Using Access-groups to Block/Allow Traffic in AOS

When setting up an AOS unit, it is important to control which traffic is allowed in and out. In many cases, the built-in AOS firewall is the most efficient way to properly block and allow traffic through the AOS unit, especially when NAT functionality is required. The firewall is made to protect a unit with a public Internet connection from the outside world. You can read more about the AOS firewall and how to set it up at https://supportforums.adtran.com/docs/DOC-2288.

Internetworking equipment that resides on private networks is not always safe from security concerns, despite the lack of a public connection. There may be resources that should only be available to certain personnel, or networks that should not be able to communicate with other networks in the infrastructure. The AOS firewall includes many options and utilities that aren't needed in a private network security plan. In this case, access-groups can be the best option to properly control traffic, save resources on the AOS unit, and keep the configuration of the router simplistic.

Access-groups offer advantages over the AOS firewall. The most notable advantage is that access-groups process traffic before the packet is sent to be processed by other functions in the AOS unit, like SSH, HTTP, or the route-table. So when trying to simply block or allow certain types of traffic, access-groups can be a more efficient than the firewall because resources aren't allocated for the packet until it is accepted in the access-group. Furthermore, the AOS firewall blocks traffic, by default, that it considers to be a threat like traffic it detects as "spoofed" (spoofed traffic are packets that comes in an interface from a source that, according to the route table, would not be located out of that interface). On a public network, it is vital to block spoofed traffic, but on a private network there are many situations where traffic the firewall perceives as spoofed would actually be legitimate (like A-symmetrical routing). Access-groups provide the ability to make small traffic-controlling rules without blocking unwanted traffic.

Before creating an access-group, it is important to understand where you want to block/allow traffic in regards to each interface. An access-group can be applied to any interface in either the outbound or inbound direction. It is important to perform this action as close to the source of the traffic as possible, so that the traffic takes up the least amount of processing for the AOS unit. There are several examples below.

Example 1: Blocking SSH Traffic Inbound

In this example, a network administrator wants to add a rule to block SSH attempts into the AOS device on the Ethernet 0/1 interface. All other traffic is permitted. The Ethernet 0/1 interface configuration is below:

```
interface eth 0/1
  ip address 10.10.10.1 255.255.255.0
  no shutdown
```

First, an access-control list (ACL) needs to be created to take the proper actions on traffic. Since it is known that all SSH traffic to the AOS device needs to be blocked, an ACL rule can be made to deny that traffic:

```
ip access-list extended BLOCK-SSH
  deny tcp any host 10.10.10.1 eq ssh
```

Access-groups strictly use the ACL's "permit" and "deny" functions, where as the AOS firewall uses the ACL's permit and deny as a "match" and "do not match." In this case, if you do not have any permit statements in an ACL referenced as an access-group, no traffic will be permitted. In this case, the network administrator only wants to block SSH into 10.10.10.1, so a **permit ip any any** can be added at the bottom, so that all other traffic is allowed (instead of the traffic being denied by default using the implicit deny t the bottom of all ACLs). The final ACL is below:

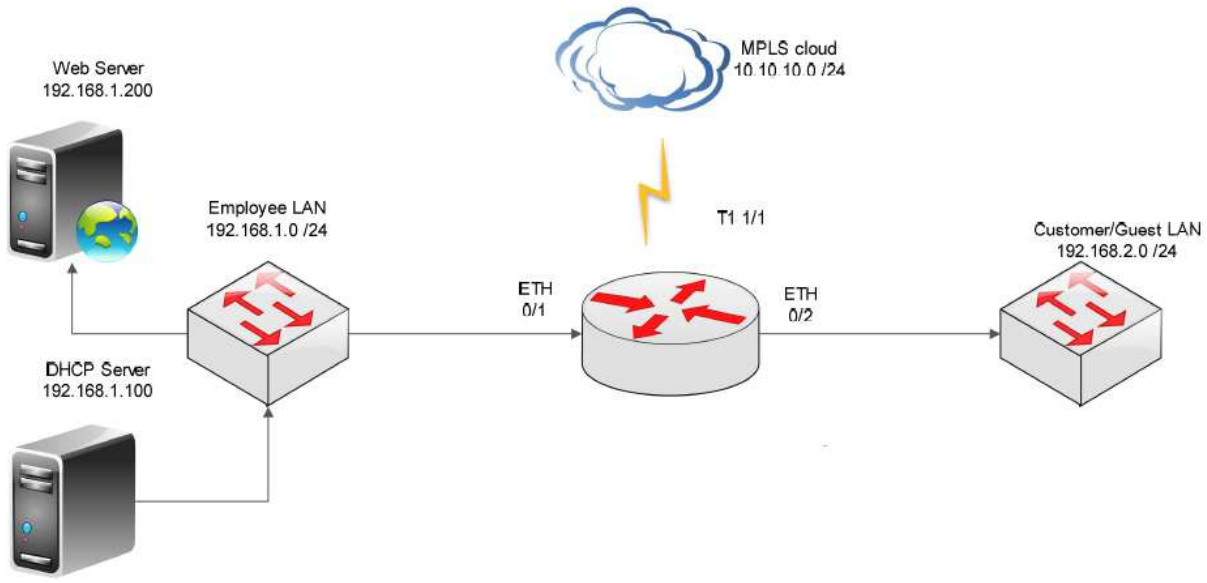
```
ip access-list extended BLOCK-SSH
  deny tcp any host 10.10.10.1 eq ssh
  permit ip any any
```

To make it an access group, it will need to be applied to the interface in the inbound direction, because it is blocking inbound SSH connections:

```
interface eth 0/1
  ip access-group BLOCK-SSH in
```

Example 2: Blocking/Allowing Traffic with Multiple Network Subnets

A company has the following network:



The network administrator is trying to secure the servers and interfaces on the network, but would like to do so without enabling the firewall on the AOS unit. The network requirements are below:

Customer/Guest LAN

1. Permit access to the Web server and the DHCP server on the Employee LAN.
2. Deny access to the remaining Employee LAN and the AOS unit.
3. Deny access to the MPLS network.

Employee LAN

1. Permit access to the MPLS network.
2. Permit access to the AOS device via SSH.
3. Deny access to the Customer Network unless the source of the traffic is the DHCP server or Web server.

MPLS Network

1. Permit access to the Employee LAN.
2. Deny access to the Customer LAN.

3. Permit access to the AOS device via SSH only, if the source IP address is 10.10.10.100.

Below are the current configurations of the AOS unit's three interfaces:

```
interface eth 0/1
 ip address 192.168.1.1 255.255.255.0
 no shutdown
!
interface eth 0/2
 ip address 192.168.2.1 255.255.255.0
 no shutdown
!
interface t1 1/1
 tdm-group 1  timeslots 1-24 speed 64
 no shutdown
!
interface ppp 1
 ip address 10.10.10.1 255.255.255.0
 cross-connect 1 t1 1/1 1 ppp 1
 no shutdown
```

To start, build an ACL for each set of criteria and then apply it to the corresponding interface:

<u>Customer/Guest LAN</u>

1. Permit access to the Web server and the DHCP server on the Employee LAN.
   permit ip any host 192.168.1.100
   permit ip any host 192.168.1.200
2. Deny access to the remaining Employee LAN and the router itself.
   Implicit deny at the bottom of the ACL
3. Deny access to the MPLS network.
   Implicit deny at the bottom of the ACL

```
ip access-list extended CUSTOMER-LAN
 permit ip any host 192.168.1.100
 permit ip any host 192.168.1.200
```

When traffic reaches the ACL, it will only be allowed to pass if it is destined for either the IP address of 192.168.1.100 or 192.168.1.200. If not, it will be matched by the implicit deny at the bottom of the ACL, and the traffic will be blocked.

