config object-group  
object-group service MYSERVICES tcp  
port-object eq www  
port-object eq ftp  
object-group icmp-type PING  
icmp-object echo  
icmp-object echo-reply
```

```

icmp-object unreachable
object-group network FTPSERVERS
    network-object host 192.168.P.11
object-group network ALLSERVERS
    group-object FTPSERVERS
    network-object host 192.168.P.10
    network-object host 192.168.P.6
    network-object host 192.168.P.7

```

(where P = pod number)

- f. Return to configuration mode:

```
PixP(config-network)# exit
```

(where P = pod number)

Step 4 Configure an Inbound ACL With Object Groups

Complete the following steps to configure an inbound ACL to perform the following:

- Allow inbound web traffic from a peer pod network to the bastion host.
 - Allow inbound FTP traffic from a peer pod internal host to the bastion host.
- a. Test web access to the peer pod bastion host by completing the following substeps. The test to the peer bastion host should fail.
 - i. Open a web browser on the student PC.
 - ii. Use the web browser to access the bastion host of the peer pod group by entering **http://192.168.Q.11**.
(where Q = peer pod number)
 - b. Test web access to the inside host of the peer pod by completing the following substeps. The test to the peer inside host should fail.
 - i. Open a web browser on the student PC.
 - ii. Use the web browser to access the inside host of the peer pod group by entering **http://192.168.Q.10**.
(where Q = peer pod number)
Why have these connection attempts failed?

- c. From the FTP client, test FTP access to the peer pod bastion host. Access to the peer bastion host using FTP should fail.

Start>Run>ftp 192.168.Q.11

(where Q = peer pod number)

- d. Use the MYSERVICES group to create an ACL permitting inbound web and FTP access to the bastion host from the peer outside network:

```
PixP(config)# access-list ACLIN permit tcp 192.168.Q.0 255.255.255.0
object-group FTPSERVERS object-group MYSERVICES
```

(where Q = peer pod number)

- e. Bind the ACL to the outside interface:

```
PixP(config)# access-group ACLIN in interface outside
```

- f. View the ACLs:

```
PixP(config)# show access-list  
  
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max  
4096)          alert-interval 300  
  
access-list ACLIN; 2 elements  
  
access-list ACLIN line 1 extended permit tcp 192.168.Q.0  
255.255.255.0 object-group FTPSERVERS object-group MYSERVICES  
  
access-list ACLIN line 1 extended permit tcp 192.168.Q.0  
255.255.255.0 host 192.168.P.11 eq www (hitcnt=0)  
  
access-list ACLIN line 1 extended permit tcp 192.168.Q.0  
255.255.255.0 host 192.168.P.11 eq ftp (hitcnt=0)  
  
PixP(config)#
```

- g. Ping the peer pod inside host. The ping should fail.

```
C:\>ping 192.168.Q.10  
  
Pinging 192.168.Q.10 with 32 bytes of data:  
  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.  
  
(where Q = peer pod number)
```

- h. Ping the peer pod bastion host. The ping should fail.

```
C:\>ping 192.168.Q.11  
  
Pinging 192.168.Q.11 with 32 bytes of data:  
  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.  
  
(where Q = peer pod number)
```

- i. Test web access to the peer pod bastion host by completing the following substeps. Access to the peer bastion host should be successful.
- Open a web browser on the student PC.
 - Use the web browser to access the bastion host of the peer pod group by entering **http://192.168.Q.11**.
(where Q = peer pod number)
- j. Test web access to the peer pod inside host by completing the following substeps. Access to the peer pod inside host should fail.
- Open a web browser on the client PC.
 - Use the web browser to access the inside host of the peer pod group by entering **http://192.168.Q.10**.
(where Q = peer pod number)

- k. From the FTP client, test FTP access to the peer pod bastion host. Access to the peer bastion host via FTP should be successful.

```
Start>Run>ftp 192.168.Q.11
```

(where Q = peer pod number)

- l. From the FTP client, test FTP access to the peer pod inside hosts. Access to the peer inside host via FTP should fail.

