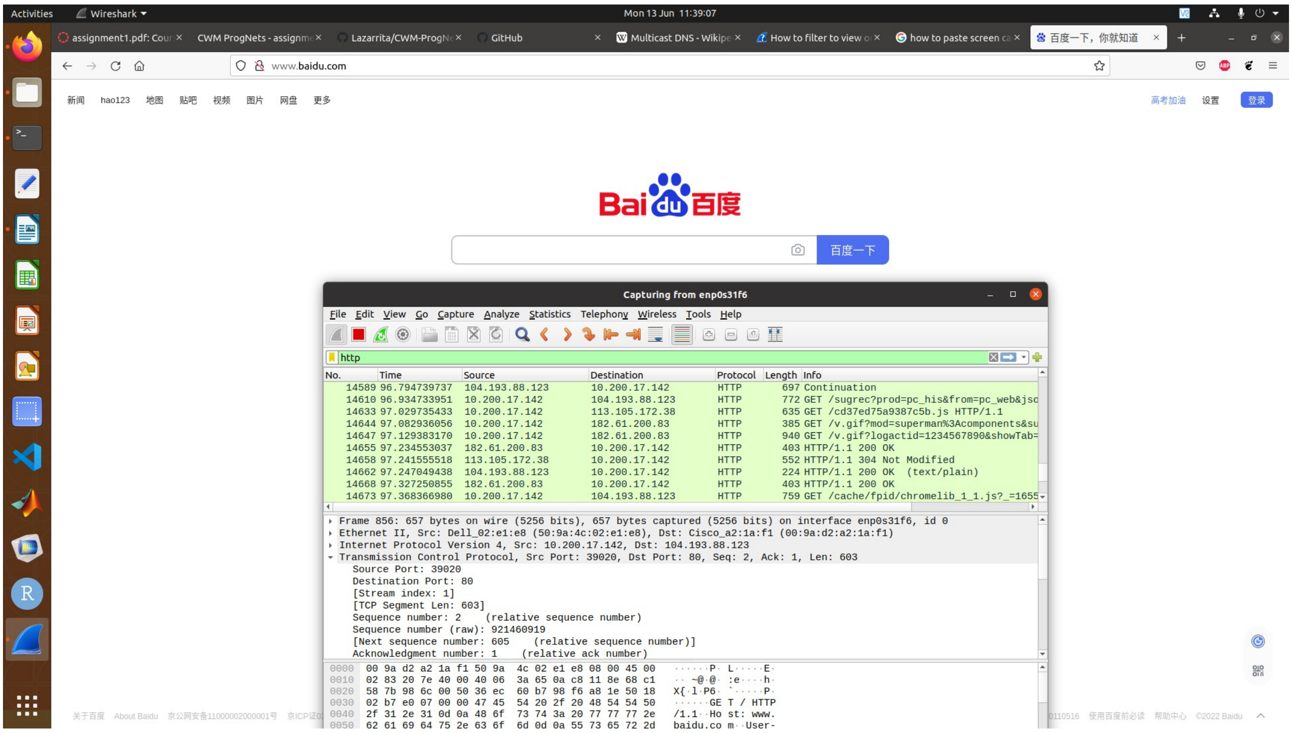
