Assignment 11
Configuring Security and Access-Lists on Cisco Devices and Simulating Infrastructure Attacks

Siddhartha Pandey (s316620)

November 11, 2016

Contents

Part I
Introduction

The purpose of this assignment is to learn how to secure Cisco devices and how to configure an access list on the routers, which is basically a firewall. To complete the assignment you need to have Cisco Packet Tracer installed on your system.

Part II
Methods and Materials

1 Security Measures

a Deploying the routers and the switch

We use the 1841 model when deploying the router and the model 2960-24TT for the switch. We deploy both the routers and the switch in Packet Tracer.

b Configuring Passwords

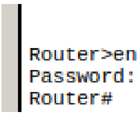
After deploying the router and the switch, we start configuration for the router. We use the **enable** command to enter the privileged EXEC mode, then the **configure terminal** command to enter the configuration mode. We add security to the router by configuring a password. We use the command **enable secret <*your password*>** to enable the password, where *your password* is replaced with the actual password. There is two ways to configure the password to access the router's priveleged EXEC mode. We can either use the **enable secret** or the **enable password** command. The *enable secret* takes precedence over the *enable password* command.

i *enable password*

The *enable secret* command sets a password for the various privilege levels. We use the command **enable password <*password*>** to set the password. This command does not encrypt the password saved in the *running-config* file.

ii *enable secret*

The *enable secret* command establishes a password for the privilege levels. The difference to the *enable password* is that this command encrypts the password stored in the *running-config* file providing better security.



```
Router>en
Password:
Router#
```

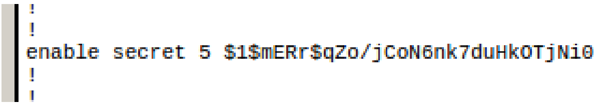
Figure 1: Entering the privelege EXEC mode with password enable with **enable secret** command.

c Observing the changes with password enabled

After we issue the **enable secret** command we have to enter the password when going from the user EXEC mode to the priveleged EXEC mode, we can this in Figure 1.

d Storing password in *running-config*

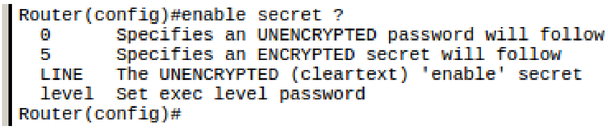
We can see the *running-config* using the **show running-config** command in the CLI of the router. We can observe as seen in Figure 2 that the password is encrypted.



```
!
!
enable secret 5 $1$mERr$qZo/jCoN6nk7duHkOTjNi0
!
!
```

Figure 2: The *running-config* in router, storing the password after **enable secret** command.

The 5 that follows after *enable secret* implies that the following is a hash for the password, encrypted using MD5 algorithm. We can see this in Figure 3, there is an encrypted hash of the password following the 5.

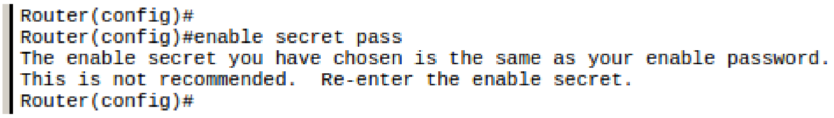


```
Router(config)#enable secret ?
  0      Specifies an UNENCRYPTED password will follow
  5      Specifies an ENCRYPTED secret will follow
  LINE   The UNENCRYPTED (cleartext) 'enable' secret
  level  Set exec level password
Router(config)#
```

Figure 3: Checking *enable secret* options

i Extra

We can use both the **enable secret** and the **enable password** command at the same time. The router uses the encrypted password when both are present. When using the same password for both the encrypted and the unencrypted password, the CLI warns us to use different but still allows it as seen in Figure 4.



```
Router(config)#
Router(config)#enable secret pass
The enable secret you have chosen is the same as your enable password.
This is not recommended.  Re-enter the enable secret.
Router(config)#
```

Figure 4: Using the same password for both **enable secret** and **enable password**

e Enable password for console access

Then we will also configure a password for the console access. We use the commands **line con 0** in the configuration mode. This lets us configure the primary terminal line.

i Set password

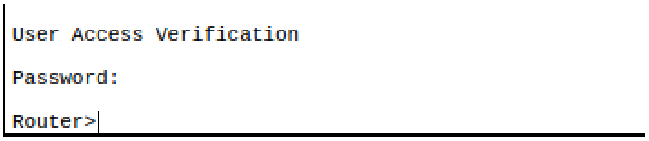
Then we use the **password** command to set the password.

ii Enable login for console access

We use the **login** command to enable password checking.

iii Loggin prompt

After setting the password for console access we get prompted to enter the password when we open the CLI, we need to enter the password even to enter the user EXEC mode as can be seen in Figure 5.



```
User Access Verification

Password:

Router>
```

Figure 5: Login prompt for access to the console.

f Console Access Password in *running-conf*

The password set for console access is not encrypted when we check the *running-configuration* using **show running-conf**, the password is stored in plain text.

g configuring password for telnet

We enter the telnet configuration mode using the **line vty 0 4** command in the global configuration mode.

