

Foundations Network Security

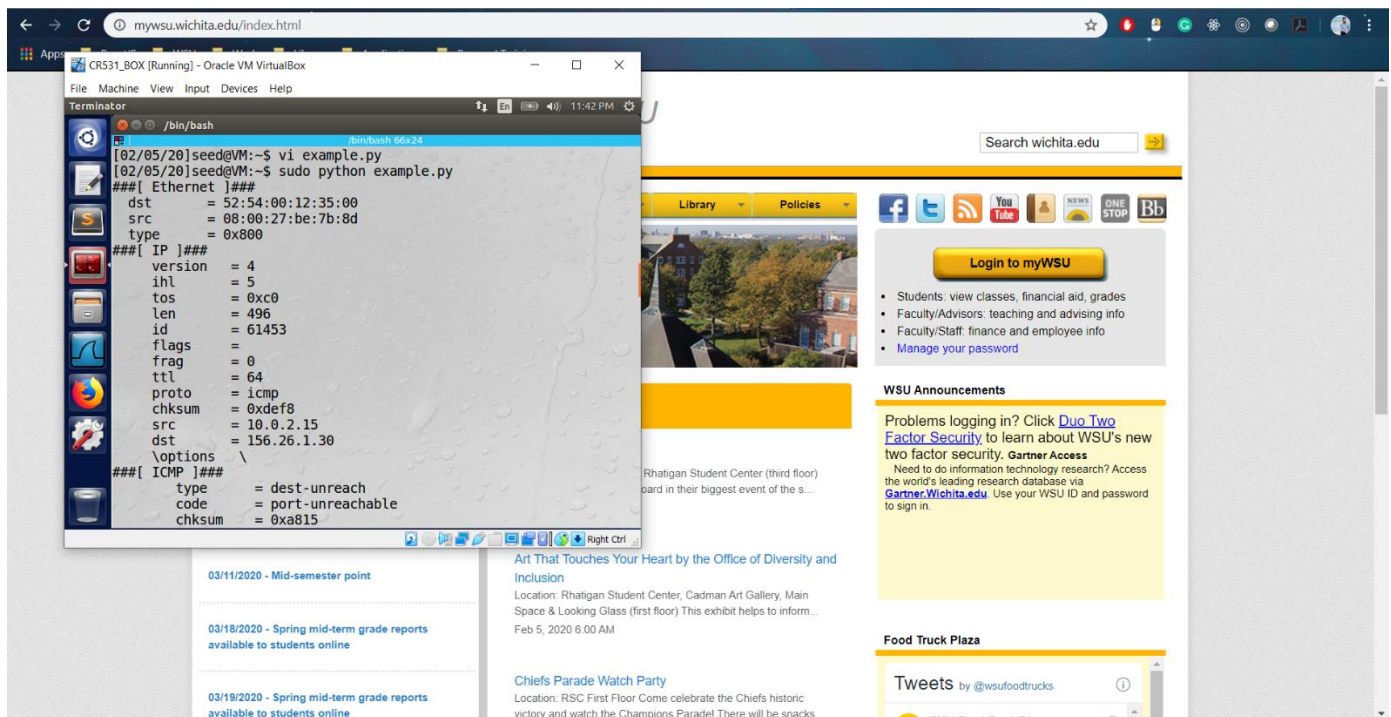
Chakradhar Reddy Donuri

E949F496

Task 1.1A :

Include a screen shot of the output of your Python script on the terminal showing the packets it captured while using the root privilege

\$ sudo python example.py (with root privilege)



Why is the output of your program different without using the root privilege?