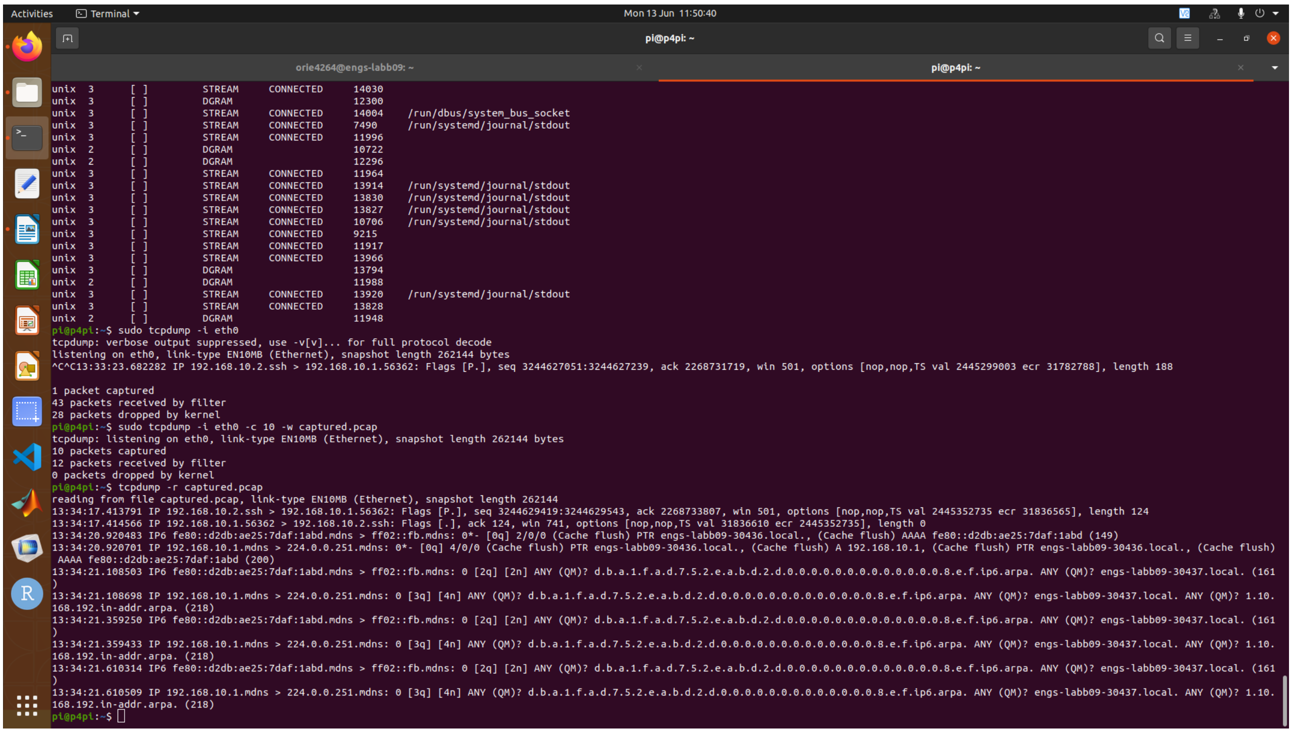
Initial capture are all UDP traffic

