SUNY Polytechnic Institute

NCS 450

Network Security

Lab 5: Basic Access Control Lists

Author:
Jess Yanarella
Tony Perez

Professor:
Ronny Bull

March 2015

1 Introduction:

In this lab, we are setting up and configuring Access Control lists on our router. We want only our pod PC to have access to the router and block everything else. We don't want any outside users trying to connect to our router and being able to make changes. We also need to set up access to the ftp server so we can get our saved configuration files.

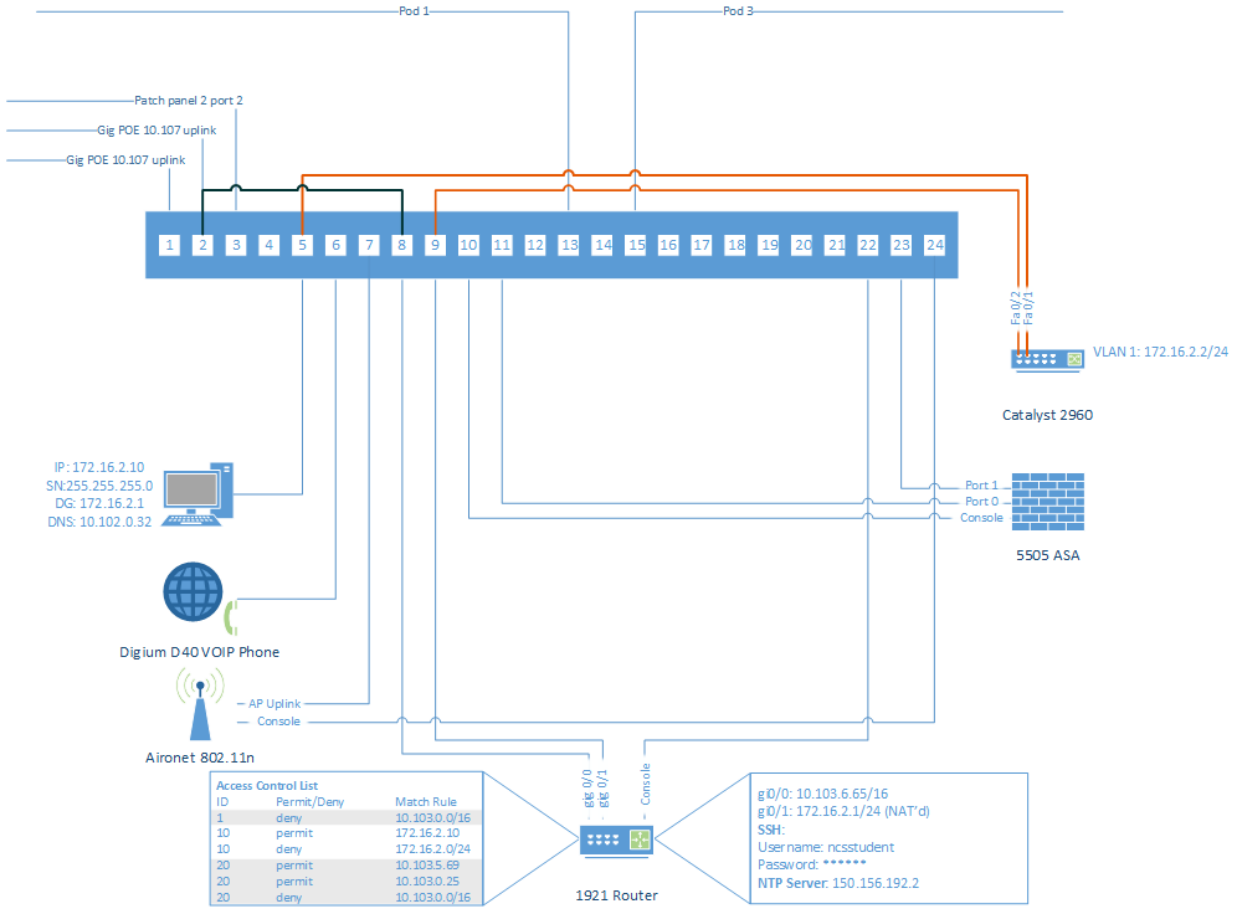

Figure 1: Network Diagram with ACL Implemented

2 SSH Access:

After we setup our switch and router with the configuration from our previous labs using the TFTP server, we must test SSH. Our POD PC needs to be configured so it is on the same network as the router. It is the same process as other labs by following simple commands to change the eth0 interface, which changes the IP address. Also, for the connection to work with the router, the default gateway must be configured.

```
# sudo ifconfig eth0 172.16.2.10 netmask 255.255.255.0
# sudo route add default gw 172.16.2.1
```

