COMP 3721 – Fall 2024

GNS3 Lab1

1. Creating your first topology in GNS3

Now it is time to create your first topology in GNS3. Since we need to use Alpine Linux as a part of this lab, please install Alpine Linux appliance in your GNS3 based on the guidelines provided in section 2.4 of Lab0. Also, please ensure that GNS3 VM is running (using VMware Workstation Player or VMware Fusion) as a prerequisite prior to using GNS3 GUI.

First, create a "Blank Project" from the **File** menu, and then build the topology shown in Figure. 1 by dragging and dropping three elements into the project. These three elements are 2 x **Alpine Linux** hosts, 1 x **Ethernet Switch**, and 1 x **NAT cloud**.

To inter-connect the elements that were formed this topology, you can add link between them via the highlighted part with Red Circle in Figure. 1. Then, you need to power on the active elements of your topology by powering them on. To do this, press the Start/Resume all nodes button on the toolbox (highlighted with a Green Arrow in Figure. 1. After a few seconds, both Alpine Linux hosts will be up and you can access them via a console. To open these consoles, click on the part highlighted with Blue Square in Figure. 1. This opens two consoles for you to manage **AlpineLinux-1** and **AlpineLinux-2** hosts. You can do the same thing by right-click on each Alpine Linux host and choosing the "**>_ Console**" option. Figure. 2 shows the opened consoles along with the created topology.

Another important point during building this network topology is choosing the **GNS VM** as the Server whenever you drag-and-drop a new element (such as Ethernet Switch or NAT) from the side bar into the topology. This key point is shown in Figure. 3 and it ensure that the GNS3 VM as the server for these elements will provide necessary resources and capabilities to accurately simulate the behavior of Ethernet switches and NAT devices within your GNS3 topology.

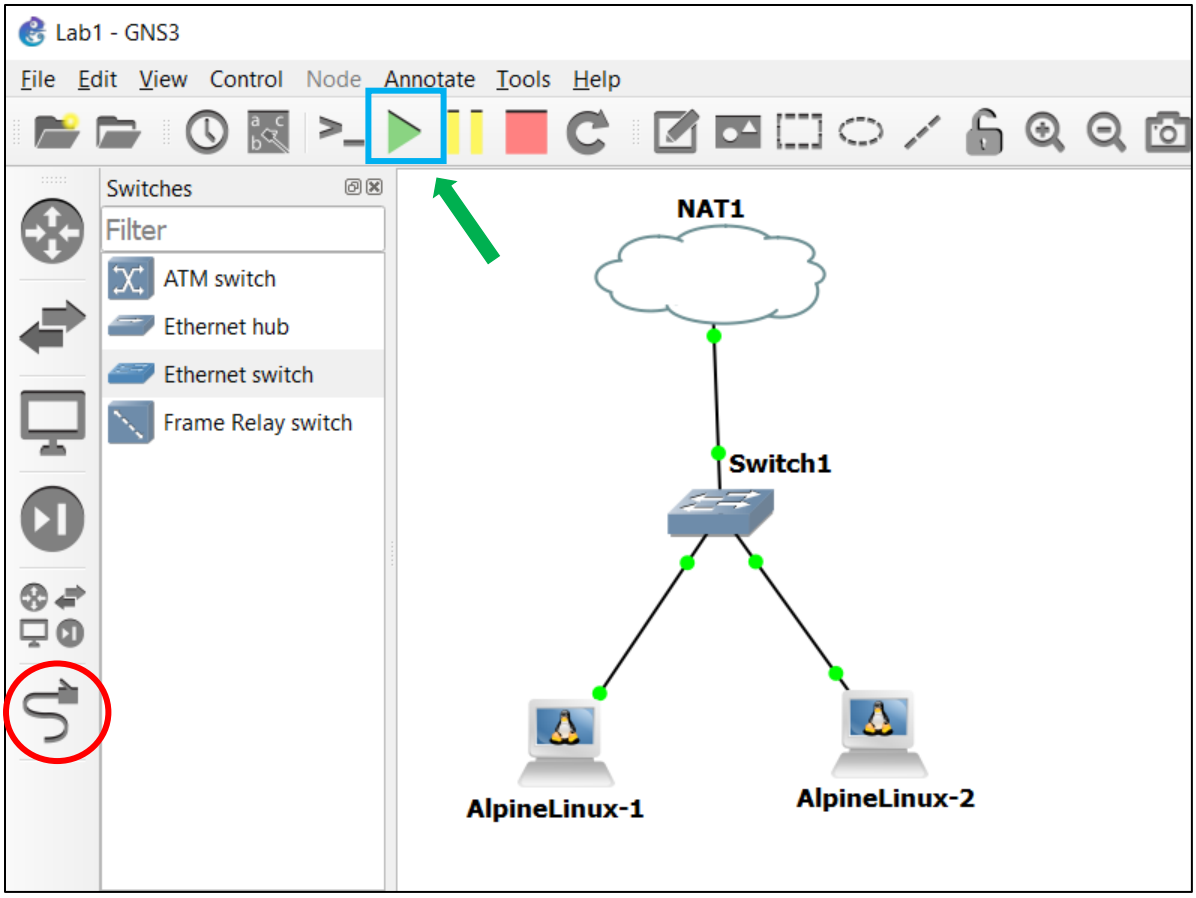

Figure. 1. The topology of first GNS3 project.

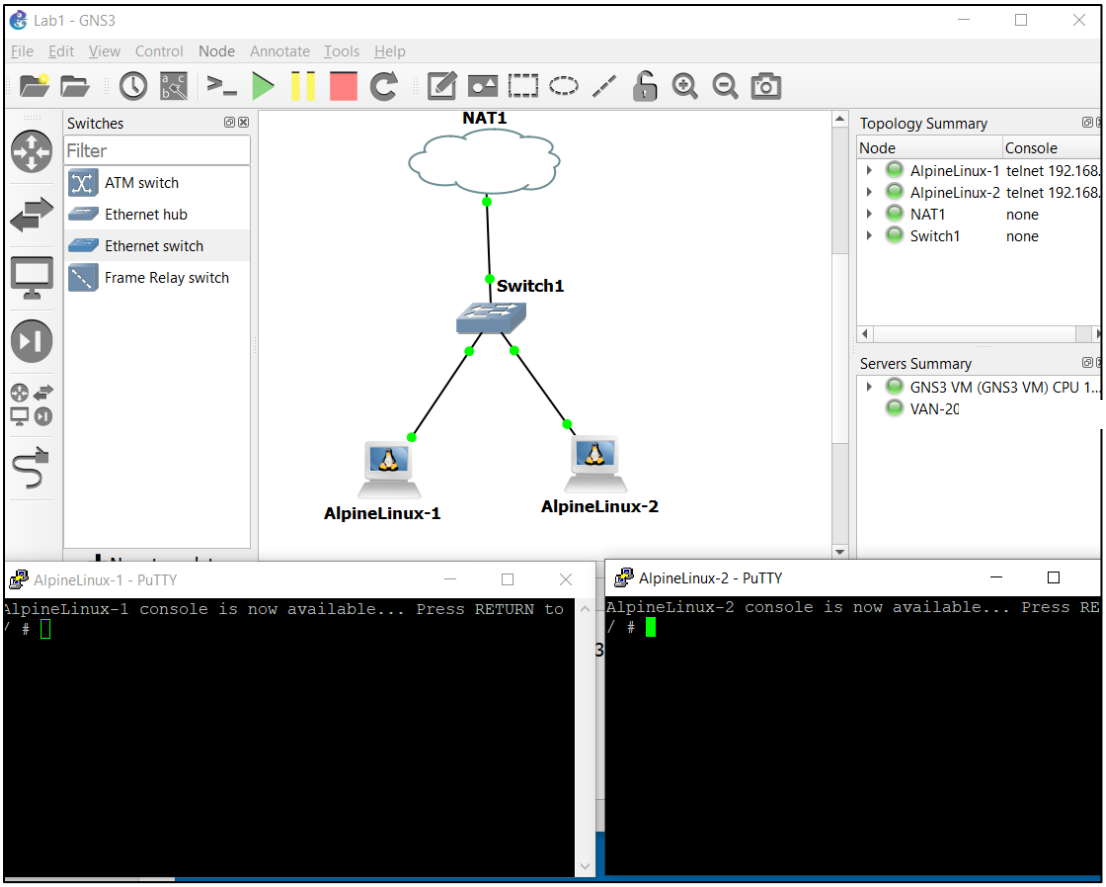

Figure. 2. Two consoles to access the Alpine Linux hosts.

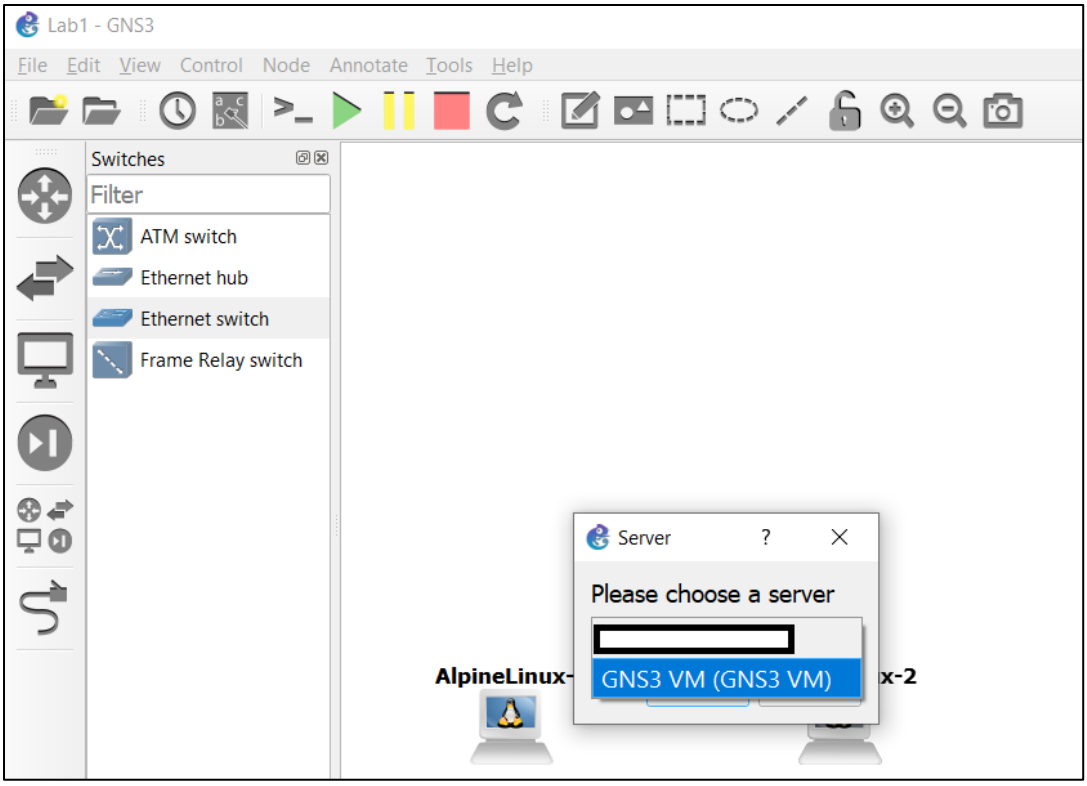

Figure. 3. Selecting GNS3 VM as the server when adding the Ethernet Switch into the topology.

1.1. Generating network traffic in your GNS3 topology

Now, you need to configure the IP addresses on the network interfaces of the Alpine Linux hosts to establish communication with both the Internet and other hosts within your topology. To do this, first run 'ifconfig' command on each Alpine Linux host to see the network interfaces available in your Alpine Linux. By default, each Alpine Linux has one **eth0** (Ethernet 0) and one **lo** (loopback) interface. The first interface (eth0) is the one that you can send/receive traffic using it. Before generating any traffic, you have to assign an IP address to this interface and it can be done by running '**udhcpc –i eth0**' command. If the Alpine Linux gets an IP address from the DHCP server (in your network) successfully, you will see the new IP address in the output of '**udhcpc**' as well as '**ifconfig**' commands, as shown in Figure. 4.

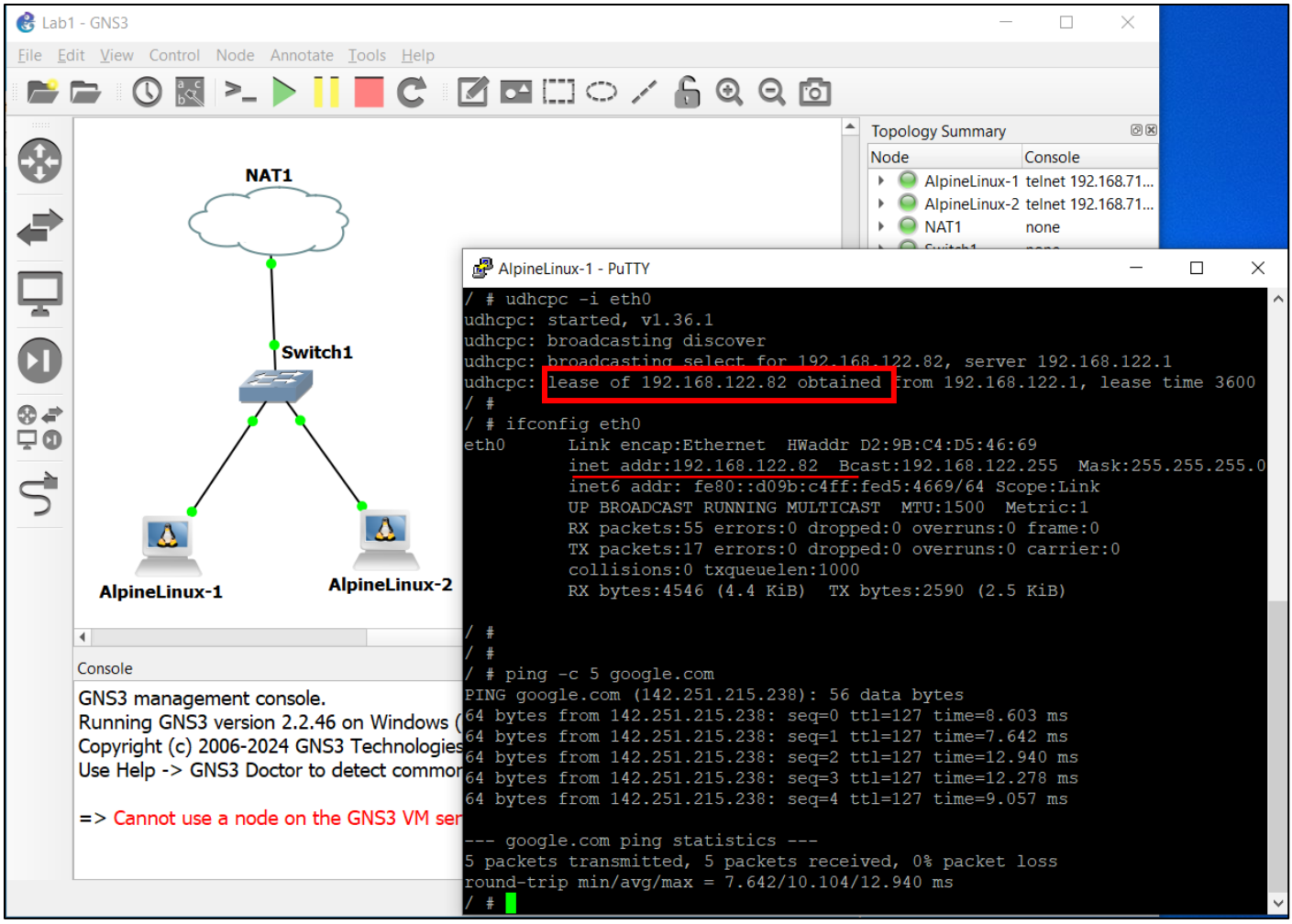

Figure. 4. IP assignment to Alpine Linux hosts using a `DHCP client.

After taking the above steps, answer the following questions:

1) What are the IP addresses that each Alpine Linux host obtained using 'dhcpc' command? You can run **'ifconfig'** command to see the IP address assigned to each host.
2) Run the '**ping**' command as shown in Figure. 4 to reach google.com from Alpine Linux 1. From the output of ping command, how much is average value of the Round-Trip time (RTT) in milliseconds (ms)?
3) After getting the IP address assigned to Alpine Linux2, open a console on Alpine Linux1 and run 'ping' command for the IP address of Alpine Linux2, i.e., '**ping –c 5 <Alpine Linux2 IP>**'. Then, record the average RTT value from the output of ping command.
4) How much is the time difference between answers from the last two questions? Why?
5) Run '**apk update'** and then '**apk add tcpdump**' commands on Alpine Linux1 host to install 'tcpdump' which is a known packet capturing tool in Linux

distributions. Then, run '**tcpdump –h**' command to check the packet installed successfully. From the output of last command, what is the version of 'tcpdump' package that you just installed?

6) Now, run '**tcpdump –i eth0 –w capture_lab1.pcap**' on Alpine Linux1. This command enables packet capturing on interface eth0 of this host and stores the captured traffic in a file named capture_lab1.pcap (note that the name is different in the example figure below). While tcpdump command is running on Alpine Linux1, open a console on Alpine Linux2 and ping Alpine Linux1 from this host. To do this, you may run '**ping –c 5 <Alpine Linux1 IP>**' command on Alpine Linux2. Once the execution of ping command is completed, stop tcpdump command by pressing **Ctrl + C** keys on Alpine Linux1 console. To check the stored file on Alpine Linux1, run '**ls -l**' command on the console which shows you '**capture_lab1.pcap**' as shown in Figure. 5. From the output of '**ls -l**' command, determine how much is the size of this file?

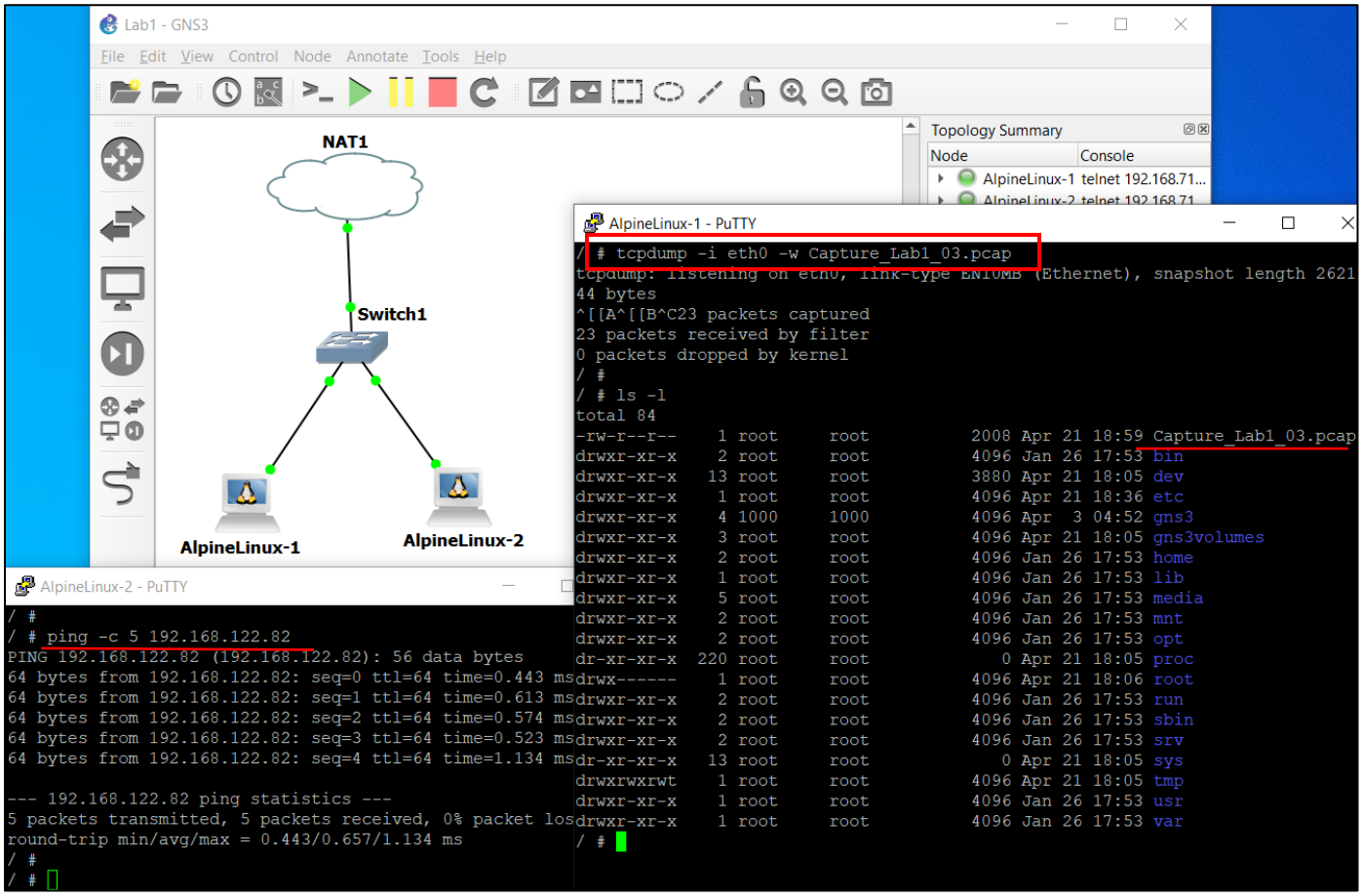

Figure. 5. Example: capturing generated ICMP traffic at Alpine Linux2 using tcpdump.

1.2 File transfer from GNS3 to your computer

We have the captured traffic via interface eth0 will be written inside the **capture_lab1.pcap** file. Now, the question is how this file can be transferred from the virtualized environment in GNS3 (i.e., the Alpine VM) to the operating system installed on your computer. There are multiple ways to do this, but one of the simplest ways is using SSH (Secure Shell) service. The first step is enabling the SSH service on your Microsoft Windows 10 or macOS.

In Windows, the SSH service is not installed by default, but it can be done by taking these steps. First, you need to go to your Windows '**Settings'** section by clicking on the Windows logo located the task bar and typing 'settings'. Then, you may choose '**System'** and next '**Optional Features**'. Now, by selecting '**Add a feature**' and searching for '**OpenSSH Server**' you can install it on the Windows. After a successful installation, you should be able to search and see OpenSSH Server in the 'Installed features' section as it is shown in Figure 6. The sequence of required steps to install SSH service in Windows 10 is displayed here as well.

Settings → System → Optional Features → Add a feature → OpenSSH Server

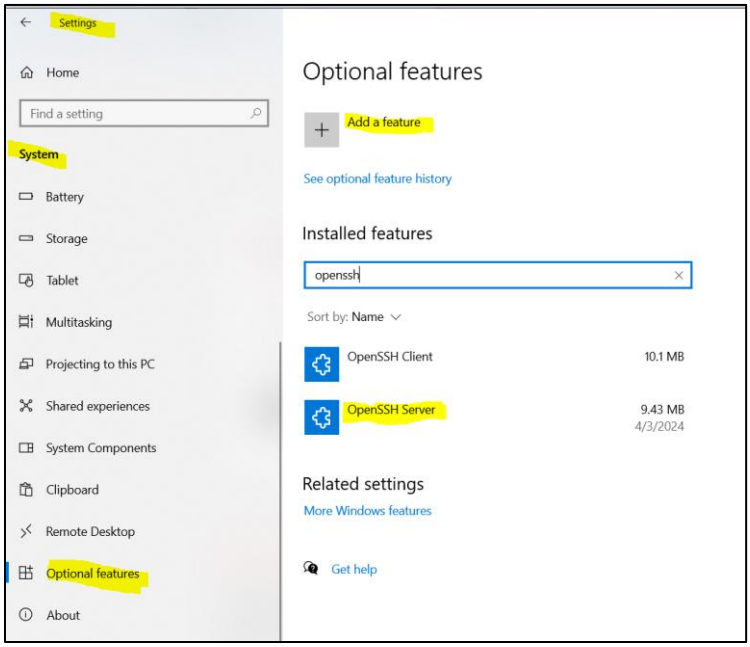

Figure 6. Installing OpenSSH Service in Windows 10.

When the OpenSSH service is installed, you have to start it by running '**services.msc**' command and right-click on '**OpenSSH Server**' and selecting the '**Start**' option. If the service successfully started, the service status will be changed to 'Running' as displayed in Figure 7.