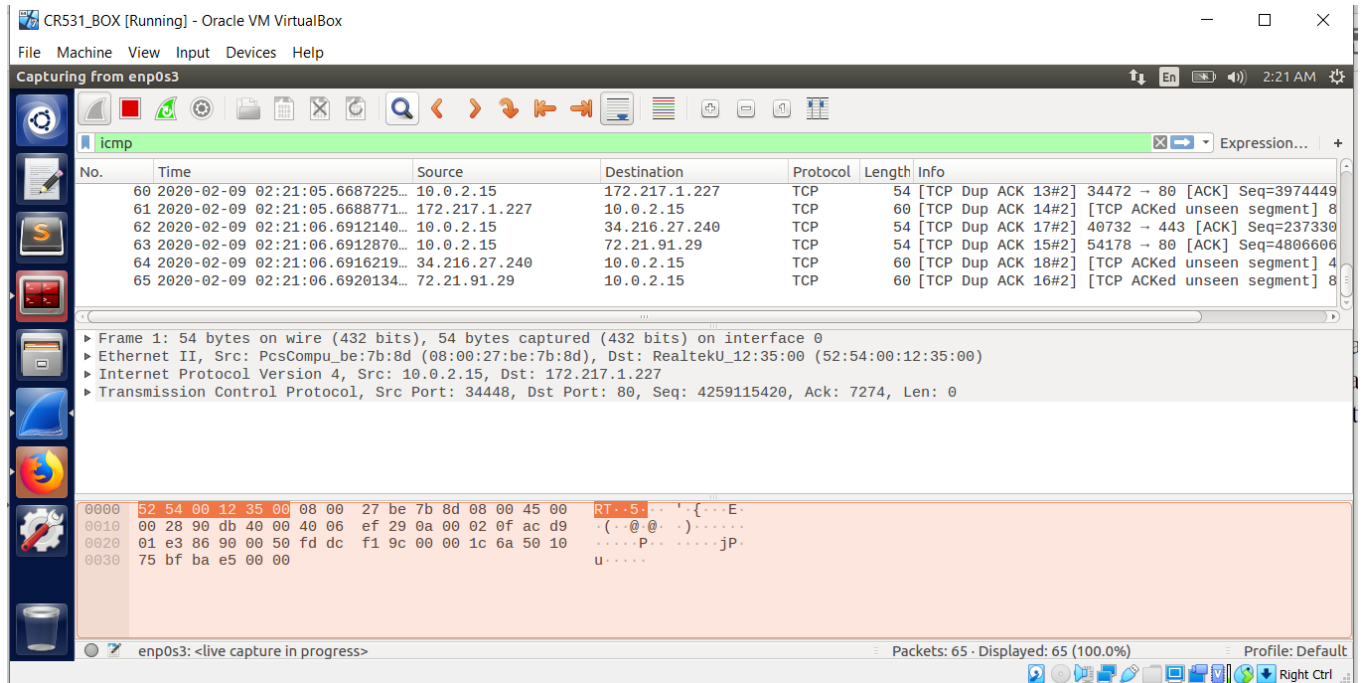
\$ python example.py (without root privilege)

Without the root privilege, it just shows an error. Because I'm writing the given python code in the text editor, I use the terminal to translate the code from the text editor into its output. In the code, there's a header that describes what file will be used with the python code. scapy function is pivotal for the sniffing packets. I ran the program and saw the capturing of packets with and without root privilege.

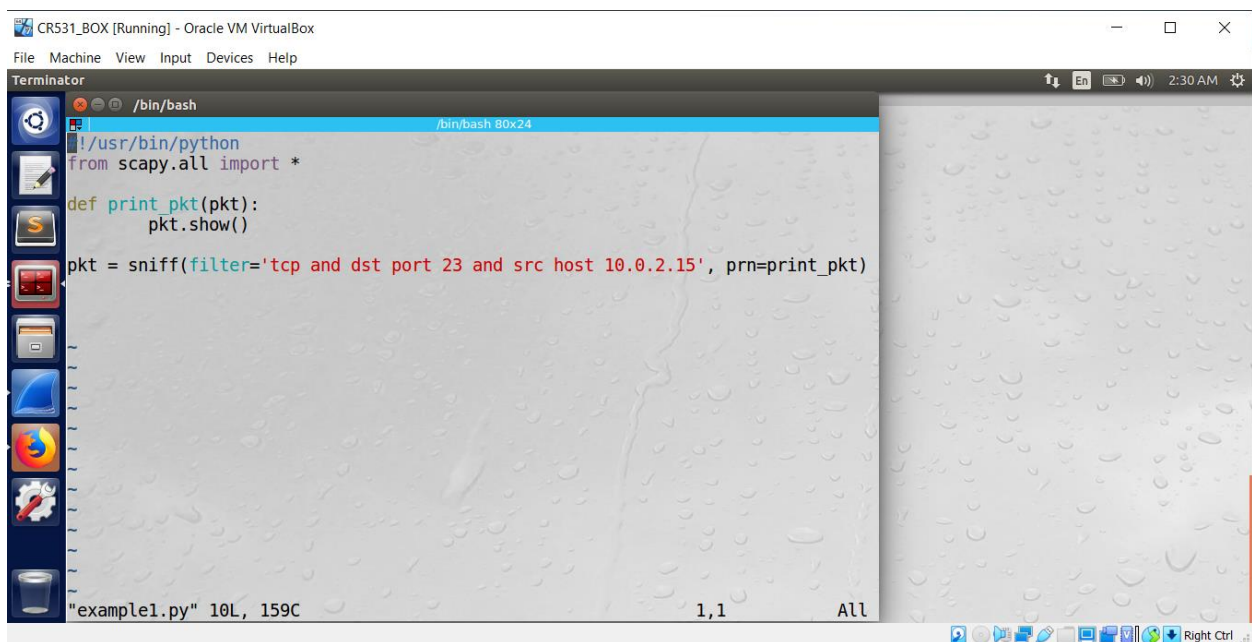
Task 1.1B.