To make this ACL an Access-group, it will be applied to the Ethernet 0/2 interface in the inbound direction, which is closest to the source of the traffic being blocked (the customer traffic):

interface eth 0/2
ip access-group CUSTOMER-LAN in

Employee LAN

1. Permit access to the MPLS network.
   permit ip any 10.10.10.0 0.0.0.255
2. Permit access to the router via SSH.
   permit tcp any host 192.168.1.1 eq ssh
3. Deny access to the Customer Network unless the source IP address is the DHCP server or Web server.
   This rule can be accomplished in several ways. Since there is an implicit deny at the bottom of the ACL, the simplest way is to allow traffic sourced from the two servers, and let the implicit deny block everything else

   permit ip host 192.168.1.100 192.168.2.0 0.0.0.255
   permit ip host 192.168.1.200 192.168.2.0 0.0.0.255

ip access-list extended EMPLOYEE-LAN
permit ip any 10.10.10.0 0.0.0.255
permit tcp any host 192.168.1.1 eq ssh
permit ip host 192.168.1.100 192.168.2.0 0.0.0.255
permit ip host 192.168.1.200 192.168.2.0 0.0.0.255

Now the ACL will be applied to the Ethernet 0/1 interface, in the inbound direction, since it is closest to the source of the traffic being affected (employee LAN traffic):

interface eth 0/1
ip access-group EMPLOYEE-LAN in

MPLS Network

1. Permit access to the Employee LAN.
   permit ip any 192.168.1.0 0.0.0.255
2. Deny access to the Customer LAN.
   Implicit deny at the bottom of the ACL
3. Permit access to the AOS device via SSH, only if the source is 10.10.10.100.
   Here it will again be more efficient to allow through the 10.10.10.100 SSH traffic and let the implicit deny block everything else.

   permit tcp host 10.10.10.100 any eq ssh

This makes the final ACL as shown below:

ip access-list extended MPLS-NETWORK
permit ip any 192.168.1.0 0.0.0.255
permit tcp host 10.10.10.100 any eq ssh

Now the ACL will be applied to the PPP 1 interface, in the inbound direction, since it is closest to the source of the traffic being affected (MPLS traffic):

interface ppp 1
ip access-group MPLS-NETWORK

Example 3: Using an Outbound Access-group

In this example, a network administrator would like to block SSH access to a server that is connected to the core AOS device's Ethernet 0/1 interface, regardless of the traffic's source IP address. Ethernet 0/1's configuration is below:

interface eth 0/1
ip address 10.10.10.1 255.255.255.0
no shutdown

The server's IP address is 10.10.10.2. The core AOS device has 25 VLAN interfaces on it, any of which could generate SSH traffic, which would exit the device on the Ethernet 0/1 interface destined to that server. In this case, the configuration will be more efficient for the network administrator to block the traffic outbound on the eth 0/1 interface, instead of blocking it inbound on all of the different interfaces that could try and access the server. If the access-group was located on all 25 vlans, it would also require the AOS device to check all traffic that

comes in on those interfaces, no matter the destination, to make sure it is not SSH traffic destined for that server.

The network administrator should create the following ACL:

```
ip access-list extended BLOCK-SERVER-SSH
 deny tcp any host 10.10.10.2 eq ssh
 permit ip any any
```

The **permit ip any any** at the bottom is so that the implicit deny of the ACL will not catch all other traffic. Now that the ACL has been made, the network administrator should apply the ACL as an Access-group to the Ethernet 0/1 interface in the outbound direction:

```
interface eth 0/1
 ip access-group BLOCK-SERVER-SSH out
```

Troubleshooting Access-groups

When verifying traffic is being blocked or allowed properly by Access-groups, the easiest way to confirm this is to check to see if there are matches for the ACL that the Access-group references. For example, consider the following configuration:

```
interface eth 0/1
 ip address 10.10.10.1 255.255.255.0
 ip access-group BLOCK-PING in
 no shutdown
!
ip access-list extended BLOCK-PING
 deny   icmp any  any
 permit ip any  any
```

In this case, running the command **show ip access-list** will allow the user to see which rules have been matched, and how many times they have been matched, to indicate whether the Access-group is working:

```
Extended IP access list BLOCK-PING
  deny   icmp any  any     (66 matches)
  permit ip any  any     (185 matches)
```

Issuing the command several times in succession will allow the user to check whether the matches are increasing, as the matched traffic is hitting the router. The ACL matches can be cleared by issuing the **clear ip access-list** command.