In order to capture http traffic, it was necessary to access a website that does not run on https. Baidu.com is an example that then allowed http traffic to be captured.

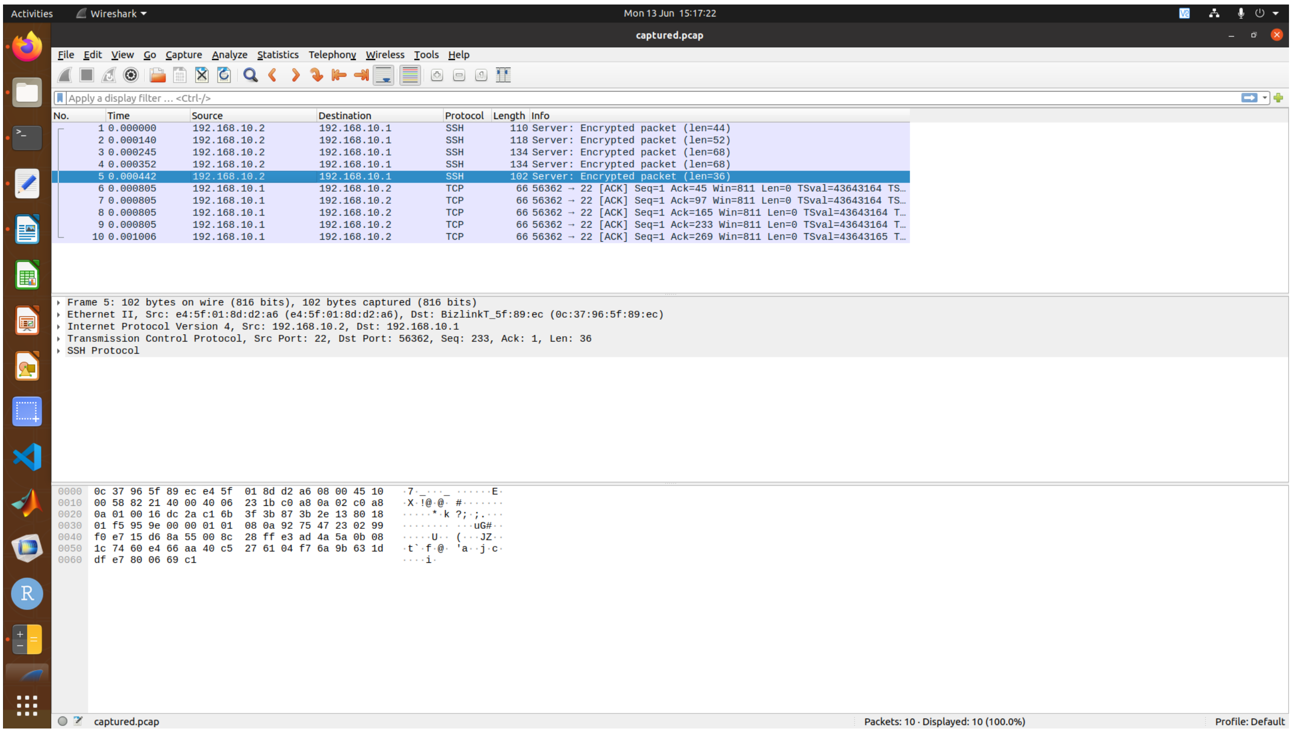


```
http
No.   Time          Source          Destination     Protocol Length Info
14589 96.794739737 104.193.88.123  10.200.17.142   HTTP     697 Continuation
14610 96.934733951 10.200.17.142   104.193.88.123  HTTP     772 GET /sugrec?prod=pc_his&from=pc_web&jsc
14633 97.029735433 10.200.17.142   113.105.172.38  HTTP     635 GET /cd37ed75a9387c5b.js HTTP/1.1
14644 97.082936056 10.200.17.142   182.61.200.83   HTTP     385 GET /v.gif?mod=superman%3Acomponents&su
14647 97.129383170 10.200.17.142   182.61.200.83   HTTP     940 GET /v.gif?logactid=1234567890&showTab=
14655 97.234553037 182.61.200.83   10.200.17.142   HTTP     403 HTTP/1.1 200 OK
14658 97.241555518 113.105.172.38  10.200.17.142   HTTP     552 HTTP/1.1 304 Not Modified
14662 97.247049438 104.193.88.123  10.200.17.142   HTTP     224 HTTP/1.1 200 OK  (text/plain)
14668 97.327250855 182.61.200.83   10.200.17.142   HTTP     403 HTTP/1.1 200 OK
14673 97.368366980 10.200.17.142   104.193.88.123  HTTP     759 GET /cache/fpid/chromelib_1_1.js?_=1655

Frame 856: 657 bytes on wire (5256 bits), 657 bytes captured (5256 bits) on interface enp0s31f6, id 0
Ethernet II, Src: Dell_02:e1:e8 (50:9a:4c:02:e1:e8), Dst: Cisco_a2:1a:f1 (00:9a:d2:a2:1a:f1)
Internet Protocol Version 4, Src: 10.200.17.142, Dst: 104.193.88.123
Transmission Control Protocol, Src Port: 39020, Dst Port: 80, Seq: 2, Ack: 1, Len: 603
  Source Port: 39020
  Destination Port: 80
  [Stream index: 1]
  [TCP Segment Len: 603]
  Sequence number: 2    (relative sequence number)
  Sequence number (raw): 921460919
  [Next sequence number: 605    (relative sequence number)]
  Acknowledgment number: 1    (relative ack number)

0000  00 9a d2 a2 1a f1 50 9a  4c 02 e1 e8 08 00 45 00   ······P· L·····E·
0010  02 83 20 7e 40 00 40 06  3a 65 0a c8 11 8e 68 c1   ·· ~@·@· :e····h·
0020  58 7b 98 6c 00 50 36 ec  60 b7 98 f6 a8 1e 50 18   X{·l·P6· `·····P·
0030  02 b7 e0 07 00 00 47 45  54 20 2f 20 48 54 54 50   ······GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 77 77 77 2e   /1.1··Ho st: www.
0050  62 61 69 64 75 2e 63 6f  6d 0d 0a 55 73 65 72 2d   baidu.co m··User-
```