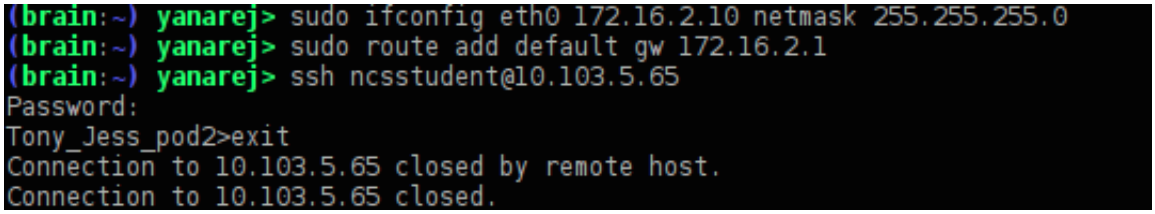

Figure 2: Pod PC SSH

Since our PC can ssh to our router, we have to check and make sure neighboring pods can connect to it as well. We asked other students to try and ssh to our router and it worked for them. Now we can go ahead and begin to create our access control list.

3 Access Control List:

We are creating an access control list to allow only our pod PC to be able to access our router. We want to prevent any other users in the class from being able to SSH to it. To create an access list, we first must enter the configuration terminal mode. Once in configuration terminal mode, we can create an access list for allowing or denying using IP addresses, subnets, netowrks, and even use ports. We must use the proper matching parameters. We have three access lists on different numbers 1, 10, and 20. The first list denies all the pod PC's access to our router

```
(config)# access-list 1 deny 10.103.0.0 0.0.255.255
```

The next list, list 10, permits the address 172.16.2.10 to access our machine. This address is our pod PC and allows us access. To deny other PC's access, we must deny any other address on the 172.16.2.0 network.

```
(config)# access-list 10 permit 172.16.2.10
(config)# access-list 10 deny 172.16.2.0 0.0.0.255
```

The list 20 allows our pod PC access with it's original IP address before we made any changes to it. We must also deny other PC's access on that network.

```
(config)# access-list 20 permit 10.103.5.69
(config)# access-list 20 10.103.0.0 0.0.255.255
```

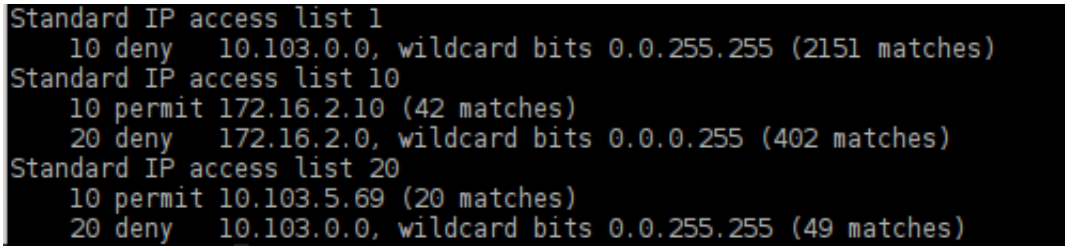

Figure 3: Access Lists

4 Testing SSH with ACL:

Now that we created our access lists, we must test to see that they are properly working. Since we allowed our pod PC access, we should be able to have access which we do. We tested an ssh access from another pod and they were not able to connect.

```
(brain:~) yanarej> ssh ncsstudent@10.103.5.65
ssh: connect to host 10.103.5.65 port 22: No route to host
(brain:~) yanarej> ssh ncsstudent@10.103.5.65
Password:
Tony Jess pod2>
```

Figure 4: SSH Test Successful

Next, we went ahead and changed our pod PC IP address to something different to see if we could have access. We allowed 172.16.2.10 in our access list so we changed it to .11 for test purposes. After the changes were made, we tried sshing and no connection was made. This is good because it means our ACL is working.

```
(brain:~) yanarej> sudo ifconfig eth0 172.16.2.11 netmask 255.255.255.0
(brain:~) yanarej> sudo route add default gw 172.16.2.1
(brain:~) yanarej> ssh ncsstudent@10.103.5.65
ssh: connect to host 10.103.5.65 port 22: No route to host
```

Figure 5: SSH Denied

5 Interfaces:

Interface gi0/0 is coonfigured with the inbound access list 20.

```
Tony_Jess_pod2#sh ip int gi0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 10.103.5.65/16
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is 20
```

Figure 6: Interface gi0/0

Interface gi0/1 is configured with the inbound access list 10.

```
Tony_Jess_pod2#sh ip int gi0/1
GigabitEthernet0/1 is up, line protocol is up
  Internet address is 172.16.2.1/24
  Broadcast address is 255.255.255.255
  Address determined by configuration file
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is 10
```

Figure 7: Interface gi0/1

6 TFTP Server:

We ran into an issue at the end of the lab when trying to save our new configuration. We weren't able to access the server because it was being blocked in our access control list. We made the appropriate changes to the list to permit the server's IP and after we allowed access.

```
Tony_Jess_pod2(config)#access-list 20 permit 10.103.5.69
Tony_Jess_pod2(config)#access-list 20 permit 10.103.0.25
Tony_Jess_pod2(config)#access-list 20 deny 10.103.0.0 0.0.255.255
```

Figure 8: Allow TFTP Server