ECE 8860: Distributed Denial of Service Attacks

Spring 2023

Laboratory 1

Name: Sandali Sanjay Nemmaniwar

Email: sandaln@g.clemson.edu

Major: Computer Science

Abstract

This report is about the study that imparts knowledge on traffic sniffing and spoofing, which are essential for network traffic analysis. The first section, part A, explains how to gather network packets with standard tools and select data collecting points on a network. While Part A also gives us an understanding of low-level network communication and the structure of a network packet, part B throws insight on how to use network tools like Nmap and Scapy and guides us through the methods employed in spoofing and detection.

Part A- Traffic Sniffing

Introduction:

1) What is the problem?
   Due to the heavy traffic, it is difficult to gather data from a network to undergo network traffic analysis.

2) How do you propose a solution to solve the problem?
   The solution to this problem is being able to select only necessary packets and place it on the network. This will ease the process of data collection.

<u>Methodology:</u>

Setup:

1) Configure listed networks.
2) Create traffic between Nodes 1 and 2.
3) Using Wireshark or Tshark, gather and save packets from the indicated locations.
4) Use appropriate capture and display filters to limit the data collection for a particular packet type (IP address, protocol type, etc.).

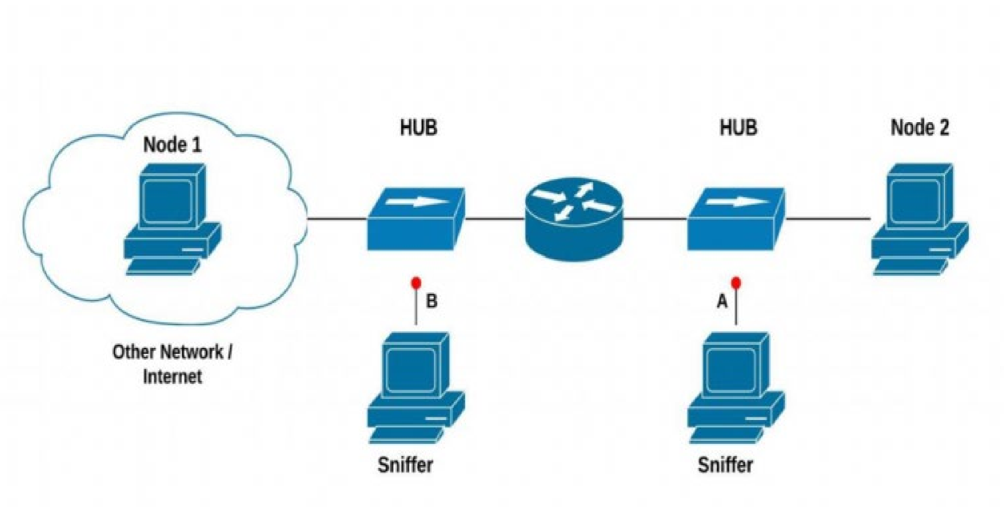


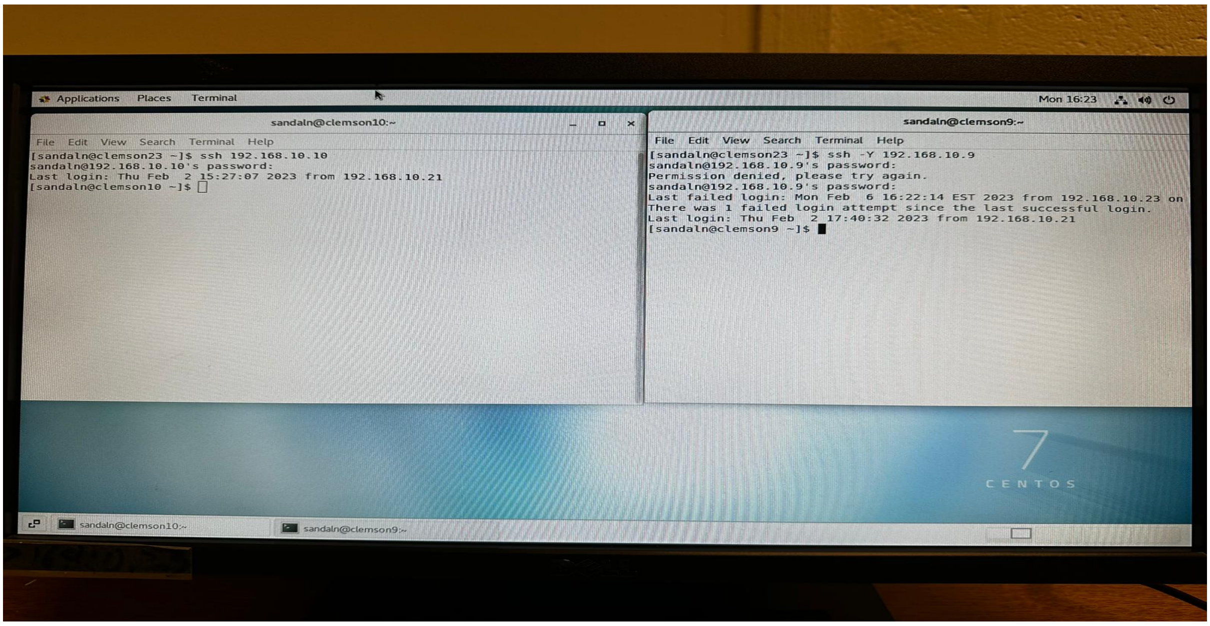
Fig: Experiment setup for sniffing using a router. Red dots are the sniffing points.

<u>Analysis:</u>

1) Used the TCP and ICMP filters to capture packets on wireshark that used a particular protocol type.

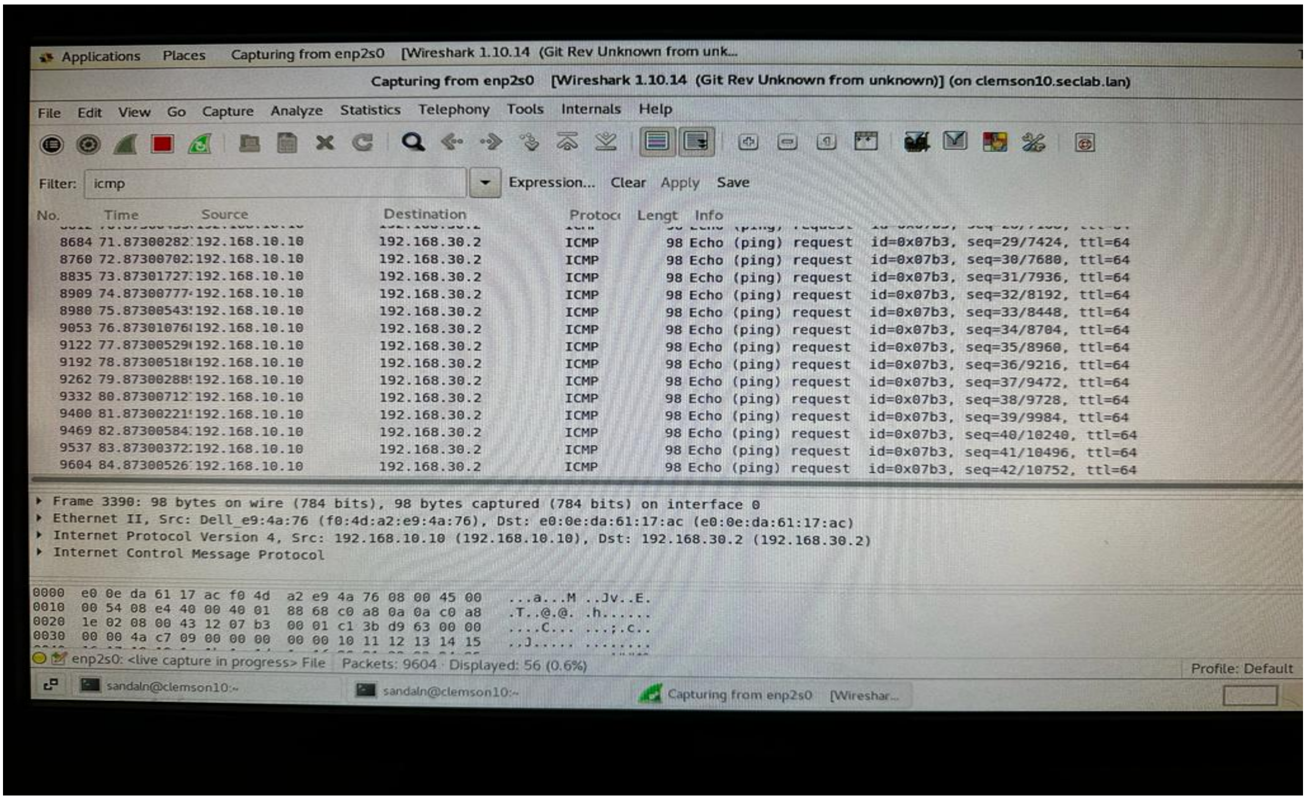
2) To display data, a time-series graph, or a graph of the quantity of packets vs time, was plotted.

Results:



```
[sandaln@clemson23 ~]$ ssh 192.168.10.10
sandaln@192.168.10.10's password:
Last login: Thu Feb  2 15:27:07 2023 from 192.168.10.21
[sandaln@clemson10 ~]$ 
```

```
[sandaln@clemson23 ~]$ ssh -Y 192.168.10.9
sandaln@192.168.10.9's password:
Permission denied, please try again.
sandaln@192.168.10.9's password:
Last failed login: Mon Feb  6 16:22:14 EST 2023 from 192.168.10.23 on
There was 1 failed login attempt since the last successful login.
Last login: Thu Feb  2 17:40:32 2023 from 192.168.10.21
[sandaln@clemson9 ~]$ 
```

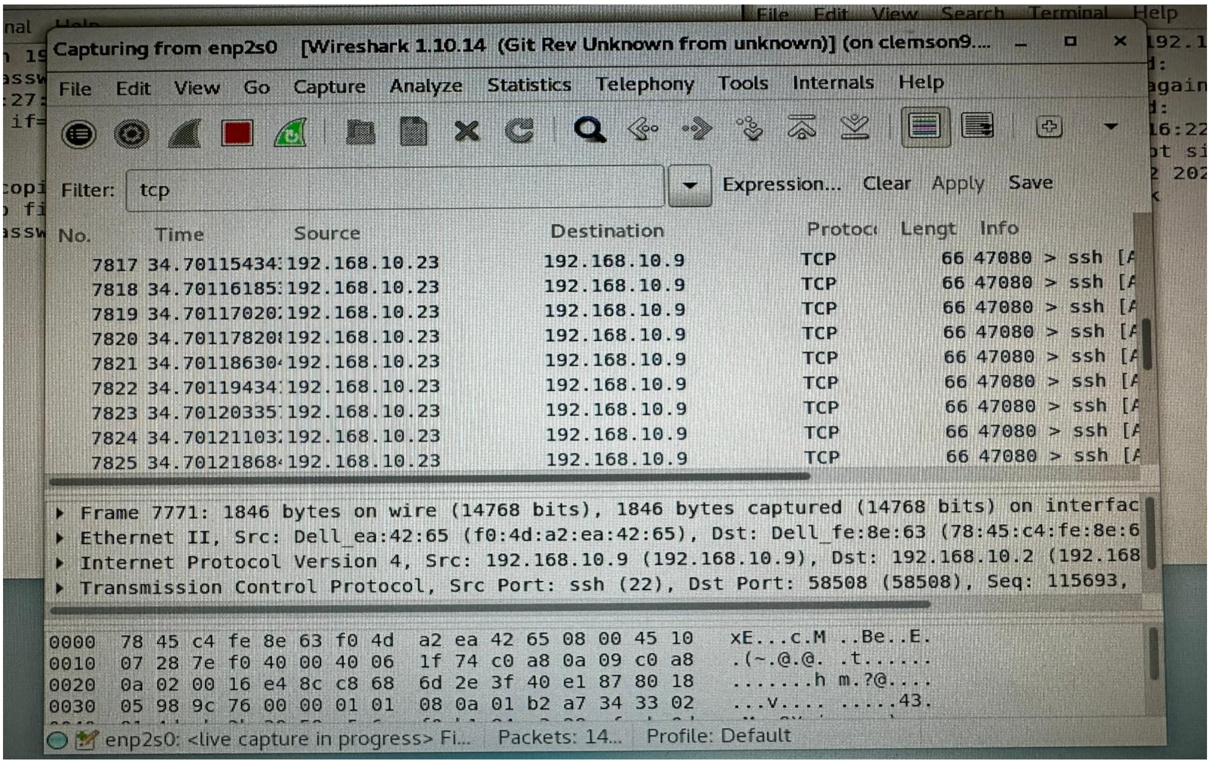


Applications
Places
Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown from unk...
Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown from unknown)] (on clemson10.seclab.lan)
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Tools
Internals
Help
Expression...
Clear
Apply
Save
Profile: Default
sandaln@clemson10:~
sandaln@clemson10:~
Capturing from enp2s0 [Wireshar...

Filter: icmp

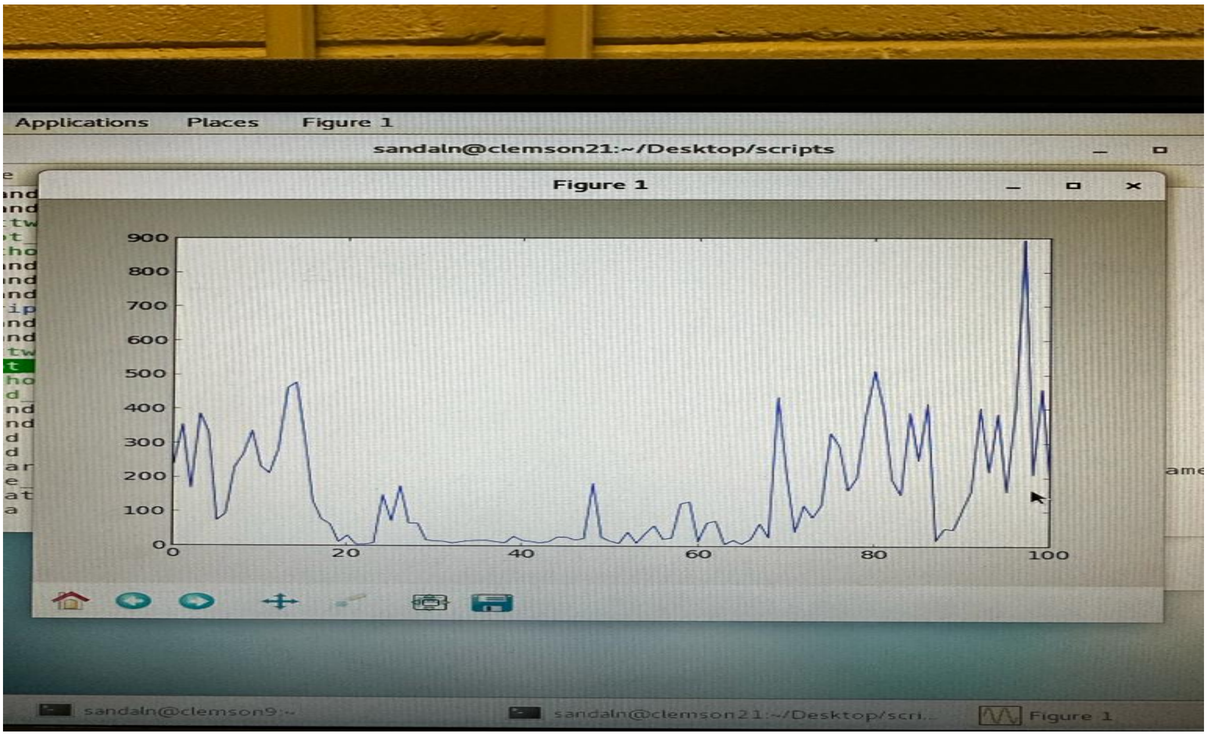
```
No.   Time         Source          Destination     Protoc  Lengt Info
8684 71.87300282  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=29/7424, ttl=64
8760 72.87300702  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=30/7680, ttl=64
8835 73.87301727  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=31/7936, ttl=64
8909 74.87300777  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=32/8192, ttl=64
8980 75.87300543  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=33/8448, ttl=64
9053 76.87301076  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=34/8704, ttl=64
9122 77.87300529  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=35/8960, ttl=64
9192 78.87300518  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=36/9216, ttl=64
9262 79.87300288  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=37/9472, ttl=64
9332 80.87300712  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=38/9728, ttl=64
9400 81.87300221  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=39/9984, ttl=64
9469 82.87300584  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=40/10240, ttl=64
9537 83.87300372  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=41/10496, ttl=64
9604 84.87300526  192.168.10.10   192.168.30.2    ICMP    98 Echo (ping) request  id=0x07b3, seq=42/10752, ttl=64

▸ Frame 3390: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: Dell_e9:4a:76 (f0:4d:a2:e9:4a:76), Dst: e0:0e:da:61:17:ac (e0:0e:da:61:17:ac)
▸ Internet Protocol Version 4, Src: 192.168.10.10 (192.168.10.10), Dst: 192.168.30.2 (192.168.30.2)
▸ Internet Control Message Protocol

0000  e0 0e da 61 17 ac f0 4d  a2 e9 4a 76 08 00 45 00   ...a...M ..Jv..E.
0010  00 54 08 e4 40 00 40 01  88 68 c0 a8 0a 0a c0 a8   .T..@.@. .h......
0020  1e 02 08 00 43 12 07 b3  00 01 c1 3b d9 63 00 00   ....C... ...;.c..
0030  00 00 4a c7 09 00 00 00  00 00 10 11 12 13 14 15   ..J..... ........

enp2s0: <live capture in progress> File   Packets: 9604 · Displayed: 56 (0.6%)
```

Fig.1: filters captured using ICMP protocol.


Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown from unknown)] (on clemson9....
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: tcp
Expression... Clear Apply Save

```
No.   Time          Source          Destination     Protoco  Lengt  Info
7817 34.70115434 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A
7818 34.70116185 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A
7819 34.70117020 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A
7820 34.70117820 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A
7821 34.70118630 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A
7822 34.70119434 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A
7823 34.70120335 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A
7824 34.70121103 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A
7825 34.70121868 192.168.10.23   192.168.10.9    TCP      66 47080 > ssh [A

▸ Frame 7771: 1846 bytes on wire (14768 bits), 1846 bytes captured (14768 bits) on interfac
▸ Ethernet II, Src: Dell_ea:42:65 (f0:4d:a2:ea:42:65), Dst: Dell_fe:8e:63 (78:45:c4:fe:8e:6
▸ Internet Protocol Version 4, Src: 192.168.10.9 (192.168.10.9), Dst: 192.168.10.2 (192.168
▸ Transmission Control Protocol, Src Port: ssh (22), Dst Port: 58508 (58508), Seq: 115693,

0000  78 45 c4 fe 8e 63 f0 4d  a2 ea 42 65 08 00 45 10   xE...c.M ..Be..E.
0010  07 28 7e f0 40 00 40 06  1f 74 c0 a8 0a 09 c0 a8   .(~.@.@. .t......
0020  0a 02 00 16 e4 8c c8 68  6d 2e 3f 40 e1 87 80 18   .......h m.?@....
0030  05 98 9c 76 00 00 01 01  08 0a 01 b2 a7 34 33 02   ...v.... .....43.

enp2s0: <live capture in progress> Fi...   Packets: 14...   Profile: Default
```

Fig.2: filters captured using TCP protocol.



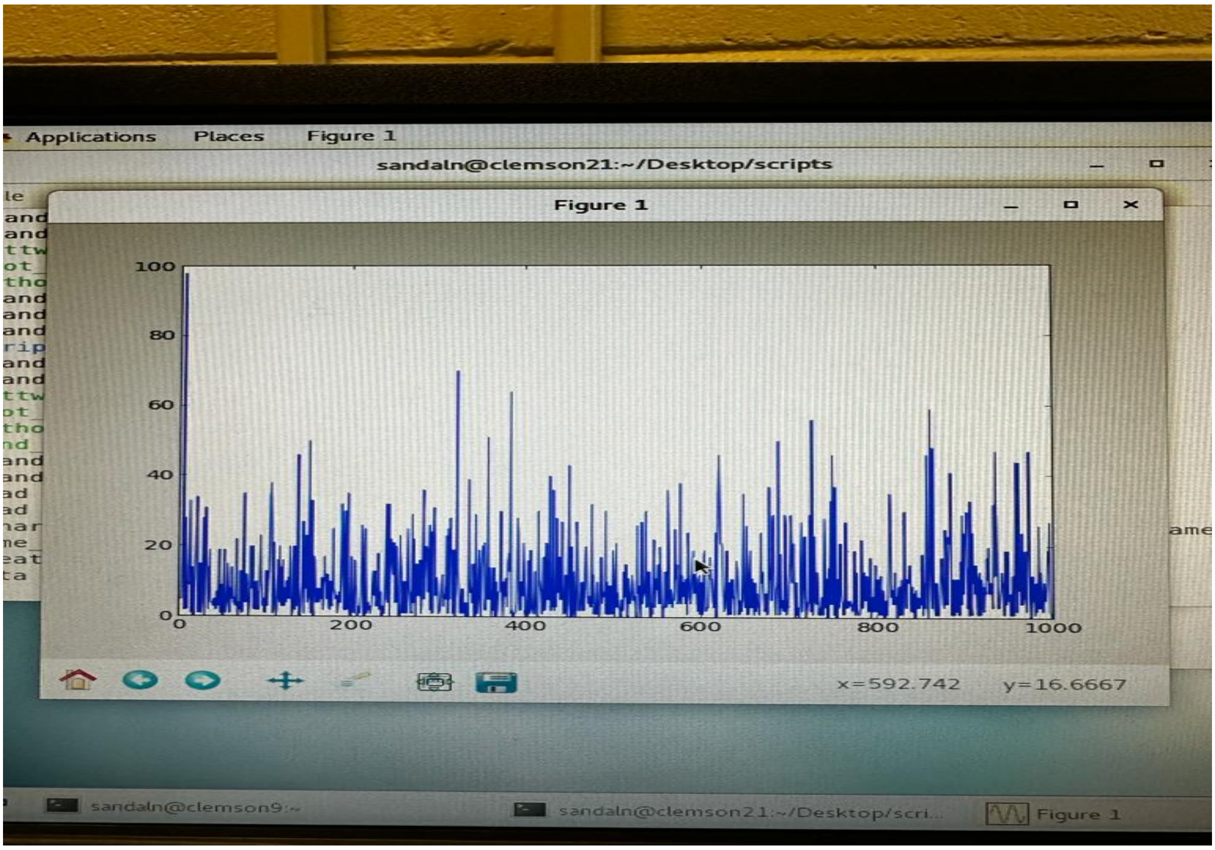


Fig.3: time series graph for the packets captured using TCP protocol.

Conclusion:

In the findings reported above, a Time series graph displays the captured filters with ICMP and the captured packets.

Questions:

1) Describe the layer and the fields of a packet captured from your network.
   The layers captured from the network are as follows:
   - **Frame 1**- A frame is merely a straightforward container for a single network packet in packet switching systems. A frame is a repeating structure that supports time-division multiplexing in other telecommunications systems.
   - **Ethernet II** - Ethernet frames are the enclosed data that the Network Access layer defines. An Ethernet frame begins with a header, which includes information such as the source and destination MAC addresses. The actual data is in the centre of the frame. The field Frame Check Sequence appears at the conclusion of the frame (FCS).
   - **Internet protocol** - A collection of guidelines for addressing and routing data on the Internet is known as the Internet Protocol (IP). TCP and UDP are just two of the transport protocols that can be used with IP.
   - **Transmission control protocol (TCP)** - TCP is a communications protocol that enables computer hardware and software to exchange messages over a network. It is made to send packets across the internet and make sure that data and messages are successfully sent through networks.

The fields captured from the network are as follows:

- **UDP** - UDP is a communications protocol that is largely used to provide low-latency, loss-tolerant connections between internet-based applications.
- **CRC** - A method of error detection that produces a string of two 8-bit block check characters using a polynomial to represent the whole block of data.

2) Explain the characteristics and functioning of a hub, switch, and router in network science.
HUB: A Hub is a networking device that allows you to connect multiple PCs to a single network. It is utilized to link LAN segments.
    - Characteristics-
        1. Broadcasting and shared bandwidth are compatible.
        2. One broadcast domain and one collision domain are present.
        3. Works at the OSI model's physical layer.
        4. A hub cannot be used to construct a virtual LAN.
        5. Support for half-duplex transmission mode is provided.
        6. There is only one broadcast domain for a hub.
        7. Not compatible with the spanning tree protocol
        8. Most packet collisions happen inside a hub.
    - Functioning-
        1. A hub uses the physical layer to operate.
        2. Frame flooding, which can be unicast, multicast, or broadcast, is done by hubs.
        3. In a hub, there is only one collision domain.
        4. Half-duplex is the transmission mode.
        5. In accordance with the OSI model, hubs operate as Layer 1 devices.

SWITCH: A network switch is a piece of hardware used in computer networking that joins various components of a computer network. Information delivered via networks in the form of electronic data may also be routed using the switch.

- Characteristics-
    1. Device with Datalink Layer (Layer 2)
    2. It utilizes a fixed bandwidth.
    3. It keeps a table of MAC addresses.
    4. Enables you to build a virtual LAN.
    5. It functions as a multiple-port bridge.
    6. Typically has 24 to 48 ports.
    7. fully and half-duplex transmission modes are supported.

- Functioning-
    1. A switch operates on the data link layer.
    2. When necessary, it executes broadcast, followed by unicast and multicast.
    3. Different ports each have their own collision domain.

4. Full duplex transmission is the mode.
5. You can operate at Layer 2 of the OSI model thanks to network switches.

ROUTER: Routers are networking devices that operate at layer 3, or a network layer, of the OSI model. Receiving, analyzing, and transmitting data packets between the connected computer networks is their responsibility.

- Characteristics –
  1. A router is a layer 3 or network layer device.
  2. It transfers data packets from one network to another and links many networks together.
  3. Both LANs (Local Area Networks) and WANs can use routers (Wide Area Networks).
  4. IP packets are used to transport the data. It employs the IP address listed in the IP packet's destination field to transmit data.
- Functioning –
  1. With various network topologies, routers help transmit packets by navigating the sea of interconnected networking devices.
  2. Routers are clever machines that keep records of the networks to which they are connected.
  3. To convert from LAN framing to WAN framing, routers work with a channel service unit/data service unit (CSU/DSU). Because LANs and WANs employ distinct network protocols, this is necessary.
  4. Border routers are such routers. They function at the edge of your network and act as the LAN's external connection to the WAN.

3) Why is the use of SDNs as a tool necessary for DDoS?
SDNs stands for software-defined networking, which is the ability to dynamically initialize control over, modify, and manage network activity. Future networks could benefit from the architecture offered by software-defined networking. DDoS assaults are simple to recognize and respond to because of SDN's capabilities, such as software-based traffic analysis and other features.

Part B- Spoofing

Introduction:

The motive of this lab is the students spoof other students' IP addresses and examine if it was spoofed. This session aids in the learning of common network utilities like Nmap and Scapy.

Methodology:

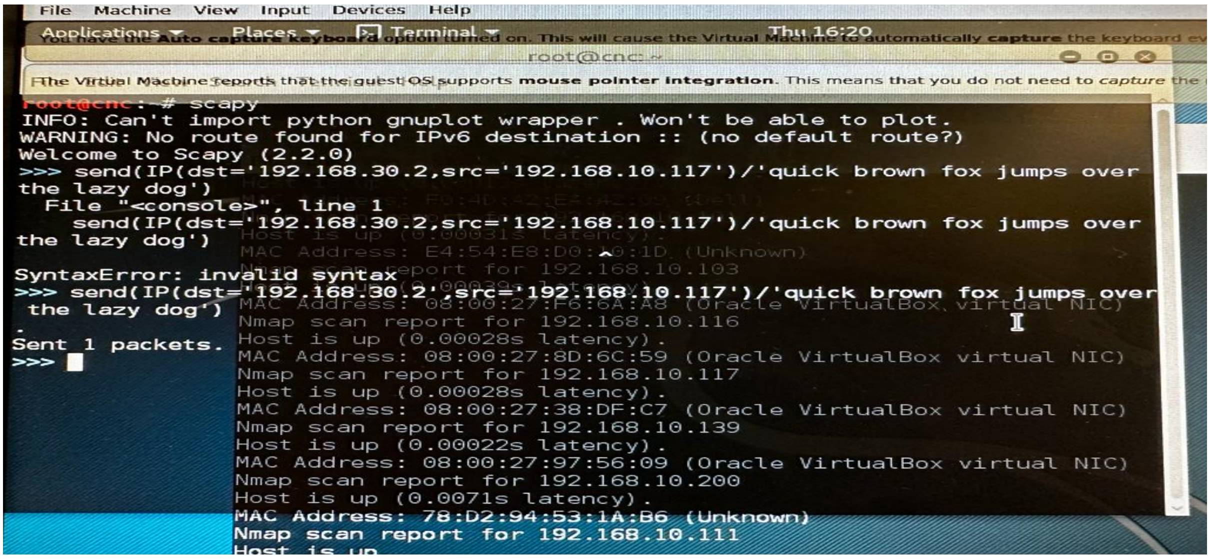
Setup:

1) connect to the machines in the security lab.
2) List every host that is currently online in the security lab's subnet.
3) Sniffing with Wireshark/Tshark
4) Sending out and catching spoof packets

Analysis:

1) Connected to the Virtual box through the spoof terminal.
2) Used Nmap to find all of the active hosts in the subnet, which provided the hosts' Mac and IP addresses, allowing me to utilize them to send packets.

Results:



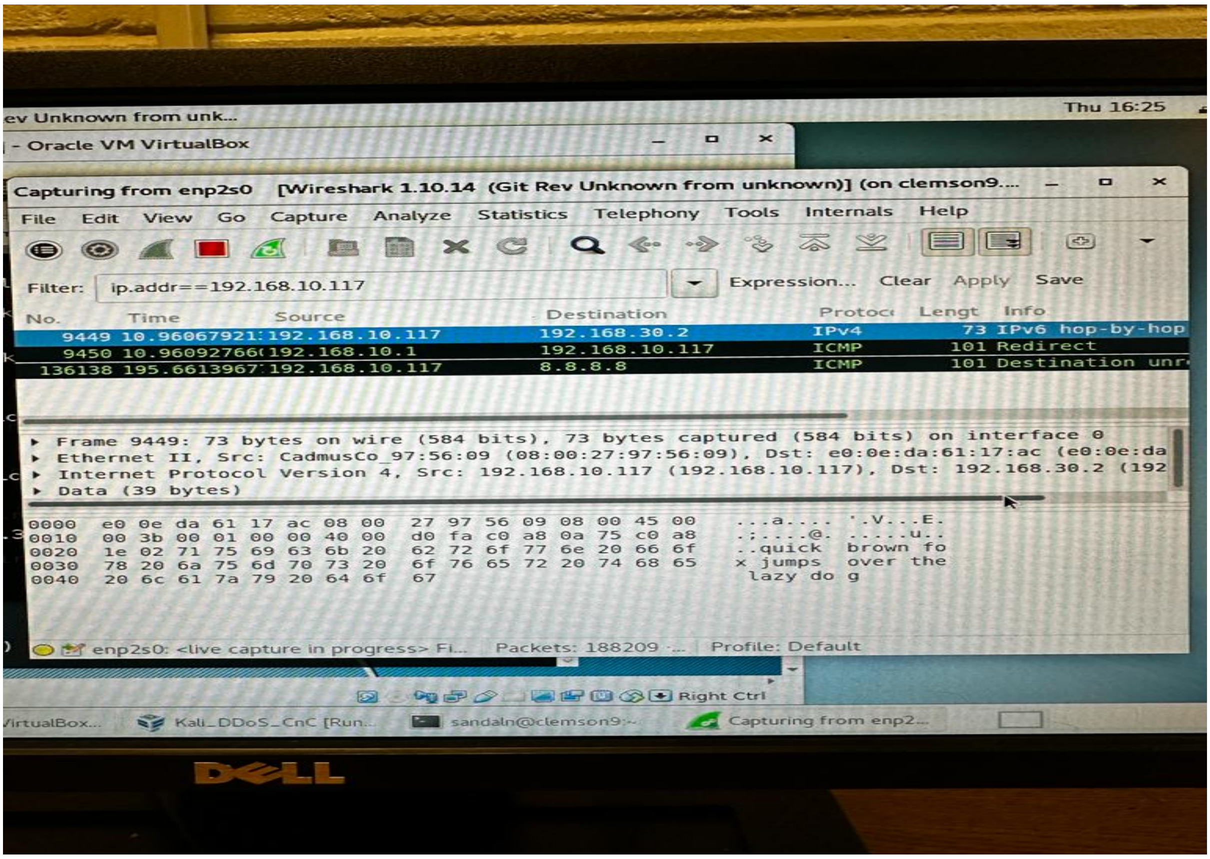
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard ev

The Virtual Machine reports that the guest OS supports mouse pointer integration. This means that you do not need to capture the

```
root@cnc:~# scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> send(IP(dst='192.168.30.2,src='192.168.10.117')/'quick brown fox jumps over
the lazy dog')
  File "<console>", line 1
    send(IP(dst='192.168.30.2,src='192.168.10.117')/'quick brown fox jumps over
the lazy dog')
                                          ^
SyntaxError: invalid syntax
>>> send(IP(dst='192.168.30.2',src='192.168.10.117')/'quick brown fox jumps over
 the lazy dog')
.
Sent 1 packets.
>>> 
```

```
MAC Address: E4:54:E8:D0:10:1D (Unknown)
Nmap scan report for 192.168.10.103
MAC Address: 08:00:27:F6:6A:A8 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.116
Host is up (0.00028s latency).
MAC Address: 08:00:27:8D:6C:59 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.117
Host is up (0.00028s latency).
MAC Address: 08:00:27:38:DF:C7 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.139
Host is up (0.00022s latency).
MAC Address: 08:00:27:97:56:09 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.200
Host is up (0.0071s latency).
MAC Address: 78:D2:94:53:1A:B6 (Unknown)
Nmap scan report for 192.168.10.111
Host is up
```

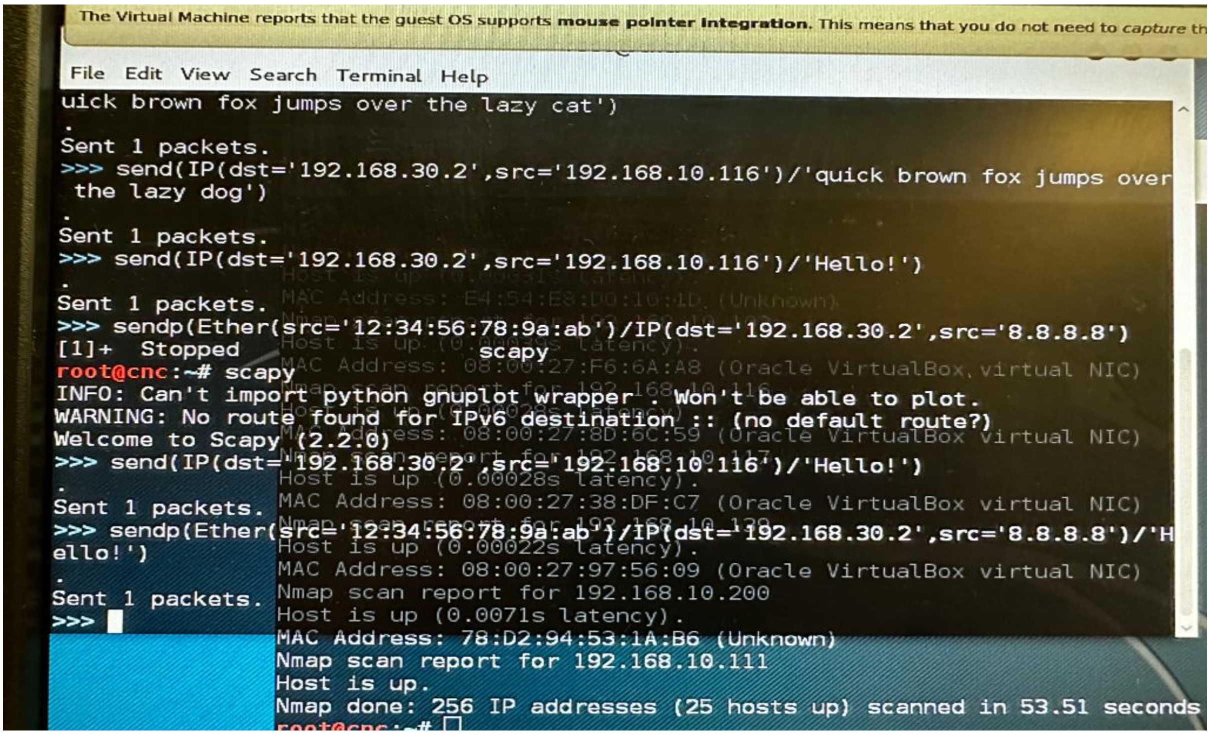
Fig:1-DNS requests with subnet IP address and payload message for destination IP



ev Unknown from unk...
Thu 16:25
- Oracle VM VirtualBox
Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown from unknown)] (on clemson9....
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save
Right Ctrl
VirtualBox...
Kali_DDoS_CnC [Run...
sandaln@clemson9:~
Capturing from enp2...

```
Filter: ip.addr==192.168.10.117

No.     Time              Source            Destination       Protoc  Lengt Info
  9449 10.96067921:192.168.10.117      192.168.30.2       IPv4        73 IPv6 hop-by-hop
  9450 10.96092766(192.168.10.1        192.168.10.117     ICMP       101 Redirect
136138 195.6613967:192.168.10.117      8.8.8.8            ICMP       101 Destination unr

▸ Frame 9449: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
▸ Ethernet II, Src: CadmusCo_97:56:09 (08:00:27:97:56:09), Dst: e0:0e:da:61:17:ac (e0:0e:da
▸ Internet Protocol Version 4, Src: 192.168.10.117 (192.168.10.117), Dst: 192.168.30.2 (192
▸ Data (39 bytes)

0000  e0 0e da 61 17 ac 08 00  27 97 56 09 08 00 45 00   ...a.... '.V...E.
0010  00 3b 00 01 00 00 40 00  d0 fa c0 a8 0a 75 c0 a8   .;....@. .....u..
0020  1e 02 71 75 69 63 6b 20  62 72 6f 77 6e 20 66 6f   ..quick  brown fo
0030  78 20 6a 75 6d 70 73 20  6f 76 65 72 20 74 68 65   x jumps  over the
0040  20 6c 61 7a 79 20 64 6f  67                         lazy do g

enp2s0: <live capture in progress> Fi...   Packets: 188209 ...   Profile: Default
```

Fig:2- Payload message for IP address 192.168.10.117

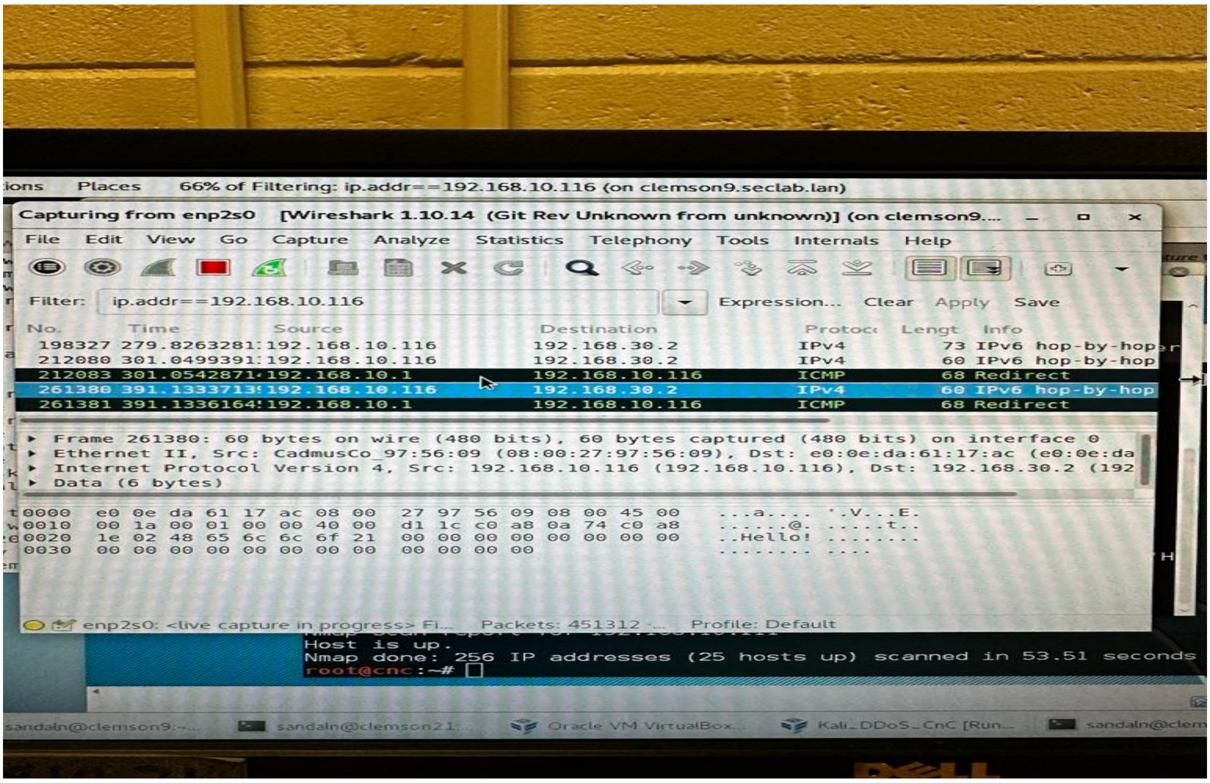


The Virtual Machine reports that the guest OS supports mouse pointer integration. This means that you do not need to capture th
File Edit View Search Terminal Help

```
uick brown fox jumps over the lazy cat')
.
Sent 1 packets.
>>> send(IP(dst='192.168.30.2',src='192.168.10.116')/'quick brown fox jumps over
 the lazy dog')
.
Sent 1 packets.
>>> send(IP(dst='192.168.30.2',src='192.168.10.116')/'Hello!')
.
Sent 1 packets.
>>> sendp(Ether(src='12:34:56:78:9a:ab')/IP(dst='192.168.30.2',src='8.8.8.8')
[1]+  Stopped                 scapy
root@cnc:~# scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> send(IP(dst='192.168.30.2',src='192.168.10.116')/'Hello!')
.
Sent 1 packets.
>>> sendp(Ether(src='12:34:56:78:9a:ab')/IP(dst='192.168.30.2',src='8.8.8.8')/'H
ello!')
.
Sent 1 packets.
>>> █
```

```
MAC Address: E4:54:E8:D0:10:1D (Unknown)
MAC Address: 08:00:27:F6:6A:A8 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.116
MAC Address: 08:00:27:8D:6C:59 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.117
Host is up (0.00028s latency).
MAC Address: 08:00:27:38:DF:C7 (Oracle VirtualBox virtual NIC)
Host is up (0.00022s latency).
MAC Address: 08:00:27:97:56:09 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.200
Host is up (0.0071s latency).
MAC Address: 78:D2:94:53:1A:B6 (Unknown)
Nmap scan report for 192.168.10.111
Host is up.
Nmap done: 256 IP addresses (25 hosts up) scanned in 53.51 seconds
root@cnc:~# □
```

Fig:3- DNS requests with several subnet IP addresses and various payload messages for every destination IP



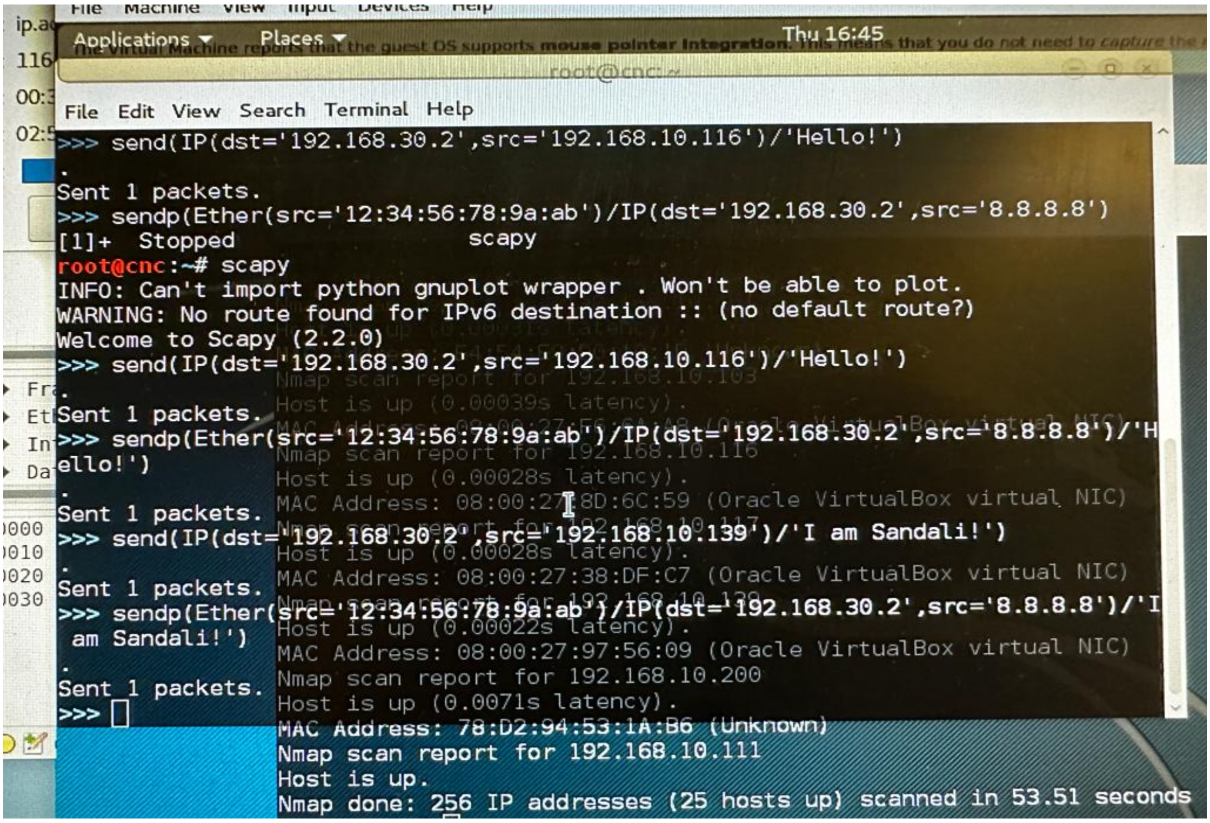
Filter: ip.addr==192.168.10.116

```
No.     Time         Source            Destination       Protoc  Lengt  Info
198327  279.8263281  192.168.10.116    192.168.30.2      IPv4    73     IPv6 hop-by-hop
212080  301.0499391  192.168.10.116    192.168.30.2      IPv4    60     IPv6 hop-by-hop
212083  301.0542871  192.168.10.1      192.168.10.116    ICMP    68     Redirect
261380  391.1333713  192.168.10.116    192.168.30.2      IPv4    60     IPv6 hop-by-hop
261381  391.1336164  192.168.10.1      192.168.10.116    ICMP    68     Redirect

▸ Frame 261380: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸ Ethernet II, Src: CadmusCo_97:56:09 (08:00:27:97:56:09), Dst: e0:0e:da:61:17:ac (e0:0e:da
▸ Internet Protocol Version 4, Src: 192.168.10.116 (192.168.10.116), Dst: 192.168.30.2 (192
▸ Data (6 bytes)

0000  e0 0e da 61 17 ac 08 00  27 97 56 09 08 00 45 00   ...a.... '.V...E.
0010  00 1a 00 01 00 00 40 00  d1 1c c0 a8 0a 74 c0 a8   ......@. .....t..
0020  1e 02 48 65 6c 6c 6f 21  00 00 00 00 00 00 00 00   ..Hello! ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

enp2s0: <live capture in progress> Fi... Packets: 451312 ... Profile: Default

```
Host is up.
Nmap done: 256 IP addresses (25 hosts up) scanned in 53.51 seconds
root@cnc:~# 
```

Fig:4- Payload message for IP address 192.168.10.116

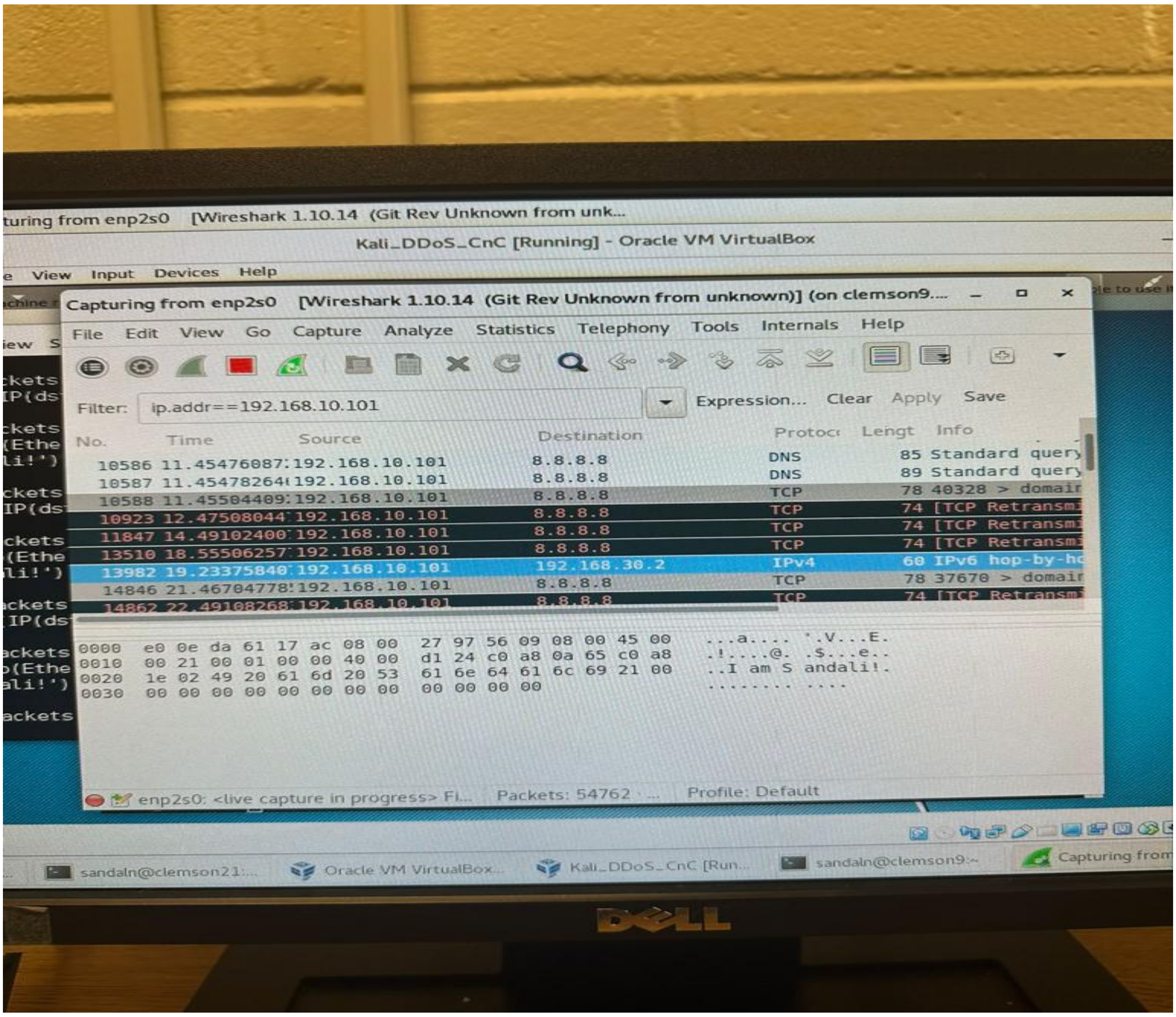


File
Machine
View
Input
Devices
Help
Applications
Places
Thu 16:45
The Virtual Machine reports that the guest OS supports mouse pointer integration. This means that you do not need to capture the
root@cnc:~
File
Edit
View
Search
Terminal
Help

```
>>> send(IP(dst='192.168.30.2',src='192.168.10.116')/'Hello!')
.
Sent 1 packets.
>>> sendp(Ether(src='12:34:56:78:9a:ab')/IP(dst='192.168.30.2',src='8.8.8.8')
[1]+  Stopped                 scapy
root@cnc:~# scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> send(IP(dst='192.168.30.2',src='192.168.10.116')/'Hello!')
.
Sent 1 packets.
>>> sendp(Ether(src='12:34:56:78:9a:ab')/IP(dst='192.168.30.2',src='8.8.8.8')/'H
ello!')
.
Sent 1 packets.
>>> send(IP(dst='192.168.30.2',src='192.168.10.139')/'I am Sandali!')
.
Sent 1 packets.
>>> sendp(Ether(src='12:34:56:78:9a:ab')/IP(dst='192.168.30.2',src='8.8.8.8')/'I
 am Sandali!')
.
Sent 1 packets.
>>> 
```

```
Nmap scan report for 192.168.10.103
Host is up (0.00039s latency).
Nmap scan report for 192.168.10.116
Host is up (0.00028s latency).
MAC Address: 08:00:27:8D:6C:59 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.117
Host is up (0.00028s latency).
MAC Address: 08:00:27:38:DF:C7 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.139
Host is up (0.00022s latency).
MAC Address: 08:00:27:97:56:09 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.200
Host is up (0.0071s latency).
MAC Address: 78:D2:94:53:1A:B6 (Unknown)
Nmap scan report for 192.168.10.111
Host is up.
Nmap done: 256 IP addresses (25 hosts up) scanned in 53.51 seconds
```

Fig:5- DNS requests with several subnet IP addresses and various payload messages for every destination IP



Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown from unknown)] (on clemson9....
Kali_DDoS_CnC [Running] - Oracle VM VirtualBox
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save
enp2s0: <live capture in progress> Fi... Packets: 54762 · ... Profile: Default

Filter: ip.addr==192.168.10.101

```
No.   Time        Source            Destination     Protoc  Lengt Info
10586 11.45476087 192.168.10.101    8.8.8.8         DNS     85 Standard query
10587 11.45478264 192.168.10.101    8.8.8.8         DNS     89 Standard query
10588 11.45504409 192.168.10.101    8.8.8.8         TCP     78 40328 > domain
10923 12.47508044 192.168.10.101    8.8.8.8         TCP     74 [TCP Retransmi
11847 14.49102400 192.168.10.101    8.8.8.8         TCP     74 [TCP Retransmi
13510 18.55506257 192.168.10.101    8.8.8.8         TCP     74 [TCP Retransmi
13982 19.23375840 192.168.10.101    192.168.30.2    IPv4    60 IPv6 hop-by-hc
14846 21.46704778 192.168.10.101    8.8.8.8         TCP     78 37670 > domair
14862 22.49108268 192.168.10.101    8.8.8.8         TCP     74 [TCP Retransmi

0000  e0 0e da 61 17 ac 08 00  27 97 56 09 08 00 45 00   ...a.... '.V...E.
0010  00 21 00 01 00 00 40 00  d1 24 c0 a8 0a 65 c0 a8   .!....@. .$...e..
0020  1e 02 49 20 61 6d 20 53  61 6e 64 61 6c 69 21 00   ..I am S andali!.
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

Fig:6- Payload message for IP address 192.168.10.101

Conclusion:

3 IP addresses in my subnet were spoofed for the experiment. There are different messages sent to each of those 3 IP addresses.

Questions:

1) How can you evade detection when spoofing others? Is it possible to ascertain the identity of the sender?
   You can evade detection while spoofing others by passing a duplicate packet which can make it easy to surpass the security safeguards. The other way is to encrypt the packets. Yes, it is possible to ascertain the identity of the sender. It can be done by filtering out the malicious traffic and working with other ISPs. You can identify if an attack is spoofed, or if it originated from another provider or a third party.
   There are 2 ways to do it:
   Router Entropy: 1) Determine the entropy of each downstream router to identify potential attack flows if an attack is present in the receiver proxy server.

2) Suspect those routers whose NE rate is below a certain threshold are assault routers.
3) Next, until we find the attack's origin, figure out the NE rate for each attack router's neighbour routers.

IP Trackback:

1) If an attack is taking place, you should first recognize the packets, find out their source IP address and mark value, and get in touch with the sender who is delivering those packets to the recipient.
2) The intermediate router compares such digest values to its digest table eateries to determine the IP address of a certain sender router.
3) These steps will be carried out until the attack's origin is discovered.

2) Explain what IP address spoofing is, and what a host on the network must do to spoof its IP address.

   To conceal the sender's identity, impersonate another computer system, or both, IP spoofing involves creating Internet Protocol (IP) packets with modified source addresses. Bad actors frequently employ this technique to launch DDoS attacks against a target device or the local infrastructure. To spoof an IP address the host must find out the Mac and IP address by using Nmap and the use scapy to send packets to the address. Last the host must change the IP address in the header to match the spoofed address.

3) Explain why an attacker cannot just grab any existing IP packet carrying UDP or TCP, change only the IP addresses in there, and expect the target host to accept the packet. Especially for TCP, you don’t have to read the entire RFC but focus on the header.

   UDP-The User Datagram Protocol is one of the core communication protocols of the Internet protocol suite used to send messages to other hosts on an Internet Protocol network.

   TCP- A network conversation that allows programs to exchange data is defined by the Transmission Control Protocol standard. TCP interacts with the internet protocol, which establishes rules for how computers exchange data packets.

   The reason an attacker can't just grab any UDP/TCP-carrying packet that already exists and modify only the IP addresses in there and expect the target host to accept the packet is as follows:

   The first byte of data in a transmitted TCP packet serves as the sequence number in the Transmission Control Protocol. The sequence number is the acknowledgment number. The order of the packets is tracked using these sequence numbers, which also allow for the detection of missing packets. These packets will be dropped if the sequence numbers are not present.Even while TCP is more secure than UDP, it occasionally uses authentication, so the attacker would need to present their credentials in order to proceed.

Comments:

The lectures and the lab sessions were equally informative, and I got to learn how the attacks take place. I also understood the vital role security plays through the course. The execution of lab experiments was successfully completed as I learnt to apply the concepts.

Bibliography:

Distributed Denial of Service Attacks, by I. Ozcelik, and R R. Brooks, CRC Press.

References:

https://research.ijcaonline.org/volume42/number1/pxc3877549.pdf

https://www.kaspersky.com/resource-center/threats/ip-spoofing

https://www.cloudflare.com/en-gb/learning/ddos/glossary/ip-spoofing/

https://www.techtarget.com/searchsecurity/feature/The-five-different-types-of-firewalls

https://www.rapid7.com/fundamentals/spoofing-attacks/