Capturing using tcpdump, and then reading from the file captured.pcap. 2nd photo below shows the captured packets in wireshark after using scp to transfer captured.pcap from raspberry pi to lab machine.



```
unix  3      [ ]         STREAM     CONNECTED     14030
unix  3      [ ]         DGRAM                    12300
unix  3      [ ]         STREAM     CONNECTED     14004    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     7490     /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     11996
unix  2      [ ]         DGRAM                    10722
unix  2      [ ]         DGRAM                    12296
unix  3      [ ]         STREAM     CONNECTED     11964
unix  3      [ ]         STREAM     CONNECTED     13914    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     13830    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     13827    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     10706    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     9215
unix  3      [ ]         STREAM     CONNECTED     11917
unix  3      [ ]         STREAM     CONNECTED     13966
unix  3      [ ]         DGRAM                    13794
unix  2      [ ]         DGRAM                    11988
unix  3      [ ]         STREAM     CONNECTED     13920    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     13828
unix  2      [ ]         DGRAM                    11948
pi@p4pi:~$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C^C13:33:23.682282 IP 192.168.10.2.ssh > 192.168.10.1.56362: Flags [P.], seq 3244627051:3244627239, ack 2268731719, win 501, options [nop,nop,TS val 2445299003 ecr 31782788], length 188

1 packet captured
43 packets received by filter
28 packets dropped by kernel
pi@p4pi:~$ sudo tcpdump -i eth0 -c 10 -w captured.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
10 packets captured
12 packets received by filter
0 packets dropped by kernel
pi@p4pi:~$ tcpdump -r captured.pcap
reading from file captured.pcap, link-type EN10MB (Ethernet), snapshot length 262144
13:34:17.413791 IP 192.168.10.2.ssh > 192.168.10.1.56362: Flags [P.], seq 3244629419:3244629543, ack 2268733807, win 501, options [nop,nop,TS val 2445352735 ecr 31836565], length 124
13:34:17.414566 IP 192.168.10.1.56362 > 192.168.10.2.ssh: Flags [.], ack 124, win 741, options [nop,nop,TS val 31836610 ecr 2445352735], length 0
13:34:20.920483 IP6 fe80::d2db:ae25:7daf:1abd.mdns > ff02::fb.mdns: 0*- [0q] 2/0/0 (Cache flush) PTR engs-labb09-30436.local., (Cache flush) AAAA fe80::d2db:ae25:7daf:1abd (149)
13:34:20.920701 IP 192.168.10.1.mdns > 224.0.0.251.mdns: 0*- [0q] 4/0/0 (Cache flush) PTR engs-labb09-30436.local., (Cache flush) A 192.168.10.1, (Cache flush) PTR engs-labb09-30436.local., (Cache flush)
 AAAA fe80::d2db:ae25:7daf:1abd (200)
13:34:21.108503 IP6 fe80::d2db:ae25:7daf:1abd.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QM)? d.b.a.1.f.a.d.7.5.2.e.a.b.d.2.d.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. ANY (QM)? engs-labb09-30437.local. (161
)
13:34:21.108698 IP 192.168.10.1.mdns > 224.0.0.251.mdns: 0 [3q] [4n] ANY (QM)? d.b.a.1.f.a.d.7.5.2.e.a.b.d.2.d.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. ANY (QM)? engs-labb09-30437.local. ANY (QM)? 1.10.
168.192.in-addr.arpa. (218)
13:34:21.359250 IP6 fe80::d2db:ae25:7daf:1abd.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QM)? d.b.a.1.f.a.d.7.5.2.e.a.b.d.2.d.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. ANY (QM)? engs-labb09-30437.local. (161
)
13:34:21.359433 IP 192.168.10.1.mdns > 224.0.0.251.mdns: 0 [3q] [4n] ANY (QM)? d.b.a.1.f.a.d.7.5.2.e.a.b.d.2.d.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. ANY (QM)? engs-labb09-30437.local. ANY (QM)? 1.10.
168.192.in-addr.arpa. (218)
13:34:21.610314 IP6 fe80::d2db:ae25:7daf:1abd.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QM)? d.b.a.1.f.a.d.7.5.2.e.a.b.d.2.d.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. ANY (QM)? engs-labb09-30437.local. (161
)
13:34:21.610509 IP 192.168.10.1.mdns > 224.0.0.251.mdns: 0 [3q] [4n] ANY (QM)? d.b.a.1.f.a.d.7.5.2.e.a.b.d.2.d.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. ANY (QM)? engs-labb09-30437.local. ANY (QM)? 1.10.
168.192.in-addr.arpa. (218)
pi@p4pi:~$
```

