**The flow of the code:**

1. The option parser- I’ve included a c++ option parser(optparse) to parse the command line options.

Source: <https://github.com/myint/optparse>

1. User can provide the below mentioned 4 arguments to the application in the order in which they are written:

* -i (interface)
* -r (file)
* -s (string)
* expression

mydump [-i interface] [-r file] [-s string] expression

-i Live capture from the network device <interface> (e.g., eth0). If not

specified, mydump should automatically select a default interface to

listen on (hint 1). Capture should continue indefinitely until the user

terminates the program.

-r Read packets from <file> in tcpdump format.

-s Keep only packets that contain <string> in their payload (after any BPF

filter is applied). You are not required to implement wildcard or regular

expression matching. A simple string matching operation should suffice

(hint 3).

<expression> is a BPF filter that specifies which packets will be dumped. If

no filter is given, all packets seen on the interface (or contained in the

trace) should be dumped. Otherwise, only packets matching <expression> should

be dumped

1. Once the application is run, it checks if a pcap file has been provided (option –r) , if yes, the pcap file is picked up and the packets captured inside it are analyzed. If the pcap file is not provided the application checks if an interface to capture packets has been provided by the user (option –i).

If the interface is also not provided, the default device is picked up.

1. The pcap\_handler\_callback() function is called for every packet.
2. It first checks if the packet is an IP Packet,

* if not, the mac addresses and ether type of the packet (ARP or RARP or other) is printed and the application exits.
* if yes, the IP and length, ether typed of the packet is printed. The application also checks the protocol type (TCP, UDP, ICMP or other) of the packet and the corresponding pointers (to print the payload and other details like the port numbers) are set.

1. TCP and UDP – port number and the payload is printed
2. ICMP and other – payload is printed
3. A packet is picked up and printed only if the packet is a valid packet (valid packet- if the user has supplied a string that the packed should contain, only the packets with payloads containing that string are printed. If no string is provided all packets are valid packets.)

**Commands:**

Compile: make all

run: bin/mydump

**Example outputs of the application**:

* <supplying a pcap file to the application>

MacBook-Pro:hw2 selinadeepkaur$ ./bin/mydump -r ./src/hw1.pcap

2013-01-13

 c4:3d:c7:17:6f:9b ->  ff:ff:ff:ff:ff:ff type 0x806 (ARP packet)

 2013-01-13

 44:6d:57:f6:7e:00 -> 01:00:5e:00:00:fb type 0x800 (IP packet) len 70

 192.168.0.11:5353 -> 224.0.0.251:5353 UDP

 00 00 00 00 00 01 00 00  00 00 00 00 0d 5f 61 70    .............\_ap

 70 6c 65 2d 6d 6f 62 64  65 76 04 5f 74 63 70 05    ple-mobdev.\_tcp.

 6c 6f 63 61 6c 00 00 0c  00 01                      local.....

* With < –s option, supplying Broadcom as the string to be matched>

MacBook-Pro:hw2 selinadeepkaur$ ./bin/mydump -r ./src/hw1.pcap -s Broadcom

2013-01-14

 c4:3d:c7:17:6f:9b -> 01:00:5e:7f:ff:fa type 0x800 (IP packet) len 326

 192.168.0.1:1900 -> 239.255.255.250:1900 UDP

 4e 4f 54 49 46 59 20 2a  20 48 54 54 50 2f 31 2e    NOTIFY \* HTTP/1.

 31 0d 0a 48 6f 73 74 3a  20 32 33 39 2e 32 35 35    1..Host: 239.255

 2e 32 35 35 2e 32 35 30  3a 31 39 30 30 0d 0a 43    .255.250:1900..C

 61 63 68 65 2d 43 6f 6e  74 72 6f 6c 3a 20 6d 61    ache-Control: ma

 78 2d 61 67 65 3d 36 30  0d 0a 4c 6f 63 61 74 69    x-age=60..Locati

 6f 6e 3a 20 68 74 74 70  3a 2f 2f 31 39 32 2e 31    on: http://192.1

 36 38 2e 30 2e 31 3a 31  39 30 30 2f 57 46 41 44    68.0.1:1900/WFAD

 65 76 69 63 65 2e 78 6d  6c 0d 0a 4e 54 53 3a 20    evice.xml..NTS:

 73 73 64 70 3a 61 6c 69  76 65 0d 0a 53 65 72 76    ssdp:alive..Serv

 65 72 3a 20 50 4f 53 49  58 2c 20 55 50 6e 50 2f    er: POSIX, UPnP/

 31 2e 30 20 42 72 6f 61  64 63 6f 6d 20 55 50 6e    1.0 Broadcom UPn

 50 20 53 74 61 63 6b 2f  65 73 74 69 6d 61 74 69    P Stack/estimati

 6f 6e 20 31 2e 30 30 0d  0a 4e 54 3a 20 75 75 69    on 1.00..NT: uui

 64 3a 46 35 31 39 33 39  30 41^C 2d 34 34 44 44 2d    d:F519390A-44DD-

* <supplying the interface, eno>

MacBook-Pro:hw2 selinadeepkaur$ ./bin/mydump -i en0

 2017-10-13

 78:4f:43:97:1c:9d -> b8:af:67:63:a3:28 type 0x800 (IP packet) len 51

 172.24.21.17:51301 -> 74.125.22.189:443 UDP

 0c d3 bd 25 51 77 2b 6f  38 d3 c7 00 23 1c 1d 0a    ...%Qw+o8...#...

 5e db 1f 53 11 52 28                                ^..S.R(?

 2017-10-13

 78:4f:43:97:1c:9d -> b8:af:67:63:a3:28 type 0x800 (IP packet) len 91

 172.24.21.17:49452 -> 34.196.41.207:443 TCP

 17 03 03 00 22 00 00 00  00 00 00 00 1a 23 f8 d8    ...."........#..

 78 11 71 ba 72 2b 88 ac  7a ad e2 70 78 c0 c9 e3    x.q.r+..z..px...

 10 4e 79 93 d5 ab 83                                .Ny....?

* With <default interface>

MacBook-Pro:hw2 selinadeepkaur$ ./bin/mydump

 2017-10-13

 b8:af:67:63:a3:28 -> 78:4f:43:97:1c:9d type 0x800 (IP packet) len 83

 34.196.41.207:443 -> 172.24.21.17:49452 TCP

 17 03 03 00 1a bd 5c 84  ce 07 d3 3c 02 ea a4 dc    ......\....<....

 d9 3e 4c 4c 8f 64 af 52  72 a6 bf ca de 68 43       .>LL.d.Rr....hC?

 2017-10-13

 78:4f:43:97:1c:9d -> b8:af:67:63:a3:28 type 0x800 (IP packet) len 52

 172.24.21.17:49452 -> 34.196.41.207:443 TCP

 2017-10-13

 78:4f:43:97:1c:9d -> b8:af:67:63:a3:28 type 0x800 (IP packet) len 87

 172.24.21.17:49452 -> 34.196.41.207:443 TCP

 17 03 03 00 1e 00 00 00  00 00 00 00 05 fd 45 bb    ..............E.

 8b df 72 3f c2 30 e6 e3  4a a5 fc 19 81 d7 93 16    ..r?.0..J.......

 24 63 b5                                            $c.

 2017-10-13

 b8:af:67:63:a3:28 -> 78:4f:43:97:1c:9d type 0x800 (IP packet) len 52

 34.196.41.207:443 -> 172.24.21.17:49452 TCP

 2017-10-13

 b8:af:67:63:a3:28 -> 78:4f:43:97:1c:9d type 0x800 (IP packet) len 132

 34.196.41.207:443 -> 172.24.21.17:49452 TCP

 17 03 03 00 4b bd 5c 84  ce 07 d3 3c 03 b0 6b 18    ....K.\....<..k.

 c1 0f de c2 50 8c ec dc  1a b9 11 b8 7a 9c 84 32    ....P.......z..2

 45 e5 f3 57 a7 91 a7 7b  67 7d c1 63 36 0c 41 a5    E..W...{g}.c6.A.

