

Shaun Lim

## Lab 8.2 – Securing Device Remote Access Using ACLs

### Topology



### Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	192.168.0.1	255.255.255.0	N/A
	G0/1	192.168.1.1	255.255.255.0	N/A
S1	VLAN 1	192.168.1.2	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.0.3	255.255.255.0	192.168.0.1

### Objectives

**Part 1: Configure Basic Device Settings**

**Part 2: Configure and Apply the Access Control List on R1**

**Part 3: Verify the Access Control List Using Telnet**

**Part 4: Challenge - Configure and Apply the Access Control List on S1**

### Background / Scenario

It is a good practice to restrict access to the router management interfaces, such as the console and vty lines. An access control list (ACL) can be used to allow access for specific IP addresses, ensuring that only the administrator PCs have permission to telnet or SSH into the router.

**Note:** In the Cisco device outputs, ACL is abbreviated as access-list.

In this lab, you will create and apply a named standard ACL to restrict remote access to the router vty lines.

After the ACL has been created and applied, you will test and verify the ACL by accessing the router from different IP addresses using Telnet.

This lab will provide the commands necessary for creating and applying the ACL.

**Note:** The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of the lab for the correct interface identifiers.

**Note:** Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

### Required Resources

- 1 Router (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 1 Switch (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)
- 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

**Note:** The Gigabit Ethernet interfaces on Cisco 1941 routers are autosensing and an Ethernet straight-through cable may be used between the router and PC-B. If using another model Cisco router, it may be necessary to use an Ethernet crossover cable.

## Part 1: Configure Basic Device Settings

In Part 1, you will set up the network topology and configure the interface IP addresses, device access, and passwords on the router.

**Step 1: Cable the network as shown in the topology diagram.**

**Step 2: Configure the PC-A and PC-B network settings according to the Addressing Table.**

**Step 3: Initialize and reload the router and switch.**

- a. Console into the router and enter global configuration mode.
- b. Copy the following basic configuration and paste it to the running-configuration on the router.

```
no ip domain-lookup
hostname R1
service password-encryption
enable secret class
banner motd #
Unauthorized access is strictly prohibited. #
Line con 0
password cisco
login
logging synchronous
line vty 0 4
password cisco
login
```

- c. Configure IP addresses on the interfaces listed in the Addressing Table.
- d. Save the running configuration to the startup configuration file.
- e. Console into the switch and enter global configuration mode.
- f. Copy the following basic configuration and paste it to the running-configuration on the switch.

```
no ip domain-lookup
hostname S1
```

```
service password-encryption
enable secret class
banner motd #
Unauthorized access is strictly prohibited. #
Line con 0
password cisco
login
logging synchronous
line vty 0 15
password cisco
login
exit
```

- g. Configure IP address on VLAN1 interface listed in the Addressing Table.
- h. Configure the default gateway for the switch.
- i. Save the running configuration to the startup configuration file.

## Part 2: Configure and Apply the Access Control List on R1

ACLs may also be used to control host access to the remote management services of a device. In Part 4, you will configure a named standard ACL and apply it to the router virtual terminal lines to restrict remote access to the router.

### Step 1: Configure and apply a standard named ACL.

- a. On R1, create a named standard ACL that conforms to the following policy:  
Permit traffic from administrator PC (PC-A) at 192.168.1.3 and an additional range of reserved addresses 192.168.1.4 to 192.168.1.7. As with best practices in formulating ACLs, the last statement must be a **deny all**. Name this ACL as ADMIN-MGT

Provide the commands needed to create this ACL:

```
ip access-list standard ADMIN-MGT
permit 192.168.1.3 0.0.0.0
permit 192.168.1.4 0.0.0.3
deny any
```

- b. Now that the named ACL is created, apply it to the vty lines.

```
R1(config)# line vty 0 15
R1(config-line)# access-class ADMIN-MGT in
R1(config-line)# exit
```

## Part 3: Verify the Access Control List Using Telnet

In Part 3, you will use Telnet to access the router, verifying that the named ACL is functioning correctly.

**Note:** SSH is more secure than Telnet; however, SSH requires that the network device be configured to accept SSH connections. Telnet is used with this lab for convenience.