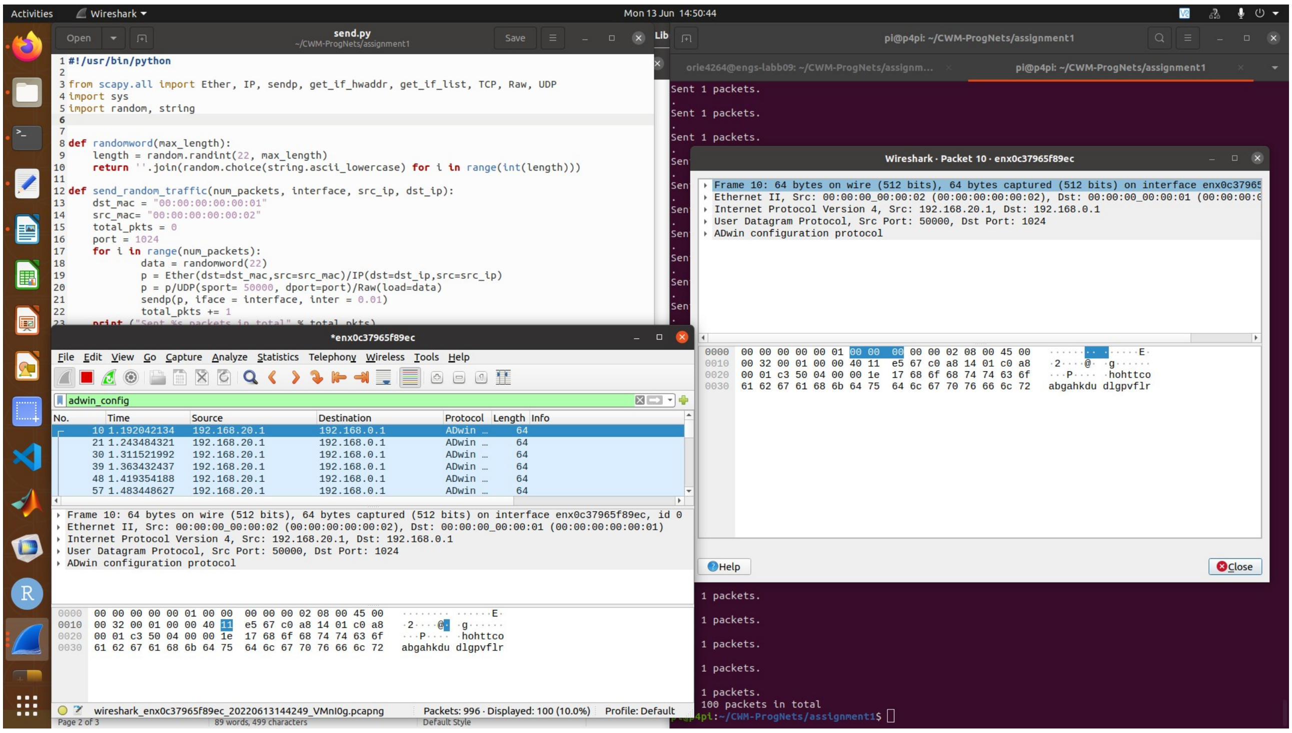

Activities
Wireshark
Mon 13 Jun 15:17:22
captured.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
captured.pcap
Packets: 10 · Displayed: 10 (100.0%)
Profile: Default

```
No.  Time      Source        Destination   Protocol Length Info
   1 0.000000  192.168.10.2  192.168.10.1  SSH      110 Server: Encrypted packet (len=44)
   2 0.000140  192.168.10.2  192.168.10.1  SSH      118 Server: Encrypted packet (len=52)
   3 0.000245  192.168.10.2  192.168.10.1  SSH      134 Server: Encrypted packet (len=68)
   4 0.000352  192.168.10.2  192.168.10.1  SSH      134 Server: Encrypted packet (len=68)
   5 0.000442  192.168.10.2  192.168.10.1  SSH      102 Server: Encrypted packet (len=36)
   6 0.000805  192.168.10.1  192.168.10.2  TCP       66 56362 → 22 [ACK] Seq=1 Ack=45 Win=811 Len=0 TSval=43643164 TS…
   7 0.000805  192.168.10.1  192.168.10.2  TCP       66 56362 → 22 [ACK] Seq=1 Ack=97 Win=811 Len=0 TSval=43643164 TS…
   8 0.000805  192.168.10.1  192.168.10.2  TCP       66 56362 → 22 [ACK] Seq=1 Ack=165 Win=811 Len=0 TSval=43643164 T…
   9 0.000805  192.168.10.1  192.168.10.2  TCP       66 56362 → 22 [ACK] Seq=1 Ack=233 Win=811 Len=0 TSval=43643164 T…
  10 0.001006  192.168.10.1  192.168.10.2  TCP       66 56362 → 22 [ACK] Seq=1 Ack=269 Win=811 Len=0 TSval=43643165 T…
```

```
› Frame 5: 102 bytes on wire (816 bits), 102 bytes captured (816 bits)
› Ethernet II, Src: e4:5f:01:8d:d2:a6 (e4:5f:01:8d:d2:a6), Dst: BizlinkT_5f:89:ec (0c:37:96:5f:89:ec)
› Internet Protocol Version 4, Src: 192.168.10.2, Dst: 192.168.10.1
› Transmission Control Protocol, Src Port: 22, Dst Port: 56362, Seq: 233, Ack: 1, Len: 36
› SSH Protocol
```

```
0000  0c 37 96 5f 89 ec e4 5f  01 8d d2 a6 08 00 45 10   ·7·_···_ ······E·
0010  00 58 82 21 40 00 40 06  23 1b c0 a8 0a 02 c0 a8   ·X·!@·@· #·······
0020  0a 01 00 16 dc 2a c1 6b  3f 3b 87 3b 2e 13 80 18   ·····*·k ?;·;.···
0030  01 f5 95 9e 00 00 01 01  08 0a 92 75 47 23 02 99   ········ ···uG#··
0040  f0 e7 15 d6 8a 55 00 8c  28 ff e3 ad 4a 5a 0b 08   ·····U·· (···JZ··
0050  1c 74 60 e4 66 aa 40 c5  27 61 04 f7 6a 9b 63 1d   ·t`·f·@· 'a··j·c·
0060  df e7 80 06 69 c1                                  ····i·
```

