PRANITA CAMASAMUDRAM

LAB REPORT 2
Network Background traffic

INTRODUCTION: This lab introduces us to the network background analysis during a ddos attack. We are taught how to generate the simulated traffic for the detection of ddos attacks.

3. What kind of traffic did you run between hosts in the second scenario (Replay a pcap file with tcpreplay). Justify why you think it can represent operational network traffic.

Ans. The traffic that was run between hosts during replay is the simulated traffic. This because simulated traffic gives a really good basis for analyzing the ddos attack patterns. It is the best representation of the operational traffic and as all the networks used are working and is the best reflection of actual traffic.

4. You can replay and forward captured pcap files on a link in a controlled network environment to use as background traffic. If you perform a DDoS attack on this link, you can observe the effects of the DDoS attack without jeopardizing the operational network. However, some of the effects cannot be observed with replayed/forwarded background traffic. List some of these effects and explain the reason why they can not be observed.

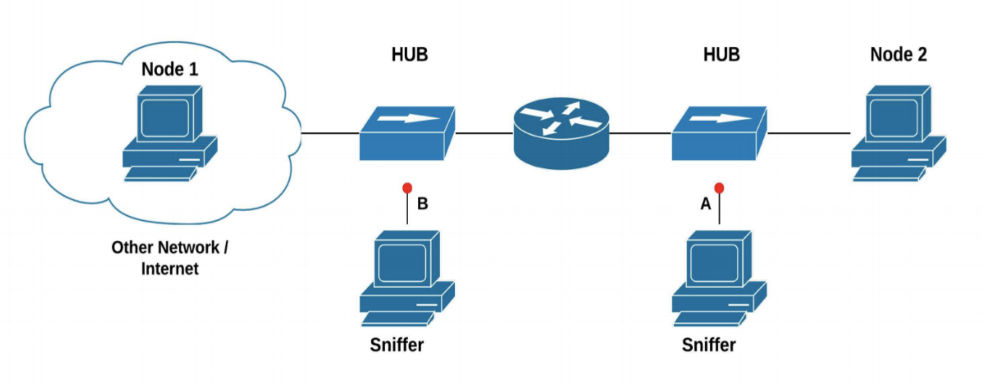
Ans. SYN attack is one that cannot be observed as it send an overwhelming amount of requests to the user. At that point it is blindly transmitting and no response would be received i.e, from transport layer to internet layer and then from

5. Number of packets received by a node on the network is one of the popular metrics used in DDoS detection applications. In this assignment, we focused on packet count statistics. Discuss what other metrics that can be used for DDoS detection.

Ans. Few of the metrics are: continuous ranked probability score (CRPS) statistical metric , exponentially smoothing (ES) scheme. They are used to quantify the differences between normal of the traffic and any new observations. The other two metrics used are Hartley entropy and Shannon entropy.

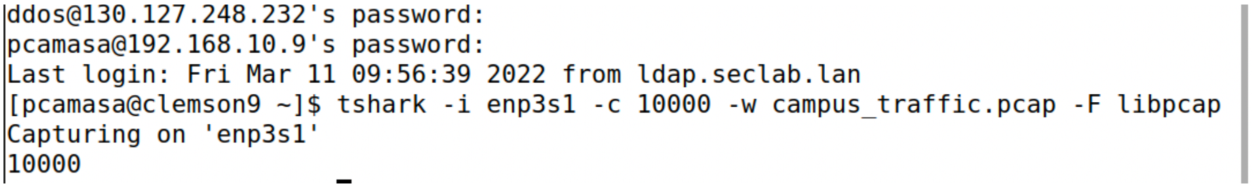
LAB SETUP:

The below picture shows the setup of the lab machines. The node 1 above is the machine in clemson with the IP address 192.168.10.10 and node 2 is supposed to be the Charleston computer but due to technical difficulties we are using the virtual machine 192.168.30.2 which acts similar to the above. The spoofing machine is used to replay the traffic and simulate. It's IP address is 192.168.10.111.



First I logged into the sniffer machine with the help of ssh and then used the below command to capture the clemson network traffic

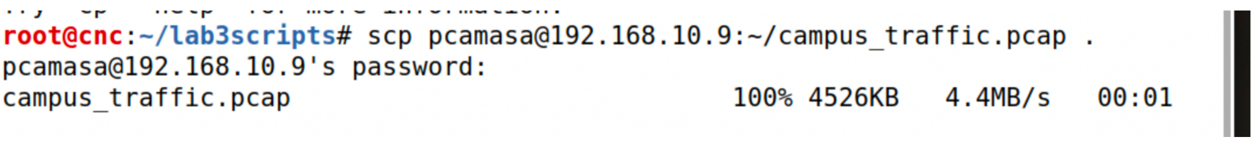
$ tshark -i enp3s1 -c 10000 -w campus_traffic.pcap -F libpcap



```
ddos@130.127.248.232's password:
pcamasa@192.168.10.9's password:
Last login: Fri Mar 11 09:56:39 2022 from ldap.seclab.lan
[pcamasa@clemson9 ~]$ tshark -i enp3s1 -c 10000 -w campus_traffic.pcap -F libpcap
Capturing on 'enp3s1'
10000
```

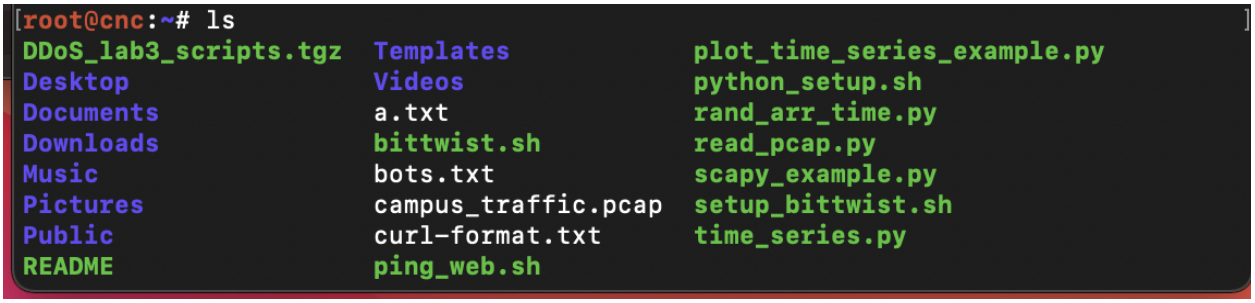
In the above screenshot I captured 10000 packets and interface enp3s1 is used. The captured pcap file is named campus_traffic.pcap.

I then copied the captured file to the spoofing machine as below:



```
Try 'cp --help' for more information.
root@cnc:~/lab3scripts# scp pcamasa@192.168.10.9:~/campus_traffic.pcap .
pcamasa@192.168.10.9's password:
campus_traffic.pcap                                  100% 4526KB   4.4MB/s   00:01
```

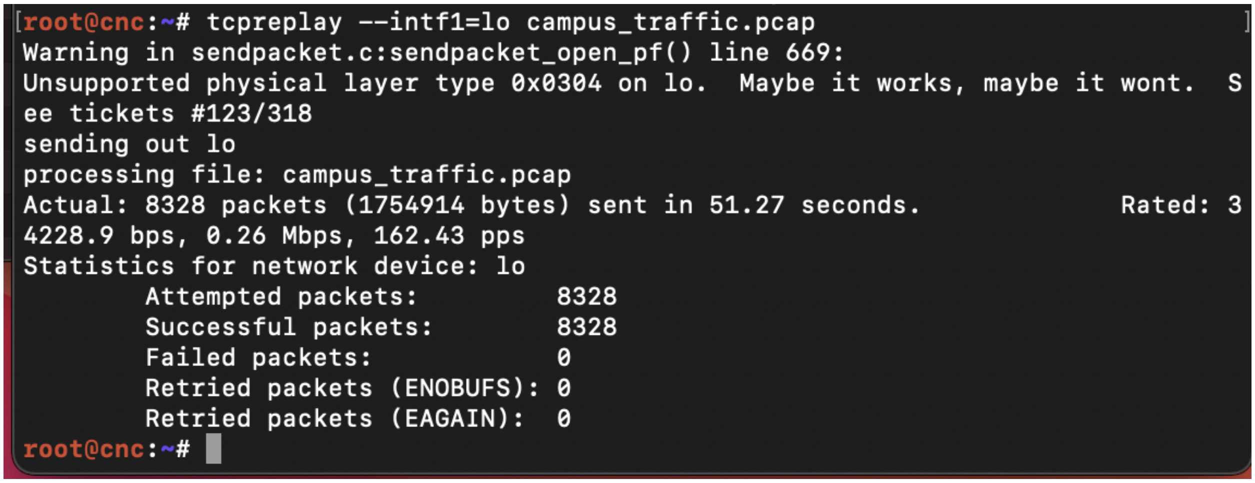
This is because the spoof machine contains the time series graph. Now when we ls in the machine we can see the following files below. The



```
[root@cnc:~# ls                                                                    ]
DDoS_lab3_scripts.tgz  Templates            plot_time_series_example.py
Desktop                Videos               python_setup.sh
Documents              a.txt                rand_arr_time.py
Downloads              bittwist.sh          read_pcap.py
Music                  bots.txt             scapy_example.py
Pictures               campus_traffic.pcap  setup_bittwist.sh
Public                 curl-format.txt      time_series.py
README                 ping_web.sh
```

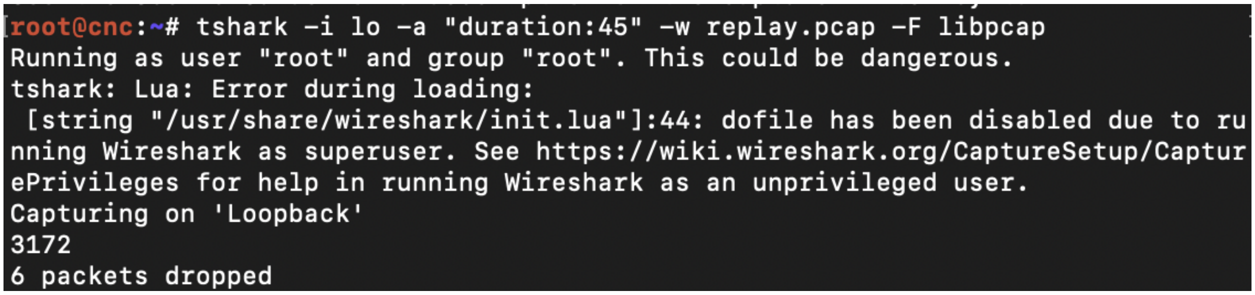
First I started capturing the data in the spoof machine. Before this I also captured the 45 second duration packets on sniffer which is why the below result is 8328.

#tcpreplay --intf1=lo campus_traffic.pcap



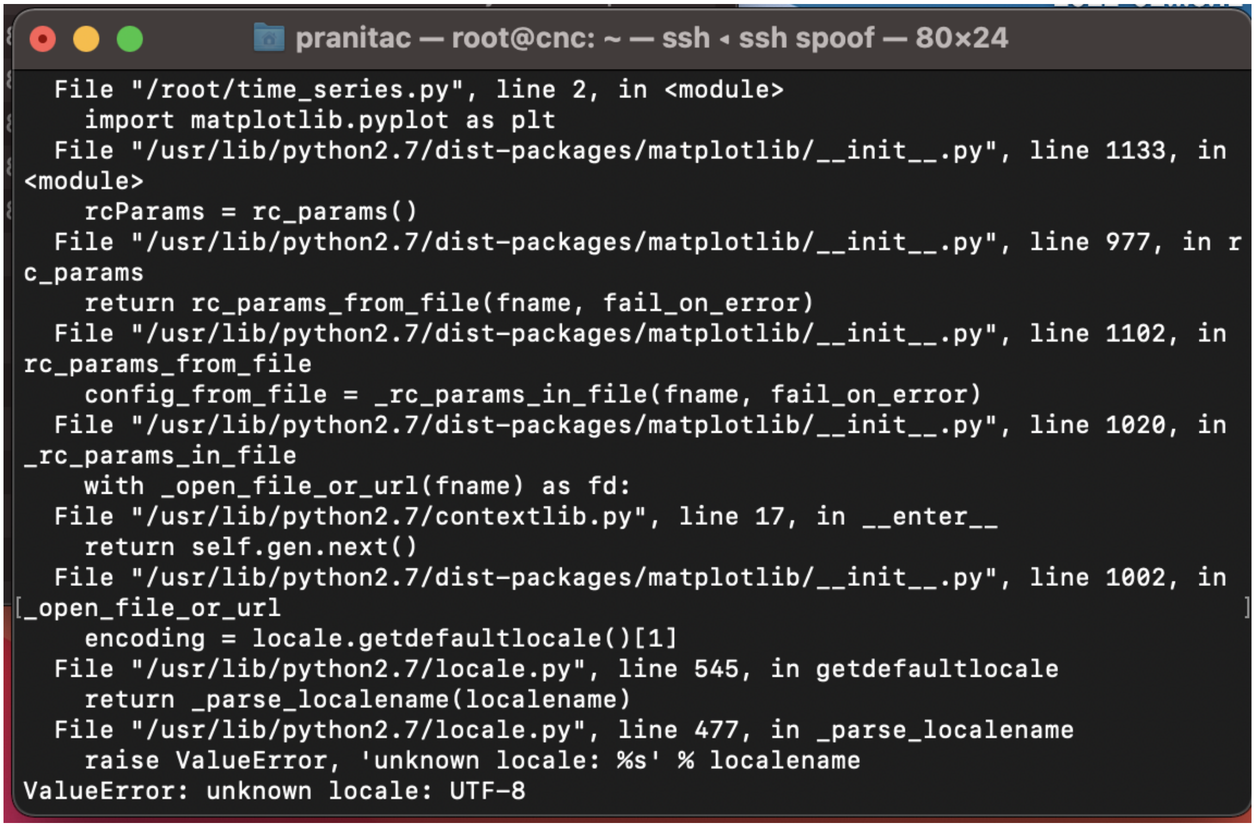
```
[root@cnc:~# tcpreplay --intf1=lo campus_traffic.pcap                              ]
Warning in sendpacket.c:sendpacket_open_pf() line 669:
Unsupported physical layer type 0x0304 on lo.  Maybe it works, maybe it wont.  S
ee tickets #123/318
sending out lo
processing file: campus_traffic.pcap
Actual: 8328 packets (1754914 bytes) sent in 51.27 seconds.          Rated: 3
4228.9 bps, 0.26 Mbps, 162.43 pps
Statistics for network device: lo
        Attempted packets:         8328
        Successful packets:        8328
        Failed packets:            0
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
root@cnc:~# 
```

Now I tried replaying the traffic using the below command. It clearly shows that few packets have been dropped.



```
root@cnc:~# tshark -i lo -a "duration:45" -w replay.pcap -F libpcap
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 [string "/usr/share/wireshark/init.lua"]:44: dofile has been disabled due to ru
nning Wireshark as superuser. See https://wiki.wireshark.org/CaptureSetup/Captur
ePrivileges for help in running Wireshark as an unprivileged user.
Capturing on 'Loopback'
3172
6 packets dropped
```

I have tried producing the time series graph for the above but it keeps giving me the below error message. I have explained this in detail in my email.



```
  File "/root/time_series.py", line 2, in <module>
    import matplotlib.pyplot as plt
  File "/usr/lib/python2.7/dist-packages/matplotlib/__init__.py", line 1133, in
<module>
    rcParams = rc_params()
  File "/usr/lib/python2.7/dist-packages/matplotlib/__init__.py", line 977, in r
c_params
    return rc_params_from_file(fname, fail_on_error)
  File "/usr/lib/python2.7/dist-packages/matplotlib/__init__.py", line 1102, in
rc_params_from_file
    config_from_file = _rc_params_in_file(fname, fail_on_error)
  File "/usr/lib/python2.7/dist-packages/matplotlib/__init__.py", line 1020, in
_rc_params_in_file
    with _open_file_or_url(fname) as fd:
  File "/usr/lib/python2.7/contextlib.py", line 17, in __enter__
    return self.gen.next()
  File "/usr/lib/python2.7/dist-packages/matplotlib/__init__.py", line 1002, in
_open_file_or_url
    encoding = locale.getdefaultlocale()[1]
  File "/usr/lib/python2.7/locale.py", line 545, in getdefaultlocale
    return _parse_localename(localename)
  File "/usr/lib/python2.7/locale.py", line 477, in _parse_localename
    raise ValueError, 'unknown locale: %s' % localename
ValueError: unknown locale: UTF-8
```

I have tried every possible troubleshooting but it does not work for me.