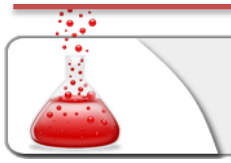
FIND THE SECRET SERVER

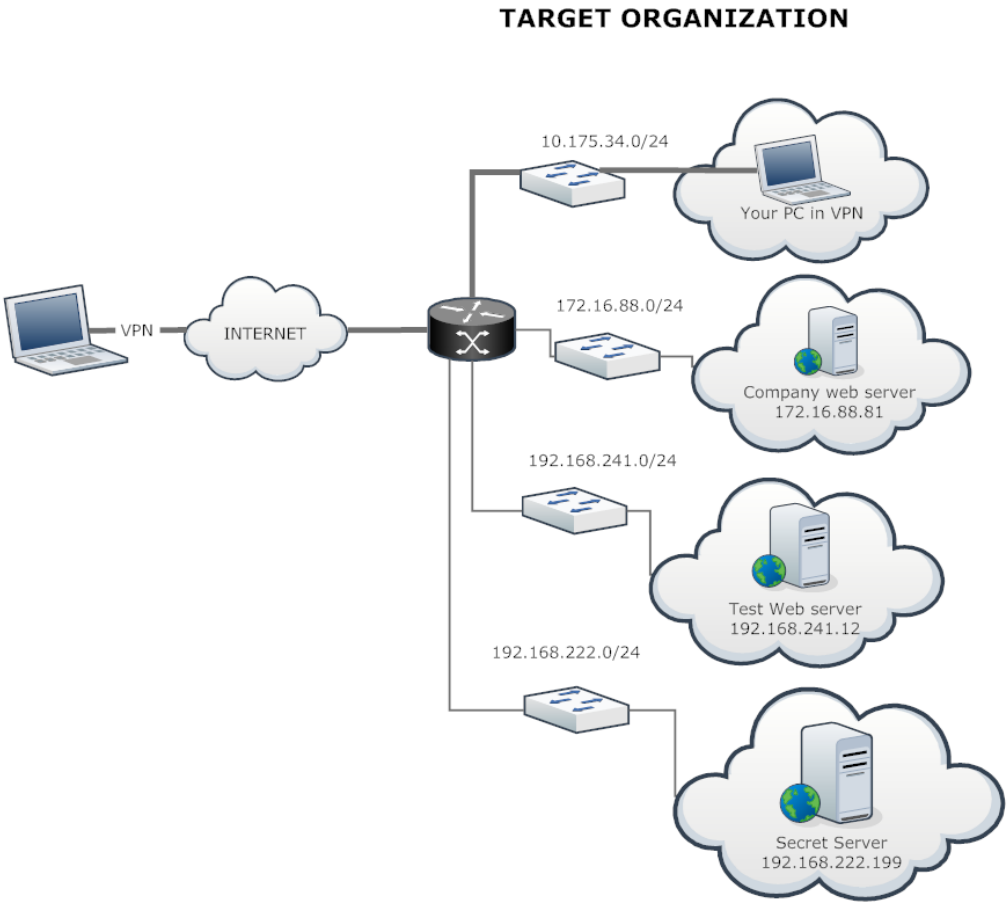


PRELIMINARY SKILLS | SECTION 1 MODULE 2 | LAB #2

LAB

1. Description

In this lab, you will learn how network routes work and how they can be manually added in order to reach different networks. The following diagram shows the network configuration of the lab:



As you can see, you are attached via VPN to the network 10.175.34.0/24 but there are also other three networks. In each network, there is a web server (you can access it by browsing its IP address with your web browser) with the following IP addresses: 172.16.88.81, 192.168.241.12 and 192.168.222.199.

2. Goal

The goal of the lab is to configure your VPN lab environment in order to reach all the hosts in the networks!

3. Tools

The best tool is, as usual, your **brain**. Then you may need:

- *OpenVPN client*
- *Web browser*

4. Steps

4.1. Check your current network configuration

Before connecting to the lab, check you current routes.

4.2. Connect to the lab and check your routes

Establish the VPN connection to the lab. If it's your first time in Hera Lab please refer to this manual: https://members.elearnsecurity.com/lab/manual

What differs from the previous output?