adwin_config is able to filter the incoming packets onto the lab machine to only capture the 100 packets sent from the raspberry pi to the lab machine. Using 192.168.20.1 allows easier checking of which files were sent without filtering. (Mainly for next part)


Activities
Wireshark
Mon 13 Jun 14:50:44
Open
send.py
~/CWM-ProgNets/assignment1
Save
Lib
pi@p4pi: ~/CWM-ProgNets/assignment1
orie4264@engs-labb09: ~/CWM-ProgNets/assignm…
pi@p4pi: ~/CWM-ProgNets/assignment1
*enx0c37965f89ec
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Packet 10 · enx0c37965f89ec
Help
Close
wireshark_enx0c37965f89ec_20220613144249_VMnI0g.pcapng
Packets: 996 · Displayed: 100 (10.0%)
Profile: Default
Page 2 of 3
89 words, 499 characters
Default Style

```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(max_length):
    length = random.randint(22, max_length)
    return ''.join(random.choice(string.ascii_lowercase) for i in range(int(length)))

def send_random_traffic(num_packets, interface, src_ip, dst_ip):
    dst_mac = "00:00:00:00:00:01"
    src_mac= "00:00:00:00:00:02"
    total_pkts = 0
    port = 1024
    for i in range(num_packets):
            data = randomword(22)
            p = Ether(dst=dst_mac,src=src_mac)/IP(dst=dst_ip,src=src_ip)
            p = p/UDP(sport= 50000, dport=port)/Raw(load=data)
            sendp(p, iface = interface, inter = 0.01)
            total_pkts += 1
    print ("Sent %s packets in total" % total_pkts)
```

adwin_config

```
No.  Time         Source        Destination  Protocol  Length Info
  10 1.192042134  192.168.20.1  192.168.0.1  ADwin …   64
  21 1.243484321  192.168.20.1  192.168.0.1  ADwin …   64
  30 1.311521992  192.168.20.1  192.168.0.1  ADwin …   64
  39 1.363432437  192.168.20.1  192.168.0.1  ADwin …   64
  48 1.419354188  192.168.20.1  192.168.0.1  ADwin …   64
  57 1.483448627  192.168.20.1  192.168.0.1  ADwin …   64
```