Include a screen shot of your sniffer only capturing the specified type of packets. Include one screen shot for each of the three items in the list below

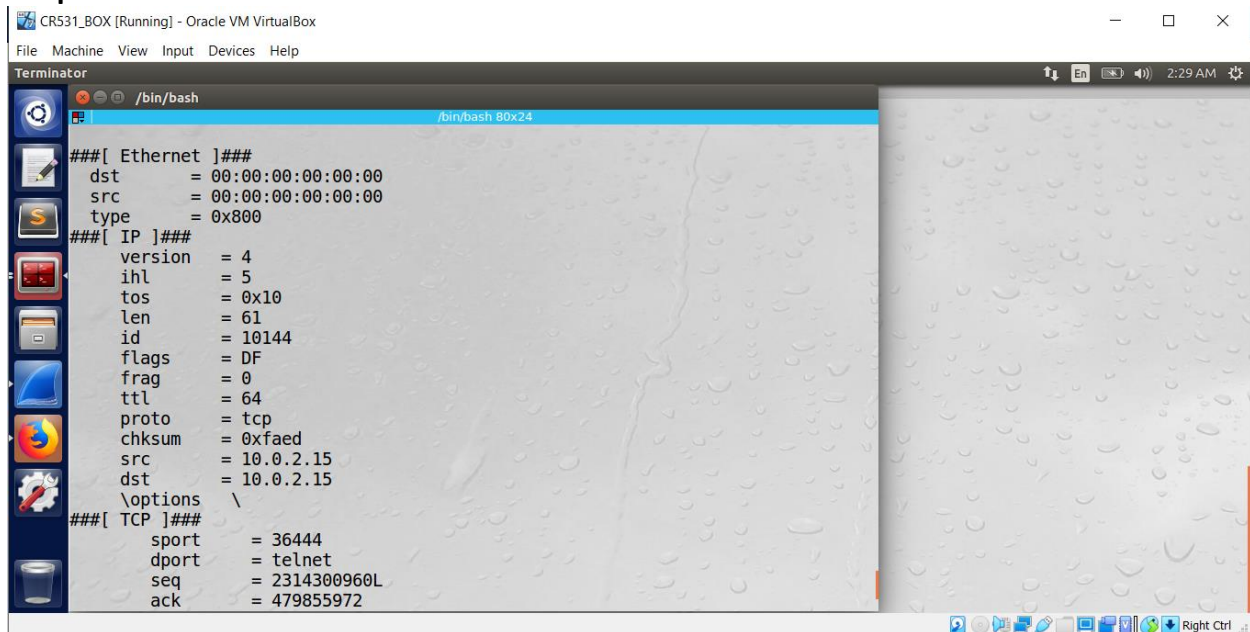
1. Capture only ICMP packets



2. Capture any TCP packet that comes from a particular IP and with a destination port number 23. You can choose any IP address.