We then set the password with the **password** command and enable login using the **login** command.

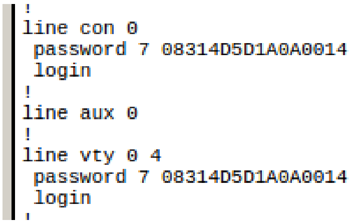
Checking the *running-configuration* we can see that the passwords for the *console access* and the *terminal access* are stored in plain text.

h Encrypt all passwords

As many of the passwords are stored in plain text we need to increase the security of the plain-text password. We use the **service password-encryption** command in the global configuration mode to encrypt all the unencrypted passwords.

i *running-conf* after encrypting password

We can see in Figure 6 that all the password have been encrypted. We can see that there is a number, 7, after password. This specifies that this was encryption Type 7.



```
!
line con 0
 password 7 08314D5D1A0A0014
 login
!
line aux 0
!
line vty 0 4
 password 7 08314D5D1A0A0014
 login
!
```

Figure 6: The encrypted password after running **service password-encryption**

i Type 5 and Type 7 encryptions

As seen in Figure 6 all the passwords that were encrypted using the **service password-encryption** command was encrypted with encryption Type 7. Previously when running the **enable secret** command the passwords were encrypted using Type 5, as seen in Figure 2.

The Type 5 encryption is done using an MD5 algorithm to generate the password hash. Whereas the Type 7 encryption is done using a weak algorithm, it is just XORs the password.

j Securing the Switch

We follow the same procedures to secure the switch by adding password and encrypting them. We use the same password as in the previous assignment.

- **enable secret** Add password for accessig the privileged EXEC mode.
- **line con 0** Configure console access

  password Set password for console access

  login Enable password login for console access

- **line vty 0 4** Configure telnet access

  password Set password for telnet access

  login Enable password login for telnet access

- **service password-encryption** Encrypt all pasword in the *runnnig-config*

2 Configuring Access lists

a Setting up the topology

We use 4 like the router previously configured and connect them to the switch that was previously configured. Our topology in Packet Tracer looks like Figure 7.

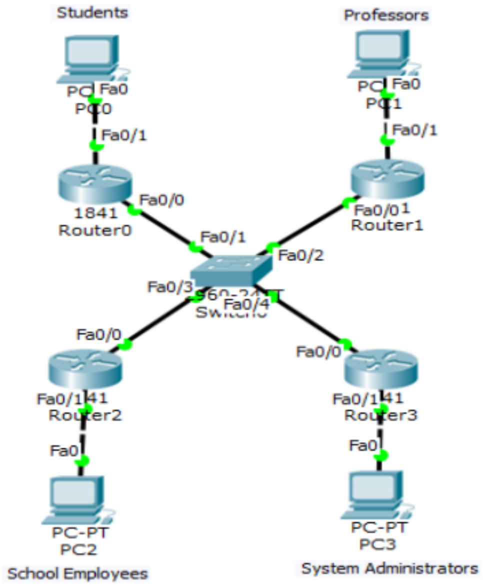


Figure 7: Topology in Packet Tracer

b Configuring the devices with IP addresses

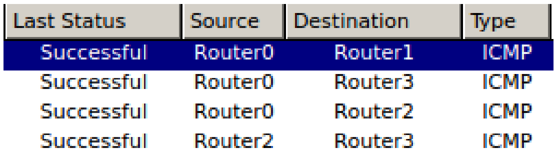
Now that we have a topology in Packet Tracer, we will configure the all the interfaces with an IP address as seen in Table 1. After all the interfaces have been assigned an IP address we will make sure that the interfaces are *up*. We will also configure the default gateway for the hosts to be the interface of the router they are each connected to respectively.

Device Name	Interface	IP Address
Router0	Fa0/0	10.0.0.1/24
Router 1	Fa0/0	10.0.0.2/24
Router 2	Fa0/0	10.0.0.3/24
Router 3	Fa0/0	10.0.0.4/24
Router0	Fa0/1	192.168.0.1/24
Router1	Fa0/1	192.168.1.1/24
Router2	Fa0/1	192.168.2.1/24
Router3	Fa0/1	192.168.3.1/24
PC0	Fa0/0	192.168.0.2/24
PC1	Fa0	192.168.1.2/24
PC2	Fa0	192.168.2.2/24
PC3	Fa0	192.168.3.2/24

Table 1: Interface IP addresses for hosts and routers.

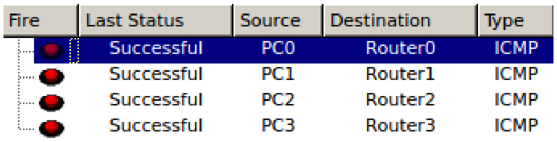
c Checking connectivity

We check the connectivity of the network to verify that everything is configured properly. The routers in the 10.0.0.0/24 network should ping each other as seen in Figure 8, and the PC should be able to ping its own router 9.



Last Status	Source	Destination	Type
Successful	Router0	Router1	ICMP
Successful	Router0	Router3	ICMP
Successful	Router0	Router2	ICMP
Successful	Router2	Router3	ICMP

Figure 8: Checking connectivity between routers.