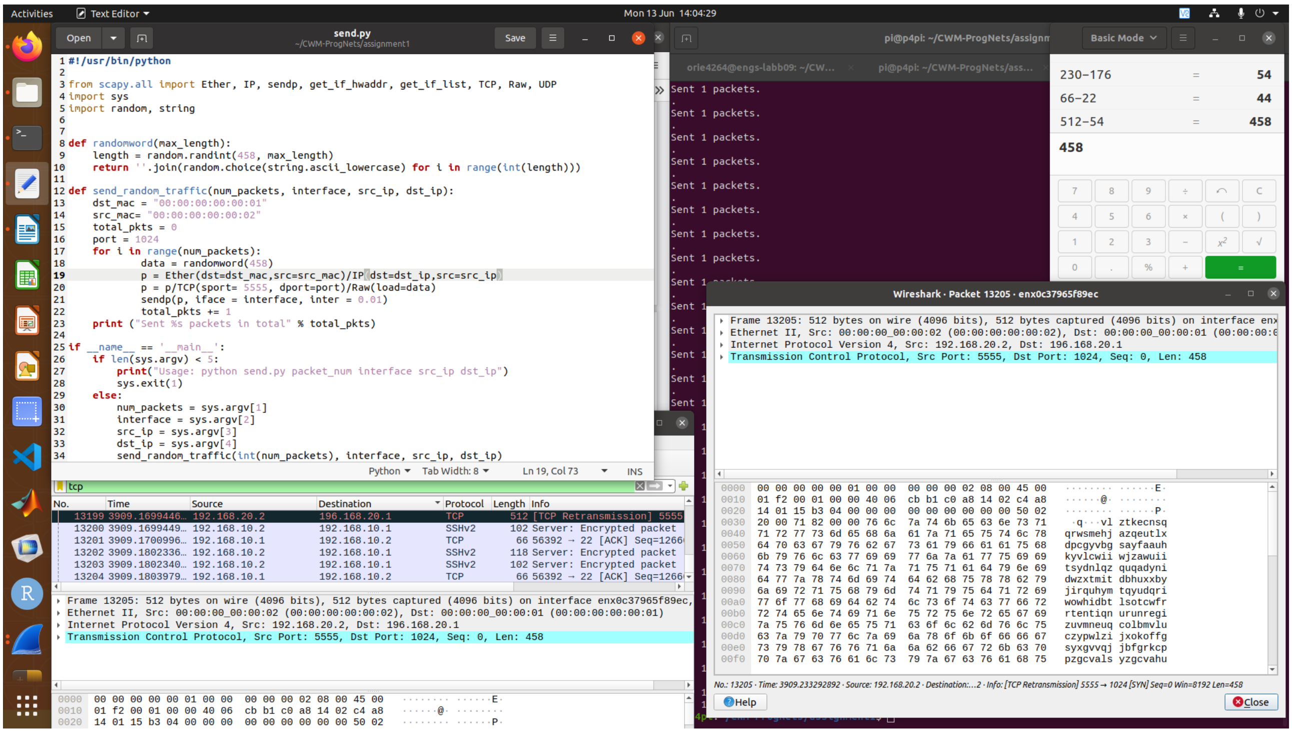
```
› Frame 10: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface enx0c37965f89ec, id 0
› Ethernet II, Src: 00:00:00_00:00:02 (00:00:00:00:00:02), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
› Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.0.1
› User Datagram Protocol, Src Port: 50000, Dst Port: 1024
› ADwin configuration protocol
```

```
0000  00 00 00 00 00 01 00 00  00 00 00 02 08 00 45 00   ········ ······E·
0010  00 32 00 01 00 00 40 11  e5 67 c0 a8 14 01 c0 a8   ·2····@· ·g······
0020  00 01 c3 50 04 00 00 1e  17 68 6f 68 74 74 63 6f   ···P···· ·hohttco
0030  61 62 67 61 68 6b 64 75  64 6c 67 70 76 66 6c 72   abgahkdu dlgpvflr
```

```
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sen
.
Sen
.
Sen
.
Sen
.
Sen
.
Sen
.
Sen
.
1 packets.

1 packets.

1 packets.

1 packets.

1 packets.
100 packets in total
pi@p4pi:~/CWM-ProgNets/assignment1$ 
```

1 packet size = 64 bytes.
Colour is blue, meaning that it is UDP protocol.

Following picture shows code used to achieve 512B in size. 458 is the length used, as I found out via experimenting with a few values that the difference between length and word size is a fixed 54 when changed to TCP. When in UDP the difference is only 42 (64-22), likely due to differences in overhead when implementing the different protocols.


Activities
Text Editor
Mon 13 Jun 14:04:29
Open
send.py
~/CWM-ProgNets/assignment1
Save
pi@p4pi: ~/CWM-ProgNets/assignm
orie4264@engs-labb09: ~/CW...
pi@p4pi: ~/CWM-ProgNets/ass...
Basic Mode
Python
Tab Width: 8
Ln 19, Col 73
INS
Wireshark · Packet 13205 · enx0c37965f89ec
Help
Close

```
#!/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(max_length):
    length = random.randint(458, max_length)
    return ''.join(random.choice(string.ascii_lowercase) for i in range(int(length)))

def send_random_traffic(num_packets, interface, src_ip, dst_ip):
    dst_mac = "00:00:00:00:00:01"
    src_mac= "00:00:00:00:00:02"
    total_pkts = 0
    port = 1024
    for i in range(num_packets):
            data = randomword(458)
            p = Ether(dst=dst_mac,src=src_mac)/IP(dst=dst_ip,src=src_ip)
            p = p/TCP(sport= 5555, dport=port)/Raw(load=data)
            sendp(p, iface = interface, inter = 0.01)
            total_pkts += 1
    print ("Sent %s packets in total" % total_pkts)

if __name__ == '__main__':
    if len(sys.argv) < 5:
        print("Usage: python send.py packet_num interface src_ip dst_ip")
        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)
```

```
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
```

```
230−176     =     54
66−22       =     44
512−54      =    458
458
```