4.3. Visit the two web servers

There are two Web Servers at the following addresses: 172.16.88.81 and 192.168.241.12. Are you able to navigate them once you are connected to the lab?

4.4. Add a route manually

We know that there is another server at the address 192.168.222.199. Right now, we do not have any route set on our machine and we are not able to reach it. Try adding the correct route to that network and see if you can reach it.

SOLUTIONS

Please go ahead **ONLY** if you have **COMPLETED** the lab or you are stuck! Checking the solutions before actually trying the concepts and techniques you studied in the course, will dramatically reduce the benefits of a hands-on lab!

[This page was intentionally left blank]

5. Solutions steps

5.1. Check your current network configuration

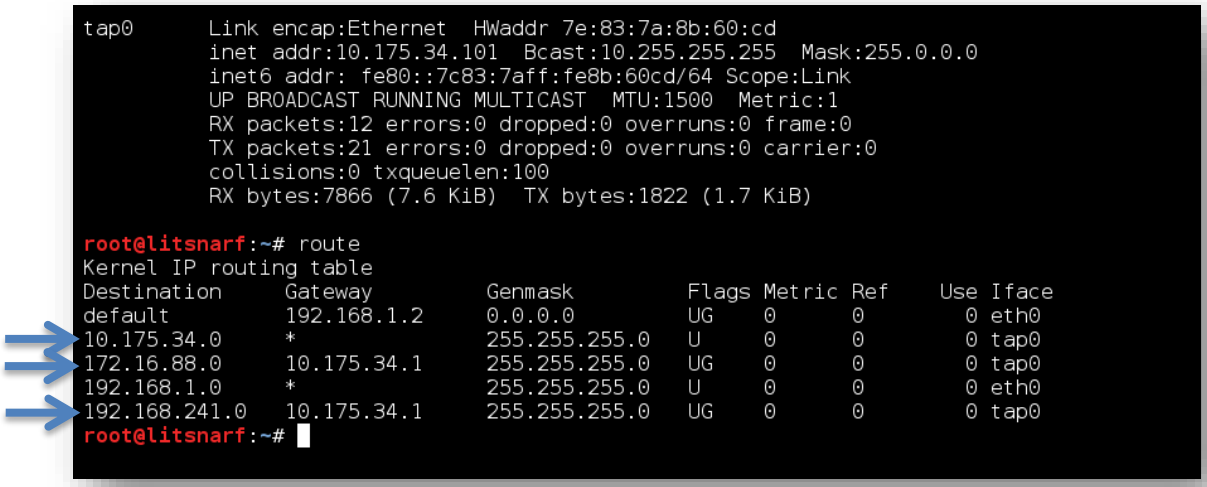
Before connecting to the lab, check your network configurations: interfaces and routes. Note that the following screenshot may differ from your output:

```
root@litsnarf:~/Desktop# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:29:ce:43
          inet addr:192.168.1.157  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe29:ce43/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:266253 errors:0 dropped:0 overruns:0 frame:0
          TX packets:289361 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:141725138 (135.1 MiB)  TX bytes:77222967 (73.6 MiB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2308808 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2308808 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:336255178 (320.6 MiB)  TX bytes:336255178 (320.6 MiB)

root@litsnarf:~/Desktop# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.1.2     0.0.0.0         UG    0      0        0 eth0
192.168.1.0     *               255.255.255.0   U     0      0        0 eth0
root@litsnarf:~/Desktop#
```

As we can see from the screenshot above, we have two interfaces: the loopback (lo) and the Ethernet interface (eth0). Moreover, in our example, we have few routes that determine what networks we can reach and how.

