

1. Introduction

In this lab, you will practice basic concepts of routing and switching using Cisco devices. This lab also serves as an introduction to Cisco hardware and software, which is an example of real-life network devices that you will encounter in an enterprise environment.

To make this lab possible, you will use Packet Tracer. This free software by Cisco is a network simulation tool for Cisco devices. It will allow you to create and set up different hardware and topologies and test them in a simulated environment. For more information, look at the Resources section at the end.

2. Prerequisites

Packet Tracer is a cross-platform tool and can be used on different operating systems. For this lab, we will use our previous Ubuntu test environment. To download Packet Tracer, you first need to [create a free account](#) and follow the steps to download the *.deb* file. Then, you can install Packet Tracer and its dependencies using *apt install* command.

Note 1: If *apt install* fails to install Packet Tracer on the latest version of Ubuntu, try first installing the packages *dialog* and *libxcb-xinerama0-dev* and then install using the following command:

```
$ dpkg --ignore-depends=libgl1-mesa-glx --install ./filename.deb
```

(This problem will be hopefully fixed in the next version of Packet Tracer)

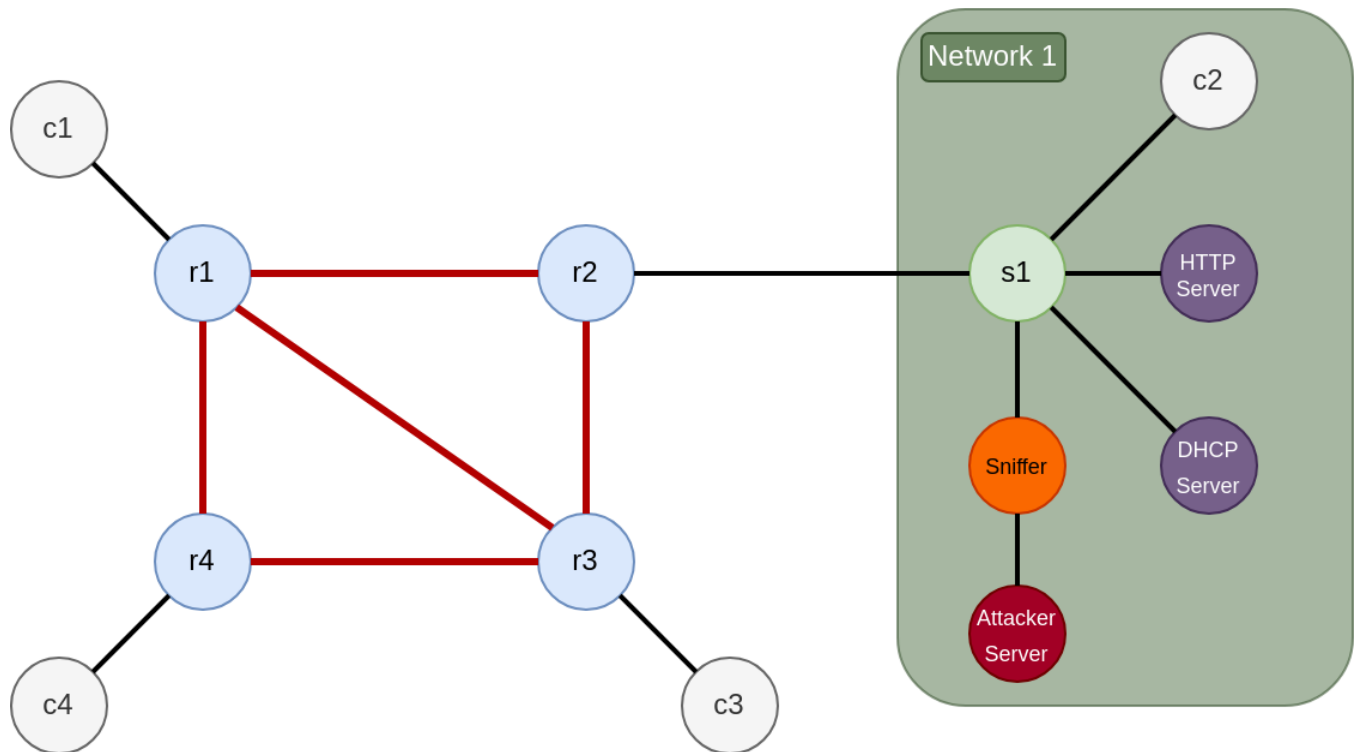
Note 2: the IP addresses for this lab are linked to your student ID. When you encounter the placeholder "XXX" in an IP address, replace it with the sum of the last four digits of your student ID, grouped in pairs. For example, if your student ID is 3612345678, XXX would be $56 + 78 = 134$.

IMPORTANT NOTE: Changes to Cisco devices configurations are by default only applied to the memory and are not permanent. That means upon restarting the device (or closing the Packet Tracer file), all unsaved settings are lost. You **MUST** copy the running-config to startup-config for the changes to be preserved during reboot. Your submitted Packet Tracer file must have all the settings applied to startup-config and work out of the box. Example of the required steps to copy running-config to startup-config:

```
r1>enable
r1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

3. Create the Network [30%]

Your first task is to create a small AS network inside the Packet Tracer. The required network is depicted in the following figure. It consists of four routers (r1-r4), one switch (s1), four clients (c1-c4), an HTTP server, a DHCP server, an attacker server, and a sniffer.



Design this network in Packet Tracer with the following requirements:

- All routers are of the model Cisco 2911
- The switch is Cisco 2960
- The clients are normal PCs
- All the links between routers (the red links) are fiber links
- The HTTP and DHCP servers are running HTTP and DHCP services respectively
- The attacker is running a server which has HTTP service enabled
- Adjust the hostnames of the routers, switch, and the PCs based on the given names in the diagram
- All IP addresses except the IP address of c2 are static. For c2, the IP address should be assigned by the DHCP server

The network must have the following subnets:

Network segment	Subnet	Network segment	Subnet
c1-r1	10.11.XXX.0/30	r1-r2	10.12.XXX.0/30
c3-r3	10.33.XXX.0/30	r1-r3	10.13.XXX.0/30
c4-r4	10.44.XXX.0/30	r1-r4	10.14.XXX.0/30
Network 1	10.22.XXX.0/24	r2-r3	10.23.XXX.0/30
		r3-r4	10.34.XXX.0/30

4. Connect the Network [40%]

For this task, we do the necessary preparation to connect the hosts through the AS network and check the connectivity and availability of the network. Your submission should include the working Packet Tracer file for this section, named “lab08_4.pkt”

a) You need to install the required forwarding tables. In this lab, you will not rely on routing protocols (e.g., OSPF) to build the forwarding tables. Instead, you will populate the four routers with *static routes* to forward packets on the shortest path for every host in the network (*in the future*, we may build multiple ASs and run BGP and OSPF to populate the forwarding tables). We assume that each link has a cost of 1. For instance, under normal operation of the network, a packet from c1 to c3 must go through $r1 \rightarrow r3$ (cost = 1) and not $r1 \rightarrow r2 \rightarrow r3$ (cost = 2). To build these tables, you need to know the **IP address of every interface** in the network. For each router, your report should include all the relevant routing commands that are used.

b) After populating the routes, you need to run the following *tracert* command and include the screenshots of the results in your report to show the correct connectivity:

- (1) c1 to c3
- (2) c2 to c4
- (3) c4 to c2
- (4) HTTP Server to c3

c) Now, we introduce link outages to the AS and observe the results. For each given scenario, turn off the required link. Then, include the screenshots of *ping* and *tracert* to/from involving clients:

Note: Do not forget to turn the links back online again after each test

- r1-r2 link is down
 - c1↔c2 traffic should now automatically re-route to r1↔r3↔r2
 - Include *ping* and *tracert* screenshots to/from c1/c2
- r1-r3 link is down
 - c3↔c1 traffic should now automatically re-route to r3↔r4↔r1 **or** r3↔r2↔r1
 - Include *ping* and *tracert* screenshots to/from c3/c1
- r1-r2 **and** r3-r4 are down
 - c4↔HTTP Server traffic should now automatically re-route to r4↔r1↔r3↔r2
 - Include *ping* and *tracert* screenshots to/from c4/HTTP Server

5. Attack the Network [30%]

A rouge employee on the network, who has an active connection to the s1 switch, is trying to carry out an attack on the internal system. They have put a sniffer device on the link and connected their own server to it. The goal of the attacker is to redirect the HTTP Server traffic to their own server and carry out a phishing attack by providing an alternative webpage to the clients.

a) Your task is to provide a method to carry out this attack and show step by step procedure to reliably do so. You then need to further support your claim by including the relevant screenshots of the sniffer log and clients' web browser before and after the attack to show the attack is successful.

Hint: Think about what happens in Layer 2 and Layer 3, and how devices find each other and communicate with each other.

Note 1: The attacker has only full access to their own server as well as the sniffer, which is connected to it. However, they have no access to any other devices on the network and cannot influence the topology of the network either.

Note 2: We assume that the attacker knows the IP address of the real HTTP Server as well as the IP address of the immediate router (r2). However, in a real-world scenario, even this assumption is not required.

Note 3: To prove that you were able to provide a fake webpage to the client, you need to change the default index.html file of the HTTP service on the attacker's server to something different from the real index.html file on the HTTP Server.

b) When the attack is happening, run the *tracert* command on clients to trace the routes to the "HTTP Server." What difference do you see from when there is no attack? Explain why.

c) Change the network in a way to defeat such attacks on the network in the future. Explain why this change stops the attack. Then, support your claim by carrying out the same attack again and showing that it is not successful anymore. Your submission should include the working Packet Tracer file for this change, named “lab08_5c.pkt”

6. Submission

- You need to submit:
 - (1) The Cisco Packet Tracer files that you have created
 - (2) A detailed lab report
- The files should be compressed in a single (.zip) archive
- The Packet Tracer files should run without any errors
- No password should be set on any devices inside Packet Tracer (we need to be able to see and verify all applied settings).

7. Resources

[Cisco Packet Tracer Course](#)

[Cisco 2900 Series Software Configuration Guide](#)

[Cisco Commands Cheat Sheet](#)

Note: Not all Cisco IOS commands are supported by Packet Tracer