tcp

No.	Time	Source	Destination	Protocol	Length	Info
13199	3909.1699446…	192.168.20.2	196.168.20.1	TCP	512	[TCP Retransmission] 5555
13200	3909.1699449…	192.168.10.2	192.168.10.1	SSHv2	102	Server: Encrypted packet
13201	3909.1700996…	192.168.10.1	192.168.10.2	TCP	66	56392 → 22 [ACK] Seq=1266
13202	3909.1802336…	192.168.10.2	192.168.10.1	SSHv2	118	Server: Encrypted packet
13203	3909.1802340…	192.168.10.2	192.168.10.1	SSHv2	102	Server: Encrypted packet
13204	3909.1803979…	192.168.10.1	192.168.10.2	TCP	66	56392 → 22 [ACK] Seq=1266

```
Frame 13205: 512 bytes on wire (4096 bits), 512 bytes captured (4096 bits) on interface enx0c37965f89ec,
Ethernet II, Src: 00:00:00_00:00:02 (00:00:00:00:00:02), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
Internet Protocol Version 4, Src: 192.168.20.2, Dst: 196.168.20.1
Transmission Control Protocol, Src Port: 5555, Dst Port: 1024, Seq: 0, Len: 458
```

```
0000  00 00 00 00 00 01 00 00  00 00 00 02 08 00 45 00   ········ ······E·
0010  01 f2 00 01 00 00 40 06  cb b1 c0 a8 14 02 c4 a8   ······@· ········
0020  14 01 15 b3 04 00 00 00  00 00 00 00 00 00 50 02   ········ ······P·
0030  20 00 71 82 00 00 76 6c  7a 74 6b 65 63 6e 73 71    ·q···vl ztkecnsq
0040  71 72 77 73 6d 65 68 6a  61 7a 71 65 75 74 6c 78   qrwsmehj azqeutlx
0050  64 70 63 67 79 76 62 67  73 61 79 66 61 61 75 68   dpcgyvbg sayfaauh
0060  6b 79 76 6c 63 77 69 69  77 6a 7a 61 77 75 69 69   kyvlcwii wjzawuii
0070  74 73 79 64 6e 6c 71 7a  71 75 71 61 64 79 6e 69   tsydnlqz quqadyni
0080  64 77 7a 78 74 6d 69 74  64 62 68 75 78 78 62 79   dwzxtmit dbhuxxby
0090  6a 69 72 71 75 68 79 6d  74 71 79 75 64 71 72 69   jirquhym tqyudqri
00a0  77 6f 77 68 69 64 62 74  6c 73 6f 74 63 77 66 72   wowhidbt lsotcwfr
00b0  72 74 65 6e 74 69 71 6e  75 72 75 6e 72 65 67 69   rtentiqn urunregi
00c0  7a 75 76 6d 6e 65 75 71  63 6f 6c 62 6d 76 6c 75   zuvmneuq colbmvlu
00d0  63 7a 79 70 77 6c 7a 69  6a 78 6f 6b 6f 66 66 67   czypwlzi jxokoffg
00e0  73 79 78 67 76 76 71 6a  6a 62 66 67 72 6b 63 70   syxgvvqj jbfgrkcp
00f0  70 7a 67 63 76 61 6c 73  79 7a 67 63 76 61 68 75   pzgcvals yzgcvahu
```

No.: 13205 · Time: 3909.233292892 · Source: 192.168.20.2 · Destination:…2 · Info: [TCP Retransmission] 5555 → 1024 [SYN] Seq=0 Win=8192 Len=458

Source port is changed to 5555. Source ip is set as 192.168.20.1 to be easier to see in wire-shark (turns up in black).

Code is copied below (via txt). Highlighted changes made

```
#!/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(max_length):
    length = random.randint(458, max_length)
    return ''.join(random.choice(string.ascii_lowercase) for i in range(int(length)))

def send_random_traffic(num_packets, interface, src_ip, dst_ip):
    dst_mac = "00:00:00:00:00:01"
    src_mac= "00:00:00:00:00:02"
    total_pkts = 0
    port = 1024
    for i in range(num_packets):
            data = randomword(458)
            p = Ether(dst=dst_mac,src=src_mac)/IP(dst=dst_ip,src=src_ip)
            p = p/TCP(sport= 5555, dport=port)/Raw(load=data)
            sendp(p, iface = interface, inter = 0.01)
            total_pkts += 1
    print ("Sent %s packets in total" % total_pkts)

if __name__ == '__main__':
    if len(sys.argv) < 5:
        print("Usage: python send.py packet_num interface src_ip dst_ip")
        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)
```