Fire	Last Status	Source	Destination	Type
●	Successful	PC0	Router0	ICMP
●	Successful	PC1	Router1	ICMP
●	Successful	PC2	Router2	ICMP
●	Successful	PC3	Router3	ICMP

Figure 9: Checking connectivity between the hosts and the routers.

d Configuring Routing with RIPv2

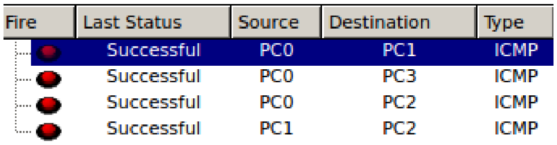
For full connectivity in the network we will configure the routers with a routing protocol. Here, the RIPv2 routing protocol will be used. We configure the routing protocol by issuing the following commands:

- **router rip** In the configuration mode.

version 2 Specifies the routing protocol as RIP version 2

network x.x.x.x This command is issued for all the networks that are connected to the router, where x.x.x.x is one specific network.

We use this method to configure all the routers in the topology. Then check that there is connectivity between all the hosts. We can see in Figure 10 that there is full connectivity between the hosts.



Fire	Last Status	Source	Destination	Type
●	Successful	PC0	PC1	ICMP
●	Successful	PC0	PC3	ICMP
●	Successful	PC0	PC2	ICMP
●	Successful	PC1	PC2	ICMP

Figure 10: Checking connectivity between hosts

e Configuring ACL

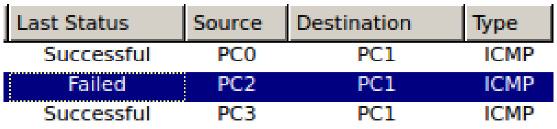
Now we wil configure ACLs in our routers. Now we want to allow all traffic originating from the students and system administrator network to reach the professor's network, but we do not want to let the school employees access the professor's network. To do this we deploy an access-list on Router1 on all the incoming traffic on interface Fa0/0. We do this by create two rules with standard ACL that allows the traffic of networks 192.168.0.0/24 and 192.168.3.0/24. We do this using the following commands, in the global configuration mode:

- **access-list 1 permit 192.168.0.0 0.0.0.255**
- **access-list 1 permit 192.168.3.0 0.0.0.255**

Then we add this two rules to the interface Fa0/0 by using the command:

- **ip access-group 1 in**

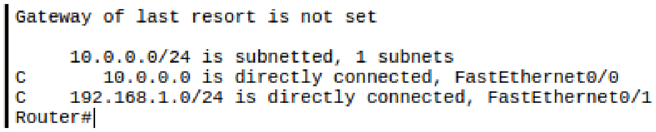
This specifies that it allows incoming traffic of access-group 1 (defined above) only at interface Fa0/0 at Router1. When checking connectivity with *ping* we can see that this will let PC0 and PC3 reach PC1, but not PC2 as seen in Figure 11.



Last Status	Source	Destination	Type
Successful	PC0	PC1	ICMP
Failed	PC2	PC1	ICMP
Successful	PC3	PC1	ICMP

Figure 11: Checking connectivity from PC0, PC2 and PC3 to PC1

We can not ping even from any of the other routers. After letting some time pass there will be no connectivity between any of the hosts as the routing table at Router1 will not be updated as only the student and system administrator's networks traffic is recieved by the interface Fa0/0 at Router1, we can see that in Figure 12. An alternative would be to configure the access-list groups at Fa0/1 and restrict incoming outgoing traffic. So the packets can be recieved by the router but would be droped before it went to PC1. The command to use would be : **ip access-group 1 out** at interface Fa0/1 in Router1. Another alternative would be to add the network 10.0.0.0/24 to access-list for inbound, though this will also allow the routers to reach the professors network (PC1).



```
Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/1
Router#
```

Figure 12: Routing table at Router1

f Configuring outbound ACL

Now configuring an outbound ACL that permits all traffic originating from the professors network. We make an access-list with number2 with the command:

- **access-list 1 permit 192.168.1.0 0.0.0.255**

And then add this access-list to the internal interface Fa0/1 at Router1. We do this using the command:

- **ip access-group 2 in**

I can ping from PC1 to PC0 and PC3, but only if I implement one of alternative methods, for restricting access to PC2, as mentioned above.

g Extended ACLs

Extended ACLs can specify some special reserved keywords, One one of the keyword is *any* which corressponds to 0.0.0.0/255.255.255.255, another is *host* which is same as typing "/0.0.0.0". The extended ACLs use higher numbers (100-199) whereas standard ACLs use lower numbers (1-99).

h Allowing only ICMP traffic

With extended ACLs we can restrict traffic to only ICMP traffic, or any other particular protocol, useing the command:

- **access-list 101 permit icmp any any**

And setting the rules to both the interfaces of a router.

i Allowign only a particular host external access

We can give access to only 1 PC from the school employee network to have access to students network by allowing traffic going only from that host to the students network. We make a rule that looks like:

- **access-list 101 permit ip host 192.168.2.2 192.168.0.0 0.0.0.255**