Output:



```
CR531_BOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
/bin/bash 80x24

###[ Ethernet ]###
dst      = 00:00:00:00:00:00
src      = 00:00:00:00:00:00
type     = 0x800
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x10
len      = 61
id       = 10144
flags    = DF
frag     = 0
ttl      = 64
proto    = tcp
chksum   = 0xfaed
src      = 10.0.2.15
dst      = 10.0.2.15
\options \
###[ TCP ]###
sport    = 36444
dport    = telnet
seq      = 2314300960L
ack      = 479855972
```

3. Capture packets that 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.



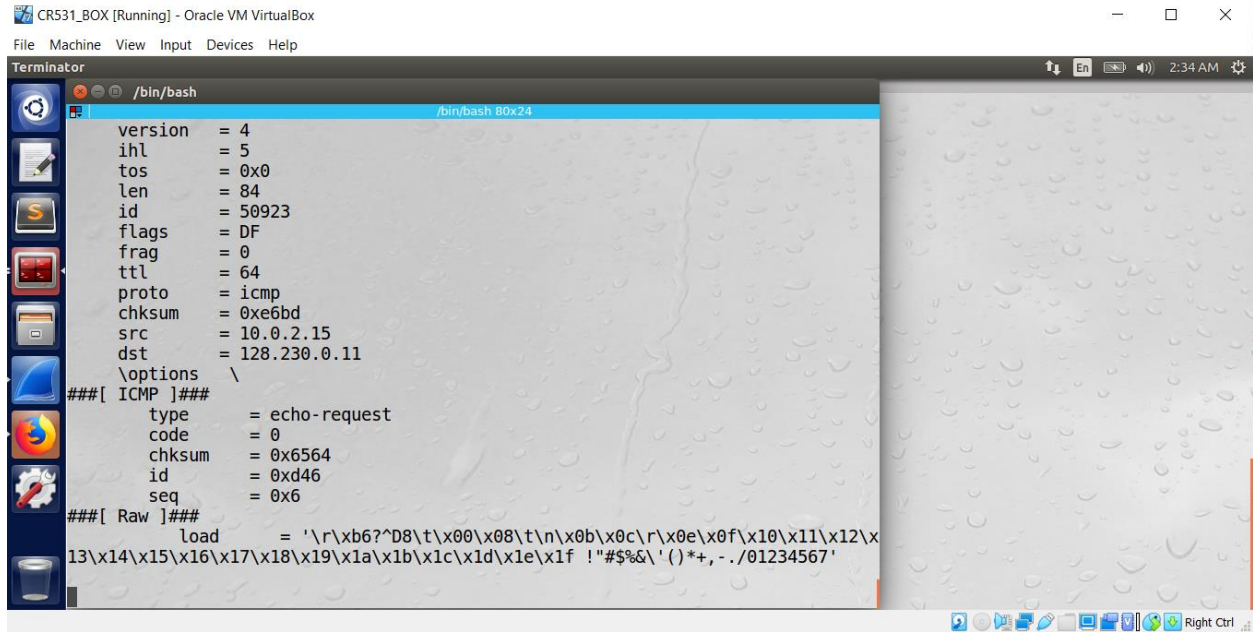
```
CR531_BOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
/bin/bash 80x24

#!/usr/bin/python
from scapy.all import *

def print_pkt(pkt):
    pkt.show()

pkt = sniff(filter='dst net 128.230.0.0/16', prn=print_pkt)

"example1.py" 10L, 139C
1,1 All
```



Task 1.2

For this task setup two VMs: VM1 and VM2. Use VM1 to craft the spoofed packet with an arbitrary source IP address. In VM2, set up the packet sniffer and capture the spoofed packet from VM1. Include a screen shot of VM1 sending the spoofed packet and a screen shot of VM2 receiving it.