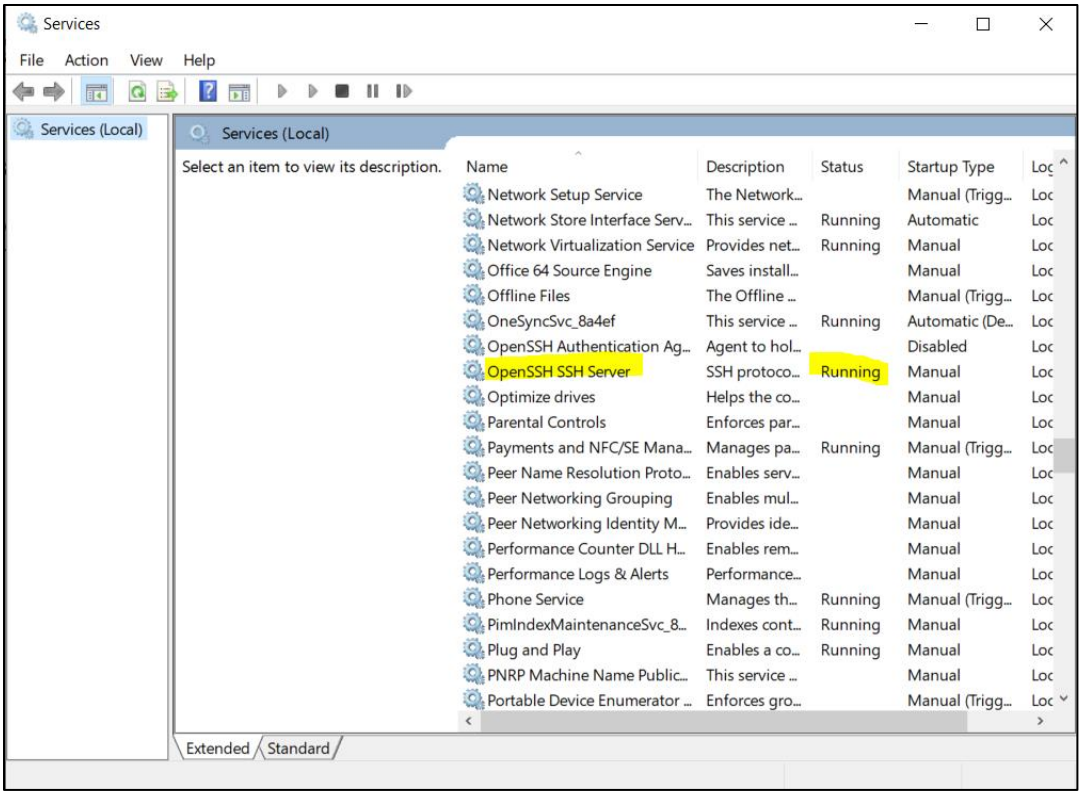

Figure 7. Enabling OpenSSH Service in Windows 10.

The last step before trying to test the OpenSSH service is creating a **Local User Account** in your Windows. If you have already an active Local User Account (which is not a domain user), you can skip this step. Now, by using '**scp**' or Secure Copy command, you can copy the file from Alpine Linux to the Windows home folder of the <username> stated in the command. By default, '**scp**' command is not installed on Alpine Linux and it can be installed by running '**apk update**' and '**apk add openssh**' commands from the terminal. Then, you have to run '**scp**' command as it is shown in the following example. If the file transfer completed successfully, you can find the **capture_lab1.pcap** file under **C:\Users\<username>** folder and you may open it via the Wireshark application.

 # scp capture_lab1.pcap <username>@<Windows IP>:~

If your username is **bob** and the IP address of your Windows is **192.168.1.10**, the '**scp**' command will be looks like this:

 # scp capture01.pcap bob@192.168.1.10:~

To get the IP address of your Windows, you can run '**ipconfig**' in the command line (cmd). If you see multiple IP addresses in the output of '**ipconfig**', use the one which is assigned to your wireless interface (if using Wi-Fi to connect the Internet).

In macOS, to enable the SSH service, you need to go to '**System Settings'** and pick the '**General**' option. Then, choose '**Sharing**' and enable '**Remote Login**' as shown in Figure 8.

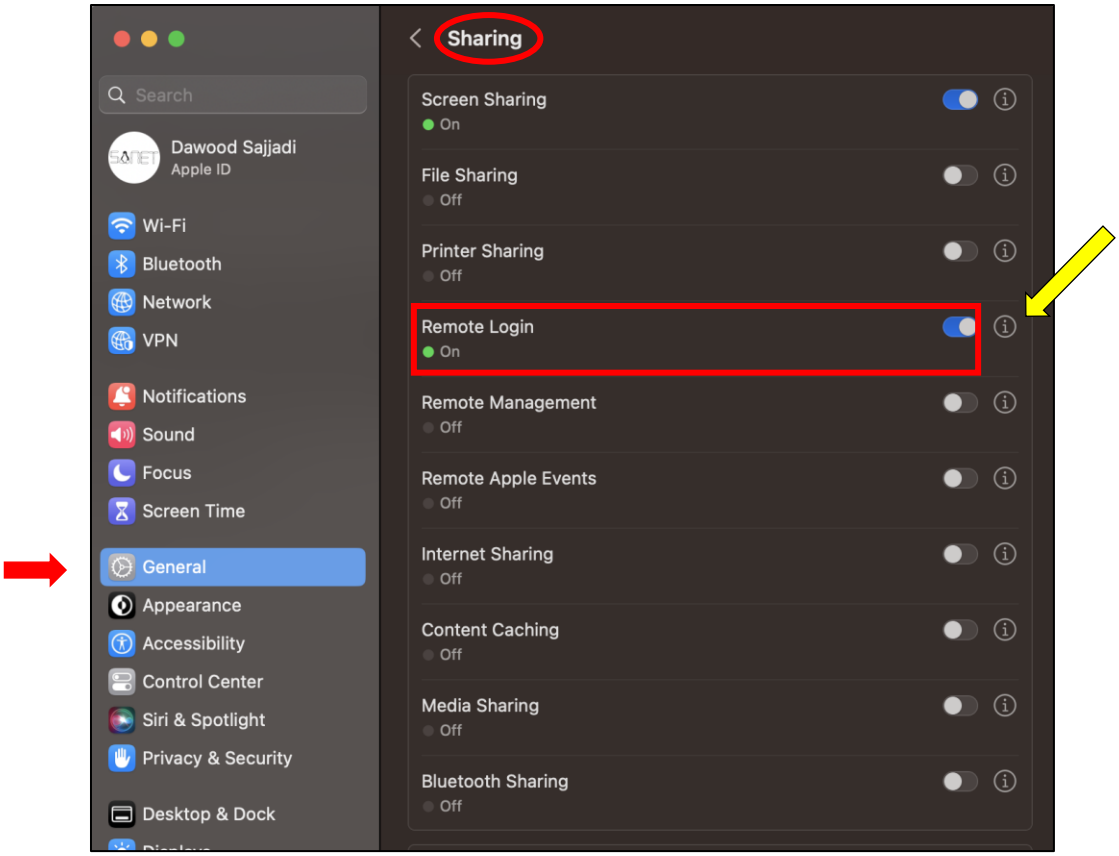

Figure 8. Enabling Remote Login (SSH) in macOS.

Also, you have to allow access to the user that want to log on to your macOS via the SSH service. To do this, you need to click on the circle which is highlighted with a yellow arrow in Figure 26. By clicking on the circle, a popup similar to the one shown in Figure 26 will be appeared and you can Add/Remove users to the existing list using +/- signs. To verify the status of SSH service on your macOS, you can run the following command in a Terminal. If there is an output for this command, then the SSH server is running and you can copy the file using '**scp**'.

```
# netstat -na | grep LISTEN | grep .22
```

On macOs, the command should be typed on the Alpine linux 1, which is the same as the windows platform
:~ is the location of the storage of file.

```
scp capture_lab1.pacap username@ip:~
```

Then, you can type **ls -l** to see the capture file listed in the current directory and confirm that the file transfer has been successful.

You can find more information about **scp** in the following:
https://linux.die.net/man/1/scp

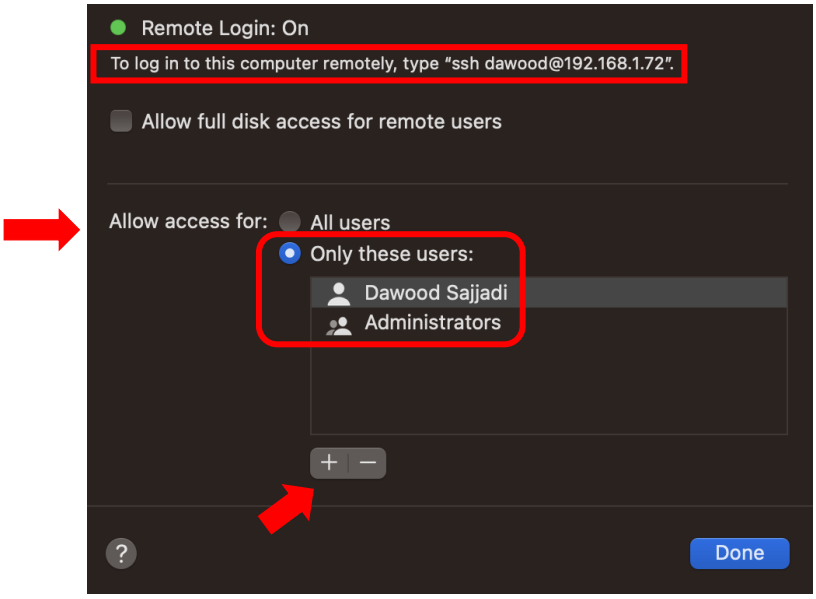

Figure 26. Allow Access to the Selected Users for Remote Login (SSH).

In the next lab, we will use Wireshark to open the captured file and analyze the packets in this file.

Written by:

Dawood Sajjadi

Maryam Tanha