```
Start>Run>ftp 192.168.Q.10
```

(where Q = peer pod number)

1. Why does the connection attempt to the peer pod inside host fail?

Step 5 Configure ACLIN

Complete the following steps to configure ACLIN to perform the following:

- Permit inbound web and ICMP traffic to all hosts behind the PIX Security Appliance
 - Deny all other traffic from the Internet
- a. Use a network hosts group to add an ACL entry permitting web traffic to all hosts behind the PIX Security Appliance:

```
PixP(config)# access-list ACLIN permit tcp any object-group  
ALLSERVERS eq www
```

- b. Permit ICMP traffic to all hosts behind the PIX Security Appliance:

```
PixP(config)# access-list ACLIN permit icmp any any object-group  
PING
```

- c. Deny all other traffic from the Internet:

```
PixP(config)# access-list ACLIN deny ip any any
```

- d. Bind the ACL to the outside interface:

```
PixP(config)# access-group ACLIN in interface outside
```

- e. Create an ACL to permit echo replies to the inside host from the bastion host:

```
PixP(config)# access-list ACLDMZ permit icmp any any object-group  
PING
```

- f. Bind the ACL to the demilitarized zone (DMZ) interface:

```
PixP(config)# access-group ACLDMZ in interface dmz
```

- g. Display the ACLs and observe the hit counts:

```
PixP(config)# show access-list  
  
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max  
4096)          alert-interval 300  
  
access-list ACLIN: 10 elements  
  
access-list ACLIN line 1 extended permit tcp 192.168.Q.0  
255.255.255.0 object-group FTPSERVERS object-group MYSERVICES  
  
access-list ACLIN line 1 extended permit tcp 192.168.Q.0  
255.255.255.0 host 192.168.1.11 eq www (hitcnt=0)
```

```

access-list ACLIN line 1 extended permit tcp 192.168.Q.0
255.255.255.0 host 192.168.1.11 eq ftp (hitcnt=0)

access-list ACLIN line 2 extended permit tcp any object-group
ALLSERVERS eq www

access-list ACLIN line 2 extended permit tcp any host 192.168.P.11
eq www (hitcnt=0)

access-list ACLIN line 2 extended permit tcp any host 192.168.P.10
eq www (hitcnt=0)

access-list ACLIN line 2 extended permit tcp any host 192.168.P.6 eq
www (hitcnt=0)

access-list ACLIN line 2 extended permit tcp any host 192.168.P.7 eq
www (hitcnt=0)

access-list ACLIN line 3 extended permit icmp any any object-group
PING

access-list ACLIN line 3 extended permit icmp any any echo
(hitcnt=0)

access-list ACLIN line 3 extended permit icmp any any echo-reply
(hitcnt=0)

access-list ACLIN line 3 extended permit icmp any any unreachable
(hitcnt=0)

access-list ACLIN line 4 extended deny ip any any (hitcnt=0)

access-list ACLDMZ; 3 elements

access-list ACLDMZ line 1 extended permit icmp any any object-group
PING

access-list ACLDMZ line 1 extended permit icmp any any echo
(hitcnt=0)

access-list ACLDMZ line 1 extended permit icmp any any echo-reply
(hitcnt=0)

access-list ACLDMZ line 1 extended permit icmp any any unreachable
(hitcnt=0)

(where P=pod number, and Q = peer pod number)

```

Step 6 Test and Verify the Inbound ACL

Complete the following steps to test the inbound ACL:

- a. Ping the inside host of the peer pod:

```
C:\>ping 192.168.Q.10
```

Pinging 192.168.Q.10 with 32 bytes of data:

```
Reply from 192.168.Q.10: bytes=32 time<10ms TTL=128
```

```
Reply from 192.168.Q.10: bytes=32 time<10ms TTL=128
```

```
Reply from 192.168.Q.10: bytes=32 time<10ms TTL=128
```

```
Reply from 192.168.Q.10: bytes=32 time<10ms TTL=128
```

(where Q = peer pod number)

- b. Ping the bastion host of the peer pod:

```
C:\>ping 192.168.Q.11
```

Pinging 192.168.Q.11 with 32 bytes of data:


```
Reply from 192.168.Q.11: bytes=32 time<10ms TTL=128
Reply from 192.168.Q.11: bytes=32 time<10ms TTL=128
Reply from 192.168.Q.11: bytes=32 time<10ms TTL=128
Reply from 192.168.Q.11: bytes=32 time<10ms TTL=128
(where Q = peer pod number)
```

- c. From the student PC, ping the bastion host:

```
C:\>ping 172.16.P.2
Pinging 172.16.P.2 with 32 bytes of data:
Reply from 172.16.P.2: bytes=32 time<10ms TTL=128
Reply from 172.16.P.2: bytes=32 time<10ms TTL=128
Reply from 172.16.P.2: bytes=32 time<10ms TTL=128
Reply from 172.16.P.2: bytes=32 time<10ms TTL=128
(where P = pod number)
```

- d. From the student PC, ping the Backbone server:

```
C:\>ping 172.26.26.50
Pinging 172.26.26.50 with 32 bytes of data:
Reply from 172.26.26.50: bytes=32 time<10ms TTL=128
Reply from 172.26.26.50: bytes=32 time<10ms TTL=128
Reply from 172.26.26.50: bytes=32 time<10ms TTL=128
Reply from 172.26.26.50: bytes=32 time<10ms TTL=128
```

- e. Test web access to the peer pod bastion host by completing the following substeps. Access to the peer bastion host should be successful.
- Open a web browser on the student PC.
 - Use the web browser to access the bastion host of the peer pod group by entering **http://192.168.Q.11**.
(where Q = peer pod number)
- f. Test web access to the peer pod inside host by completing the following substeps. Access to the peer pod inside host should now be successful.
- Open a web browser on the client PC.
 - Use the web browser to access the inside host of the peer pod group by entering **http://192.168.Q.10**.
(where Q = peer pod number)
- g. From the FTP client, test FTP access to the peer pod bastion host. Access to the peer bastion host via FTP should be successful.
- ```
Start>Run>ftp 192.168.Q.11
```
- (where Q = peer pod number)
- h. From the FTP client, test FTP access to the peer pod inside host. Access to the peer inside host via FTP should fail.
- ```
Start>Run>ftp 192.168.Q.10
```
- (where Q = peer pod number)

- i. Display the ACLs again and observe the hit counts:

```
PixP(config)# show access-list

access-list cached ACL log flows: total 0, denied 0 (deny-flow-max
4096)          alert-interval 300

access-list ACLIN; 10 elements

access-list ACLIN line 1 extended permit tcp 192.168.Q.0
255.255.255.0 object-group FTPSERVERS object-group MYSERVICES

access-list ACLIN line 1 extended permit tcp 192.168.Q.0
255.255.255.0 host 192.168.1.11 eq www (hitcnt=0)

access-list ACLIN line 1 extended permit tcp 192.168.Q.0
255.255.255.0 host 192.168.1.11 eq ftp (hitcnt=0)

access-list ACLIN line 2 extended permit tcp any object-group
ALLSERVERS eq www

access-list ACLIN line 2 extended permit tcp any host 192.168.P.11
eq www (hitcnt=0)

access-list ACLIN line 2 extended permit tcp any host 192.168.P.10
eq www (hitcnt=2)

access-list ACLIN line 2 extended permit tcp any host 192.168.P.6 eq
www (hitcnt=0)

access-list ACLIN line 2 extended permit tcp any host 192.168.P.7 eq
www (hitcnt=0)

access-list ACLIN line 3 extended permit icmp any any object-group
PING

access-list ACLIN line 3 extended permit icmp any any echo
(hitcnt=0)

access-list ACLIN line 3 extended permit icmp any any echo-reply
(hitcnt=4)

access-list ACLIN line 3 extended permit icmp any any unreachable
(hitcnt=0)

access-list ACLIN line 4 extended deny ip any any (hitcnt=0)

access-list ACLDMZ; 3 elements

access-list ACLDMZ line 1 extended permit icmp any any object-group
PING

access-list ACLDMZ line 1 extended permit icmp any any echo
(hitcnt=0)

access-list ACLDMZ line 1 extended permit icmp any any echo-reply
(hitcnt=8)

access-list ACLDMZ line 1 extended permit icmp any any unreachable
(hitcnt=0)
```