 66 5f cb 83 e2 71 3d e0  73 17 b9 c1 7d bd ad 7e    f\_...q=.s...}..~

 0b 83 af 5f cc 95 dc 61  be 72 65 8d 45 37 a2 28    ...\_...a.re.E7.(

* With <expression = port 53>

MacBook-Pro:hw2 selinadeepkaur$ ./bin/mydump -r ./src/hw1.pcap port 53

2013-01-14

 c4:3d:c7:17:6f:9b -> 00:0c:29:e9:94:8e type 0x800 (IP packet) len 79

 194.168.4.100:53 -> 192.168.0.200:57270 UDP

 c6 33 81 80 00 01 00 01  00 00 00 00 06 65 78 74    .3...........ext

 72 61 73 06 75 62 75 6e  74 75 03 63 6f 6d 00 00    ras.ubuntu.com..

 01 00 01 c0 0c 00 01 00  01 00 00 01 c0 00 04 5b    ...............[

 bd 58 21                                            .X!

 2013-01-14

 c4:3d:c7:17:6f:9b -> 00:0c:29:e9:94:8e type 0x800 (IP packet) len 97

 194.168.4.100:53 -> 192.168.0.200:48254 UDP

 89 09 81 80 00 01 00 02  00 00 00 00 08 73 65 63    .............sec

 75 72 69 74 79 06 75 62  75 6e 74 75 03 63 6f 6d    urity.ubuntu.com

 00 00 01 00 01 c0 0c 00  01 00 01 00 00 00 2d 00    ..............-.

 04 5b bd 5c be c0 0c 00  01 00 01 00 00 00 2d 00    .[.\..........-.

 04 5b bd 5c c8

* With < BPF filter>

MacBook-Pro:hw2 selinadeepkaur$ ./bin/mydump -r ./src/hw1.pcap -s ubuntu port 53

 2013-01-13

 00:0c:29:e9:94:8e -> c4:3d:c7:17:6f:9b type 0x800 (IP packet) len 63

 192.168.0.200:52449 -> 194.168.4.100:53 UDP

 6b a1 01 00 00 01 00 00  00 00 00 00 06 65 78 74    k............ext

 72 61 73 06 75 62 75 6e  74 75 03 63 6f 6d 00 00    ras.ubuntu.com..

 01 00 01                                            ...

 2013-01-13

 00:0c:29:e9:94:8e -> c4:3d:c7:17:6f:9b type 0x800 (IP packet) len 67

 192.168.0.200:47755 -> 194.168.4.100:53 UDP

 ed 2e 01 00 00 01 00 00  00 00 00 00 02 75 73 07    .............us.

 61 72 63 68 69 76 65 06  75 62 75 6e 74 75 03 63    archive.ubuntu.c

 6f 6d 00 00 01 00 01                                om.....

* With <BPF filter>

 MacBook-Pro:hw2 selinadeepkaur$ ./bin/mydump -r ./src/hw1.pcap tcp

2013-01-14

 c4:3d:c7:17:6f:9b -> 00:0c:29:e9:94:8e type 0x800 (IP packet) len 233

 1.234.31.20:55672 -> 192.168.0.200:80 TCP

 47 45 54 20 2f 4d 79 41  64 6d 69 6e 2f 73 63 72    GET /MyAdmin/scr

 69 70 74 73 2f 73 65 74  75 70 2e 70 68 70 20 48    ipts/setup.php H

 54 54 50 2f 31 2e 31 0d  0a 41 63 63 65 70 74 3a    TTP/1.1..Accept:

 20 2a 2f 2a 0d 0a 41 63  63 65 70 74 2d 4c 61 6e     \*/\*..Accept-Lan

 67 75 61 67 65 3a 20 65  6e 2d 75 73 0d 0a 41 63    guage: en-us..Ac

 63 65 70 74 2d 45 6e 63  6f 64 69 6e 67 3a 20 67    cept-Encoding: g

 7a 69 70 2c 20 64 65 66  6c 61 74 65 0d 0a 55 73    zip, deflate..Us

 65 72 2d 41 67 65 6e 74  3a 20 5a 6d 45 75 0d 0a    er-Agent: ZmEu..

 48 6f 73 74 3a 20 38 36  2e 30 2e 33 33 2e 32 30    Host: 86.0.33.20

 0d 0a 43 6f 6e 6e 65 63  74 69 6f 6e 3a 20 43 6c    ..Connection: Cl

 6f 73 65 0d 0a 0d 0a                                ose....

References:

<https://github.com/myint/optparse>

<http://www.tcpdump.org/pcap.html>

<http://www.tcpdump.org/sniffex.c>