5.2. Connect to the lab and check your routes

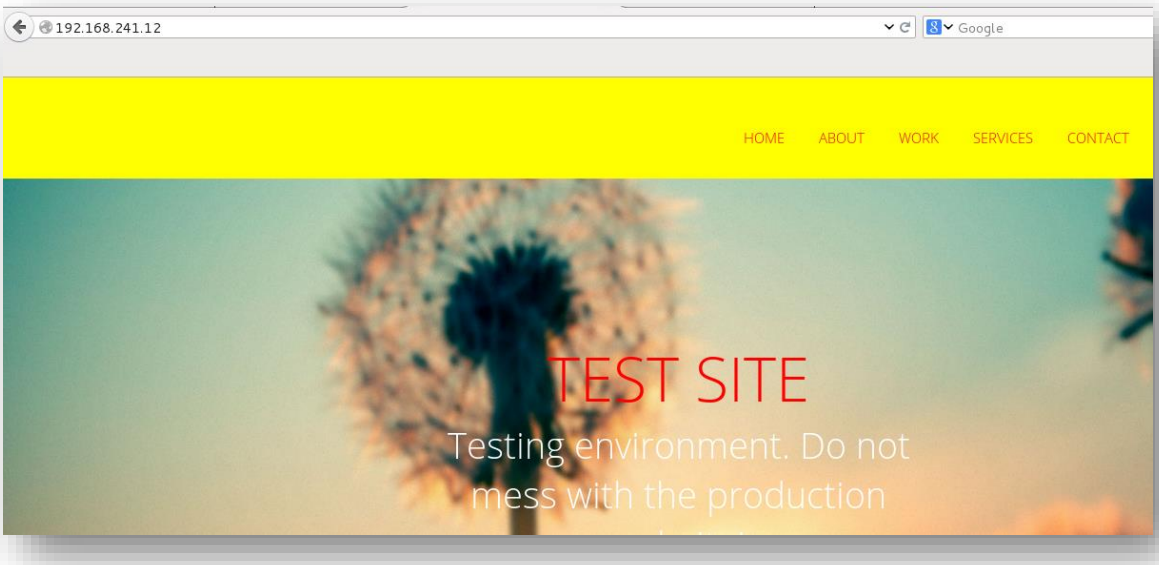
Now that we know our current configuration, let us try to connect via VPN to the lab and check the interfaces and the routes once again. You will see some differences:

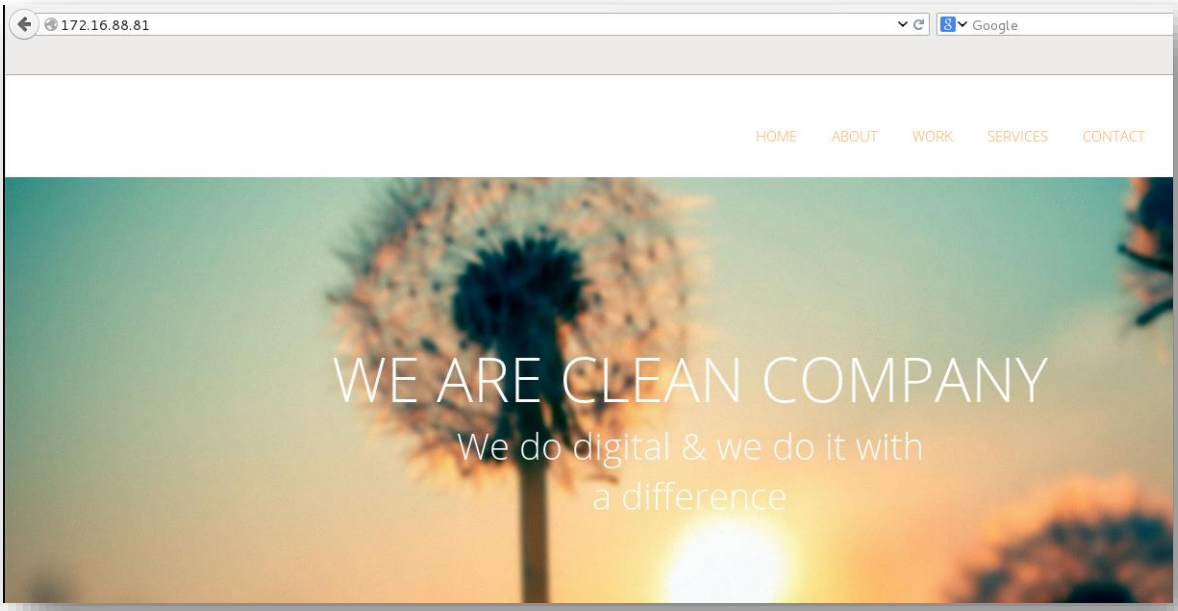


In the previous screenshot, we can see that we now have a new interface (tap0). This is a virtual interface created by *OpenVPN*. Moreover, we have three new routes: the first one (second line of the output) sets the route for the communication in the tap0 network. The second rule (third line) tells the system to route all the traffic destined to the 172.16.88.0/24 subnet through the tap0 interface. The last route (fifth line) works like the previous one, but for the 192.168.241.0/24 network.