- a. Open a command prompt on PC-A and verify that you can communicate with the router by issuing a **ping** command.

```
C:\Users\user1> ping 192.168.1.1
```

```
Pinging 192.168.1.1 with 32 bytes of data:
```

```
Reply from 192.168.1.1: bytes=32 time=5ms TTL=64
```

```
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
```

```
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
```

```
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
```

```
Ping statistics for 192.168.1.1:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 1ms, Maximum = 5ms, Average = 2ms
```

```
C:\Users\user1>
```

- b. Using the PuTTY on PC-A, telnet into the router. Enter the login and then the enable passwords. You should be successfully logged in, see the banner message, and receive an R1 router command prompt.

```
Unauthorized access is prohibited!
```

```
User Access Verification
```

```
Password:
```

```
R1>enable
```

```
Password:
```

```
R1#
```

Was the Telnet successful?	Yes
----------------------------	-----

- c. Type **exit** at the command prompt and press Enter to exit the Telnet session.
- d. Change your IP address to test if the named ACL blocks non-permitted IP addresses. Change the IPv4 address to 192.168.1.100 on PC-A.
- e. Attempt to telnet into R1 at 192.168.1.1 again. Was the Telnet session successful?

Was the Telnet successful?	No
What was the message received?	The current session has closed. Do you want to reestablish a new session with the previous session options?

- f. Change the IP address on PC-A to test if the named ACL permits a host with an IP address from the 192.168.1.4 to 192.168.1.7 range to telnet into the router. After changing the IP address on PC-A, open a Windows command prompt and attempt to telnet into router R1.

Was the Telnet session successful?

Was the Telnet successful?	Yes
----------------------------	-----

- g. From privileged EXEC mode on R1, type the **show ip access-lists** command and press Enter. From the command output, notice how the Cisco IOS automatically assigns line numbers to the ACL ACEs in increments of 10 and shows the number of times each permit ACE has been successfully matched (in parenthesis).

```
R1# show ip access-lists
```

```
Standard IP access list ADMIN-MGT
 10 permit 192.168.1.3 (2 matches)
 20 permit 192.168.1.4, wildcard bits 0.0.0.3 (2 matches)
```

Because two successful Telnet connections to the router were established, and each Telnet session was initiated from an IP address that matches one of the permit ACEs, there are matches for each permit ACE.

- h. On R1, enter into global configuration mode.
- i. Enter into access-list configuration mode for the ADMIN-MGT named access list and add a **deny any** ACE to the end of the access list.

```
R1(config)# ip access-list standard ADMIN-MGT
R1(config-std-nacl)# deny any
R1(config-std-nacl)# exit
```

**Note:** Because there is an implicit **deny any** ACE at the end of all ACLs, adding an explicit **deny any** ACE is unnecessary. However, the explicit deny any at the end of the ACLs can still be useful to the network administrator to log or simply know how many times the **deny any** access-list ACE was matched.

- j. Try to telnet from PC-B to R1. This creates a match to the **deny any** ACE in the ADMIN-MGT named access list.
- k. From privileged EXEC mode, type **show ip access-lists** command and press Enter. You should now see multiple matches to the **deny any** ACE.

```
R1# show ip access-lists
Standard IP access list ADMIN-MGT
 10 permit 192.168.1.3 (2 matches)
 20 permit 192.168.1.4, wildcard bits 0.0.0.3 (2 matches)
 30 deny any (3 matches)
```

The failed Telnet connection produces more matches to the explicit deny ACE than a successful one. Why do you think this happens?

This happens because Telnet attempts to reconnect to the device to ensure that it is most likely not just a dropped packet.

## Part 4: Challenge - Configure and Apply the Access Control List on S1

### Step 1: Configure and apply a standard named ACL for the vty lines on S1.

Without referring back to the R1 configuration commands, try to configure the ACL on S1, allowing only the PC-A IP address.

Provide the commands needed to create and apply this ACL:

```
ip access-list standard ADMIN-MGT
permit 192.168.1.3 0.0.0.0
deny any

line vty 0 15
access-group ADMIN-MGT in
```

### Step 2: Test the vty ACL on S1.

Telnet from each of the PCs to verify that the vty ACL is working properly. You should be able to telnet to S1 from PC-A, but not from PC-B.

### Reflection

1. How does an ACL applied to a device vty line interface improve the security of remote management?

Applying ACLs to a device's vty line interface enhances security of remote management by controlling access, preventing unauthorized entry, mitigating attacks, providing detailed logging, and facilitating compliance with security standards.

2. Why does it make sense to apply an ACL to vty lines instead of specific interfaces?

Applying ACLs to vty lines offers a more centralized, consistent, and scalable approach to managing remote access control, simplifying configuration and maintenance in complex network environments compared to applying ACLs on specific interfaces.

### Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
<b>Note:</b> To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				