There's an echo ping request, meaning that I successfully spoofed or impersonated an ICMP echo request packet.


```
CR531_BOX Clone [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

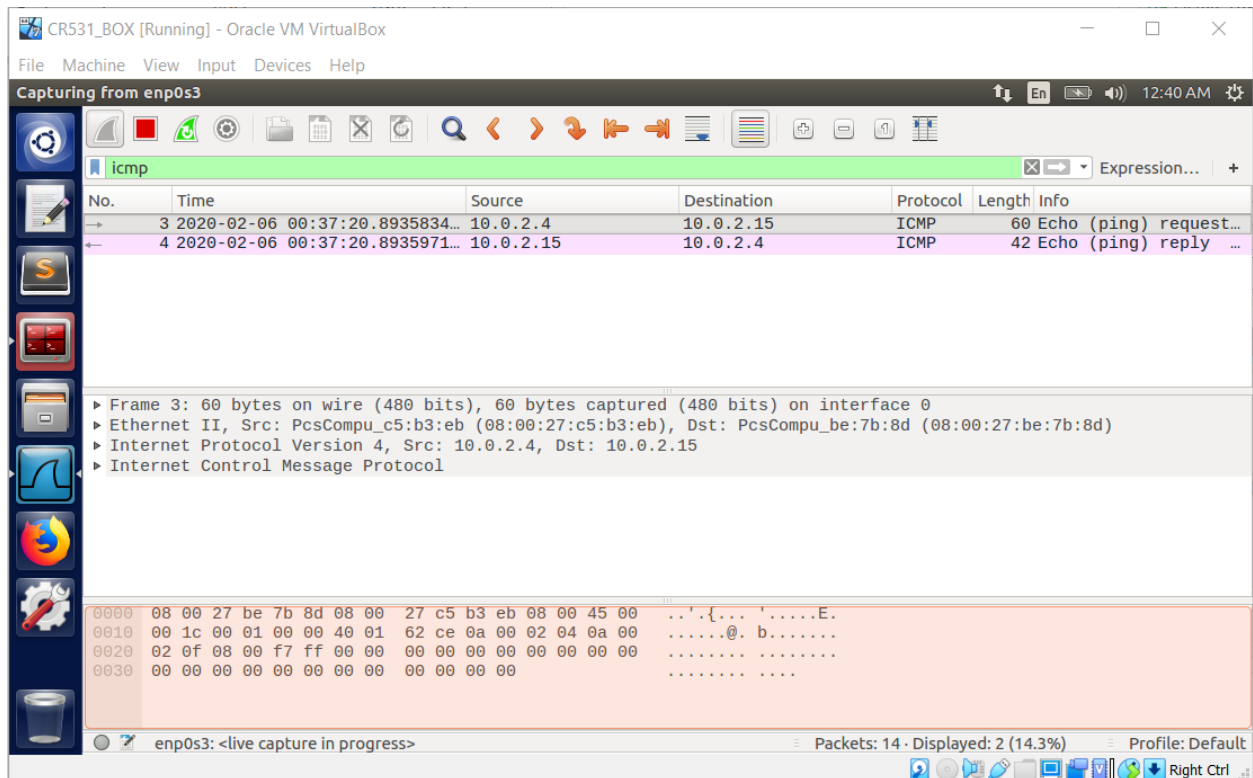
Terminator 12:41 AM
/bin/bash
/bin/bash 62x24

enp0s8  Link encap:Ethernet  HWaddr 08:00:27:c5:b3:eb
       inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255
       inet6 addr: fe80::aa21:18ef:b4f3:92cd/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:12 errors:0 dropped:0 overruns:0 frame:0
       TX packets:70 errors:0 dropped:0 overruns:0 carrier:
       collisions:0 txqueuelen:1000
       RX bytes:2180 (2.1 KB)  TX bytes:7939 (7.9 KB)

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:76 errors:0 dropped:0 overruns:0 frame:0
       TX packets:76 errors:0 dropped:0 overruns:0 carrier:
       collisions:0 txqueuelen:1
       RX bytes:21848 (21.8 KB)  TX bytes:21848 (21.8 KB)

[02/06/20]seed@VM:~$ vi example3.py
[02/06/20]seed@VM:~$ vi example3.py
[02/06/20]seed@VM:~$ sudo python example3.py
.
Sent 1 packets.
[02/06/20]seed@VM:~$
```

This is detailed information about the contents of the captured ICMP echo request packet



Task 1.3

A screen shot of the round trip times calculated by your Python script or by Wireshark. If you wrote a Python script include it here.

I typed a ttl and manually typed numbers 1, 2, 3, 4, 5, 6, 7, and so on to see if I received a Time to live field.