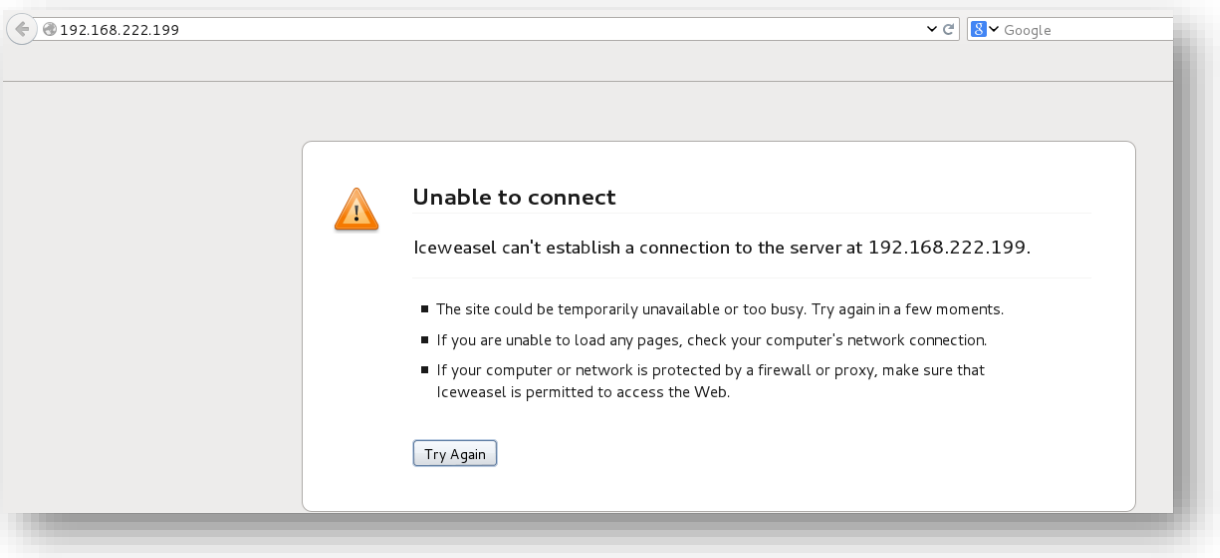
5.3. VISIT THE TWO WEB SERVERS

Right now, it seems we have the routes to reach two of the three servers. We can try to reach them by opening their IP addresses in your web browser (172.16.88.81 and 192.168.241.12).





As we can see, we are able to navigate them. This is possible because we have the routes configured in our system to reach them. However if we try to navigate the “Secret server” we obtain the following result:



We are not able to navigate the web server because we don’t have a route for that network.

5.4. ADD A ROUTE MANUALLY

Right now, we do not have any route set to reach the “Secret server”. Let us see how to add the correct route in our system!

We can simply do this with the following command:

```
root@litsnarf:~# ip route add 192.168.222.0/24 via 10.175.34.1
root@litsnarf:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.1.2     0.0.0.0         UG    0      0        0 eth0
10.175.34.0     *               255.255.255.0   U     0      0        0 tap0
172.16.88.0     10.175.34.1     255.255.255.0   UG    0      0        0 tap0
192.168.1.0     *               255.255.255.0   U     0      0        0 eth0
192.168.222.0   10.175.34.1     255.255.255.0   UG    0      0        0 tap0
192.168.241.0   10.175.34.1     255.255.255.0   UG    0      0        0 tap0
root@litsnarf:~#
```

Here we are saying our operating system to add a route for the 192.168.222.0/24 network and that the connections have to go through 10.175.34.1 (which is the gateway of the lab).

If we try to reach the Web Server once again we can see that we are now able to navigate it: