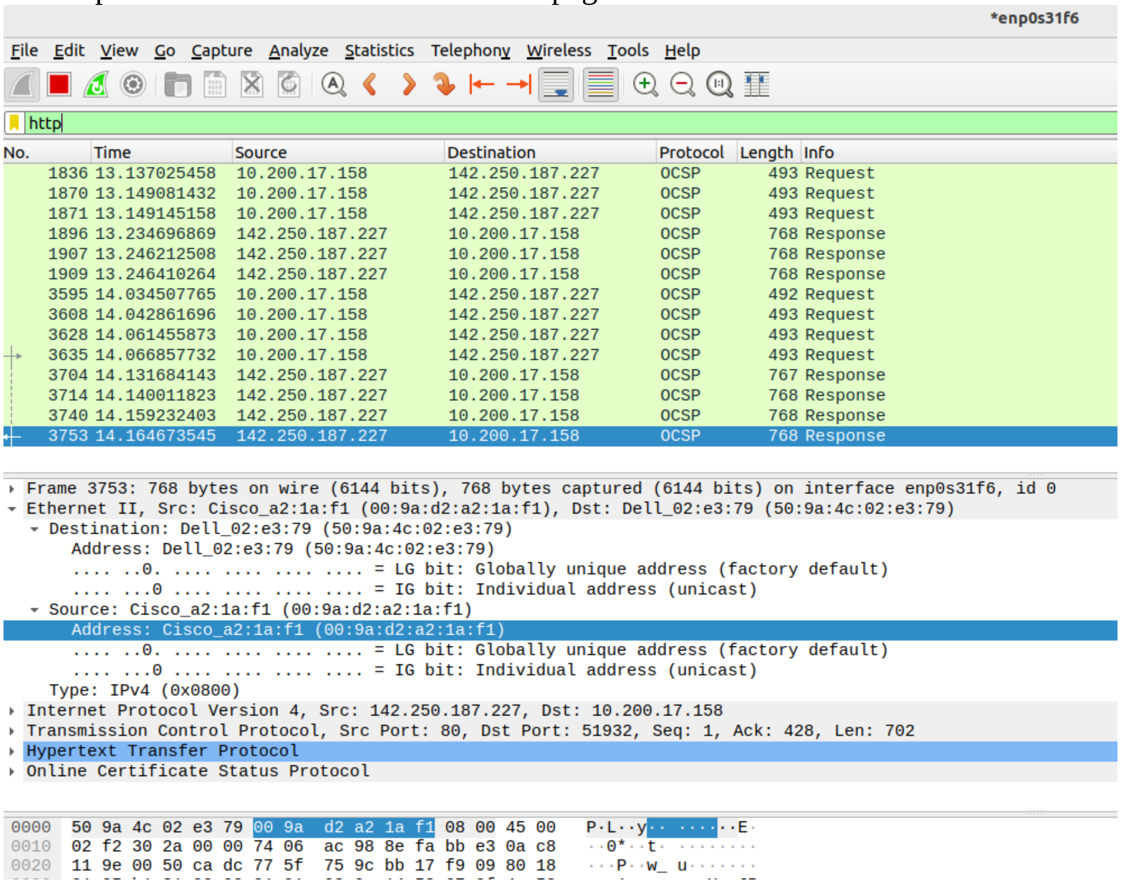
Exercise 1: Traffic Capture

Packets are initially transmitted slowly but when the rate of transmission was increased, packets can be observed. ICMP (internet control message protocol) data packets appear to be used for echo request/reply. ARP (address resolution protocol) packets. MDNS (multicast domain name system) packets for standard query. ICMPv6 (internet control message protocol version 6) packet used for router solicitation, identify other routers on the network.

To capture http traffic I needed to search for web pages in the web browser

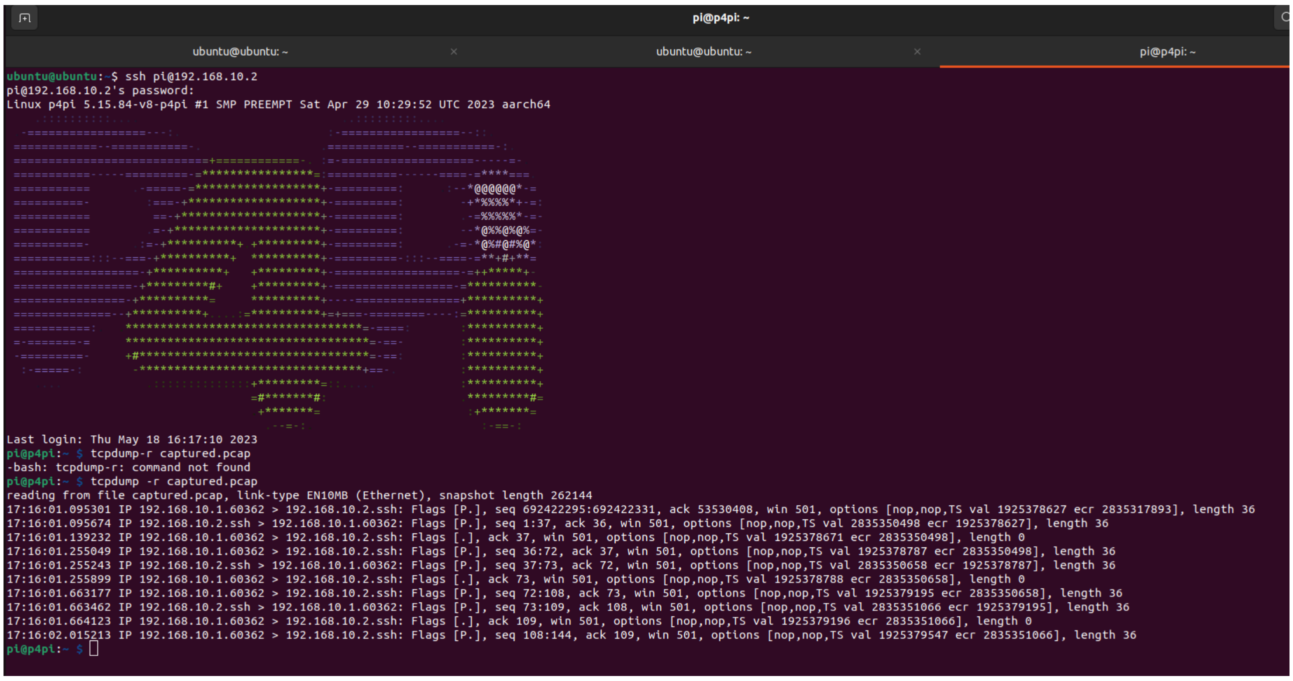

*enp0s31f6
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.    Time          Source          Destination      Protocol Length Info
  1836 13.137025458  10.200.17.158   142.250.187.227  OCSP      493 Request
  1870 13.149081432  10.200.17.158   142.250.187.227  OCSP      493 Request
  1871 13.149145158  10.200.17.158   142.250.187.227  OCSP      493 Request
  1896 13.234696869  142.250.187.227 10.200.17.158    OCSP      768 Response
  1907 13.246212508  142.250.187.227 10.200.17.158    OCSP      768 Response
  1909 13.246410264  142.250.187.227 10.200.17.158    OCSP      768 Response
  3595 14.034507765  10.200.17.158   142.250.187.227  OCSP      492 Request
  3608 14.042861696  10.200.17.158   142.250.187.227  OCSP      493 Request
  3628 14.061455873  10.200.17.158   142.250.187.227  OCSP      493 Request
  3635 14.066857732  10.200.17.158   142.250.187.227  OCSP      493 Request
  3704 14.131684143  142.250.187.227 10.200.17.158    OCSP      767 Response
  3714 14.140011823  142.250.187.227 10.200.17.158    OCSP      768 Response
  3740 14.159232403  142.250.187.227 10.200.17.158    OCSP      768 Response
  3753 14.164673545  142.250.187.227 10.200.17.158    OCSP      768 Response
```

```
Frame 3753: 768 bytes on wire (6144 bits), 768 bytes captured (6144 bits) on interface enp0s31f6, id 0
Ethernet II, Src: Cisco_a2:1a:f1 (00:9a:d2:a2:1a:f1), Dst: Dell_02:e3:79 (50:9a:4c:02:e3:79)
  Destination: Dell_02:e3:79 (50:9a:4c:02:e3:79)
    Address: Dell_02:e3:79 (50:9a:4c:02:e3:79)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: Cisco_a2:1a:f1 (00:9a:d2:a2:1a:f1)
    Address: Cisco_a2:1a:f1 (00:9a:d2:a2:1a:f1)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 142.250.187.227, Dst: 10.200.17.158
Transmission Control Protocol, Src Port: 80, Dst Port: 51932, Seq: 1, Ack: 428, Len: 702
Hypertext Transfer Protocol
Online Certificate Status Protocol
```

```
0000  50 9a 4c 02 e3 79 00 9a  d2 a2 1a f1 08 00 45 00   P·L··y·· ······E·
0010  02 f2 30 2a 00 00 74 06  ac 98 8e fa bb e3 0a c8   ··0*··t· ········
0020  11 9e 00 50 ca dc 77 5f  75 9c bb 17 f9 09 80 18   ···P··w_ u·······
```

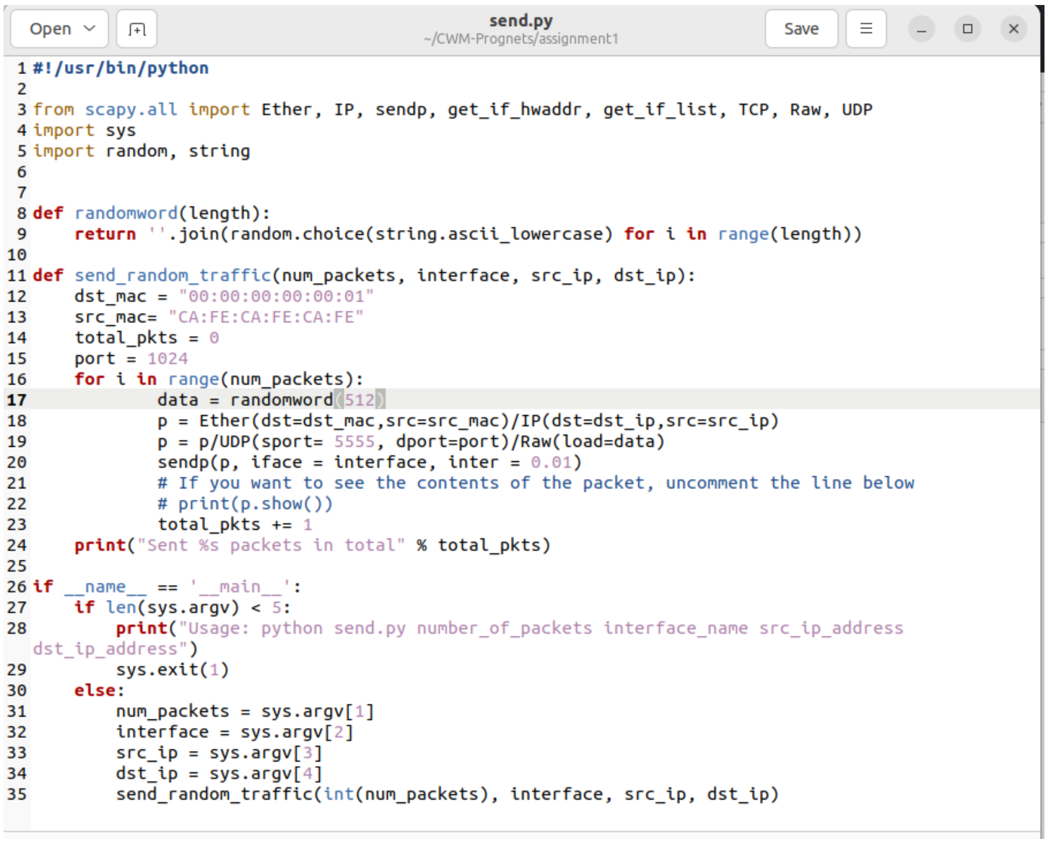
/home/ubuntu/Pictures/Screenshots/Screenshot from 2023-05-22 11-30-29.png



```
ubuntu@ubuntu:~$ ssh pi@192.168.10.2
pi@192.168.10.2's password:
Linux p4pi 5.15.84-v8-p4pi #1 SMP PREEMPT Sat Apr 29 10:29:52 UTC 2023 aarch64
    .:::::::::::....                                  ..:::::::::::...
 .-==================---:.                              :-==================--::.
 ============--============-.                          .============--============-:.
 ============================+============-.  :=-=======================-----=-.
 ===========----==========-=*****************=:===========-----=====-=****===.
 ===========        .-=====-=******************+-==========:        :--*@@@@@@*-=
 ==========-          :===-+*******************+-==========:          -+*%%%%*+-=:
 ===========           ==-+********************+-==========:          .-=%%%%%*-=-
 ===========          .=-+*********************+-==========:           --*@%%@%@%=-
 ==========-         .:=-+**********+.+**********+-==========:        .-=-*@%#@#%@*:
 ===========:::--===-+**********+   **********+-==========-:::--====-=**+#+**=
 ==================-+**********+   +*********+-====================-=++*****+-
 =================-+*********#+    +*********+-====================-=**********-
 ================-+**********=     **********+----================+**********+
 ==============--+**********+....:=**********+=+===-========----:=**********+
 ===========:.   .*********************************=-====:         :**********+
 =-=======-=      **********************************=-==-          :**********+
 -=========-      +#*********************************=-==:          :**********+
  :-=====-:        -*********************************+==-.          :**********+
     ....          .:::::::::::::::+*********=::.....               :**********+
                                   =#*******#:                      .*********#=
                                   .+*******=                        :+*******=

