

Packet Sniffing and Spoofing Lab

Due by midnight September 7, 2021

20 points

Lab Instructions

The lab manual can be obtained by visiting the Seed Labs site

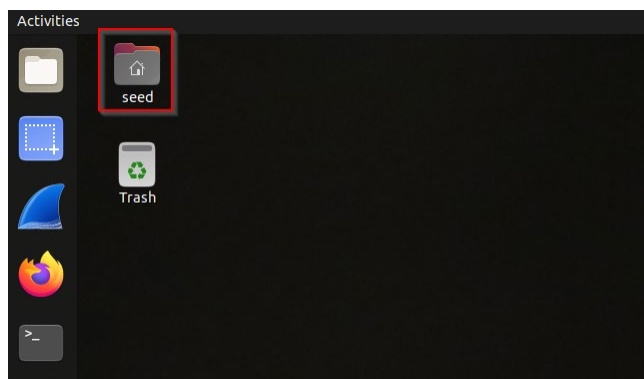
https://seedsecuritylabs.org/Labs_20.04/Networking/Sniffing_Spoofing/. Click Tasks (PDF).



- **VM version:** This lab has been tested on our [SEED Ubuntu-20.04 VM](#)
- **Lab setup files:** [Labsetup.zip](#)
- **Manual:** [Docker manual](#)

For this lab, you are required to complete both sets of the labs (Python and C) except for Task 1.3: Traceroute and Task 2.1C: Sniffing Passwords which will be counted as bonus points (5 points). You do not need to answer the six questions in Task 2.1A and Task 2.2B, respectively. You can download several sample codes **sniff.py**, **sniff.c**, **spoof_icmp.c**, **spoof.c**, **myheader.h**, and **checksum.c** from the Blackboard and use them as templates for your tasks. Below are some additional instructions and details that are not included in the original lab manual.

1. Inside the Ubuntu 20.04 VM, Click the seed folder.

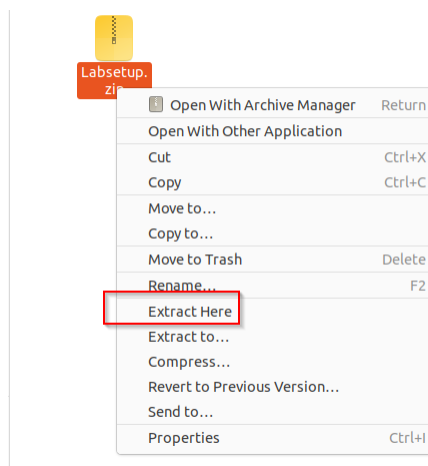


Inside the seed folder, right click the mouse and choose **New Folder**. Give a name to the new folder you just created. In my case, I named the folder **Lab**. Double click the new folder you just created to enter the folder. Right click the mouse and choose New Folder. Give a name to the new folder you just created. In my case, I named the folder **SNIFF**.

2. Bring up the Firefox in Ubuntu 20.04 and surf to the lab's website (https://seedsecuritylabs.org/Labs_20.04/Networking/Sniffing_Spoofing/). Right click on the **Lab setup files: Labsetup.zip**. In the pop up window, choose **Save Link As...** to save the Labsetup.zip file to the SNIFF folder you just created.



Double click the SNIFF folder to open it. You should see the Labsetup.zip file you just downloaded. Right click on the zip file and choose **Extract Here**.



Bring up a terminal in Ubuntu 20.04 VM and change directory to Labsetup by typing

\$ cd Lab/SNIFF/Labsetup

```
[07/17/21]seed@VM:~/Lab/SNIFF$ cd Labsetup/
[07/17/21]seed@VM:~/../Labsetup$ dcbuild
attacker uses an image, skipping
host uses an image, skipping
[07/17/21]seed@VM:~/../Labsetup$ dcup
Creating network "net-10.9.0.0" with the default driver
Creating seed-attacker ... done
Creating host-10.9.0.5 ... done
Attaching to seed-attacker, host-10.9.0.5
```

We are now ready to build lab environment using docker containers. If this is the first time you set up a SEED lab environment using containers, it is very important that you read the user manual. Make sure that you must execute the docker-compose commands within the Labsetup folder. First run **dcbuild** then **dcup** to bring up the containers.

Although you can either use the host VM or the attacker container as the attacker machine, to make your life easier, please use your Ubuntu 20.04 VM as the attacker machine instead of the attacker container.

Task 1.1B

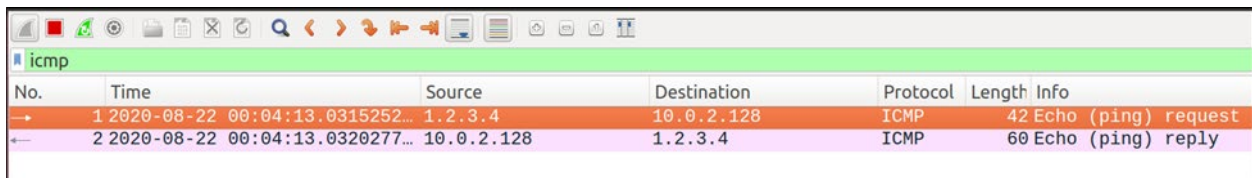
- Capture only the ICMP packet – after setting the filter on your sniff program, try to ping 8.8.8.8
- Capture any TCP packet that comes from a particular IP and with a destination port number 23 – after setting the filter on your sniff program, try to telnet from your host VM to the victim at 10.9.0.5.
- Capture packets comes from or to go to a particular subnet. You can pick any subnet, such as 128.230.0.0/16; you should not pick the subnet that your VM is attached to – you can use 153.91.1.0/24 - the subnet belongs to UCM and ping UCM's web server at 153.91.1.10 after setting the filter on your sniff program.

Task 2.1A

On the Host VM (attacker), open Firefox and surf to the Blackboard. Download **sniff.c**. You only need to modify one line of code based on your VM's NIC. Compile and run the sniff program using super user privilege. Open another terminal on the Attacker VM and ping 8.8.8.8. You should see the source IP, destination IP and protocol information printed by the sniff program.

Task 2.2B

On the Host VM (attacker), open Firefox and surf to the Blackboard. Download **spoof_icmp.c**, **spoof.c**, **myheader.h**, and **checksum.c**. The spoof_icmp.c program spoofs an ICMP echo request packet on behalf of another machine (i.e., using another machine's IP address as its source IP address). This packet should be sent to the victim container. You should turn on your Wireshark on the attacker VM, so if your spoofing is successful, you can see the echo reply coming back from the victim machine. A sample screenshot is shown below. In the example, a spoofed ICMP echo request packet with a fake non-existing IP address 1.2.3.4 as the source IP address is sent to victim machine at 10.0.2.128. An ICMP echo reply was successfully generated by the victim machine.



No.	Time	Source	Destination	Protocol	Length	Info
1	2020-08-22 00:04:13.0315252...	1.2.3.4	10.0.2.128	ICMP	42	Echo (ping) request
2	2020-08-22 00:04:13.0320277...	10.0.2.128	1.2.3.4	ICMP	60	Echo (ping) reply

You need to modify spoof_icmp.c to update the IP address you will use for the experiment. After that, compile the program by typing

```
$ gcc -o icmp_spoof spoof_icmp.c checksum.c spoof.c
```

Task 2.3

You can combine the **sniff.c** and **spoof_icmp.c** together. Please do not ping a fake IP on the same LAN of victim container. Sample screenshots are shown below. On victim container, we ping a non-existing IP address 1.2.3.4. Because of the sniffing and spoofing program running on the attacker machine, we received the spoofed reply.

```
VM:~$ gcc -o sniff_spoof sniff_spoof.c checksum.c spoof.c -lpcap
VM:~$ sudo ./sniff_spoof
```

```
[08/22/20]seed@VM:~$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
8 bytes from 1.2.3.4: icmp_seq=4 ttl=20 (truncated)
8 bytes from 1.2.3.4: icmp_seq=5 ttl=20 (truncated)
8 bytes from 1.2.3.4: icmp_seq=6 ttl=20 (truncated)
```

3. Once you finish the lab, inside the **Labsetup** folder, issue **dcdownd** to stop and remove the containers.

```
[07/17/21]seed@VM:~/.../Labsetup$ dcdownd
Stopping host-10.9.0.5 ... done
Stopping seed-attacker ... done
Removing host-10.9.0.5 ... done
Removing seed-attacker ... done
Removing network net-10.9.0.0
[07/17/21]seed@VM:~/.../Labsetup$
```

Lab Report

- please include your name and 700# at the beginning of your report
 - please upload your report to the Blackboard by the due date
 - only word or pdf format is acceptable for the report
1. provide necessary screenshots and explanations for all tasks
 2. Create a folder and name it your last name, first name, and lab number. For example, John_Doe_Lab_1. This is where you will save each of your C, Python codes and your lab report. When the lab is completed, zip (compress) this folder and submit it on the blackboard in the Lab 1 Report link
 3. The zipped folder should include
 - A lab report with all screenshots and answers to questions
 - All your C and Python codes used to complete each task in the lab. The codes must be saved using the correct file extension. Each task will be a separate C or Python file. Please name your file based on different task, e.g., Task_1_1.py. If you decide to save your codes in word and/or notepad, it will receive a zero score
 4. If you decide to only submit the lab report without the supporting C and Python codes, it will receive a zero score