                                     .--=-:.                          :-==-:
Last login: Thu May 18 16:17:10 2023
pi@p4pi:~ $ tcpdump-r captured.pcap
-bash: tcpdump-r: command not found
pi@p4pi:~ $ tcpdump -r captured.pcap
reading from file captured.pcap, link-type EN10MB (Ethernet), snapshot length 262144
17:16:01.095301 IP 192.168.10.1.60362 > 192.168.10.2.ssh: Flags [P.], seq 692422295:692422331, ack 53530408, win 501, options [nop,nop,TS val 1925378627 ecr 2835317893], length 36
17:16:01.095674 IP 192.168.10.2.ssh > 192.168.10.1.60362: Flags [P.], seq 1:37, ack 36, win 501, options [nop,nop,TS val 2835350498 ecr 1925378627], length 36
17:16:01.139232 IP 192.168.10.1.60362 > 192.168.10.2.ssh: Flags [.], ack 37, win 501, options [nop,nop,TS val 1925378671 ecr 2835350498], length 0
17:16:01.255049 IP 192.168.10.1.60362 > 192.168.10.2.ssh: Flags [P.], seq 36:72, ack 37, win 501, options [nop,nop,TS val 1925378787 ecr 2835350498], length 36
17:16:01.255243 IP 192.168.10.2.ssh > 192.168.10.1.60362: Flags [P.], seq 37:73, ack 72, win 501, options [nop,nop,TS val 2835350658 ecr 1925378787], length 36
17:16:01.255899 IP 192.168.10.1.60362 > 192.168.10.2.ssh: Flags [.], ack 73, win 501, options [nop,nop,TS val 1925378788 ecr 2835350658], length 0
17:16:01.663177 IP 192.168.10.1.60362 > 192.168.10.2.ssh: Flags [P.], seq 72:108, ack 73, win 501, options [nop,nop,TS val 1925379195 ecr 2835350658], length 36
17:16:01.663462 IP 192.168.10.2.ssh > 192.168.10.1.60362: Flags [P.], seq 73:109, ack 108, win 501, options [nop,nop,TS val 2835351066 ecr 1925379195], length 36
17:16:01.664123 IP 192.168.10.1.60362 > 192.168.10.2.ssh: Flags [.], ack 109, win 501, options [nop,nop,TS val 1925379196 ecr 2835351066], length 0
17:16:02.015213 IP 192.168.10.1.60362 > 192.168.10.2.ssh: Flags [P.], seq 108:144, ack 109, win 501, options [nop,nop,TS val 1925379547 ecr 2835351066], length 36
pi@p4pi:~ $ 
```

To capture the 100 packets sent by my command I implement the udp filter
Packet size: frame length = 64 bytes; data = 22 bytes
Protocol used: UDP



```python
#!/usr/bin/python

from scapy.all import Ether, IP, sendp, get_if_hwaddr, get_if_list, TCP, Raw, UDP
import sys
import random, string


def randomword(length):
    return ''.join(random.choice(string.ascii_lowercase) for i in range(length))

def send_random_traffic(num_packets, interface, src_ip, dst_ip):
    dst_mac = "00:00:00:00:00:01"
    src_mac= "CA:FE:CA:FE:CA:FE"
    total_pkts = 0
    port = 1024
    for i in range(num_packets):
            data = randomword(512)
            p = Ether(dst=dst_mac,src=src_mac)/IP(dst=dst_ip,src=src_ip)
            p = p/UDP(sport= 5555, dport=port)/Raw(load=data)
            sendp(p, iface = interface, inter = 0.01)
            # If you want to see the contents of the packet, uncomment the line below
            # print(p.show())
            total_pkts += 1
    print("Sent %s packets in total" % total_pkts)

if __name__ == '__main__':
    if len(sys.argv) < 5:
        print("Usage: python send.py number_of_packets interface_name src_ip_address dst_ip_address")
        sys.exit(1)
    else:
        num_packets = sys.argv[1]
        interface = sys.argv[2]
        src_ip = sys.argv[3]
        dst_ip = sys.argv[4]
        send_random_traffic(int(num_packets), interface, src_ip, dst_ip)
```