Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 08

Course code: ICT-3207

Course title: Computer Networking Lab

Date of Performance:

Date of Submission:

Submitted by

Name: Ashikur Rahman Miran & Rafiul Hasan

ID:IT-18014 & IT-18016

3rd year 2nd semester

Session: 2017-2018

Dept. of ICT

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Lab Report No.: 08

Lab Report Name: Installing Wireshark in Linux Operating System.

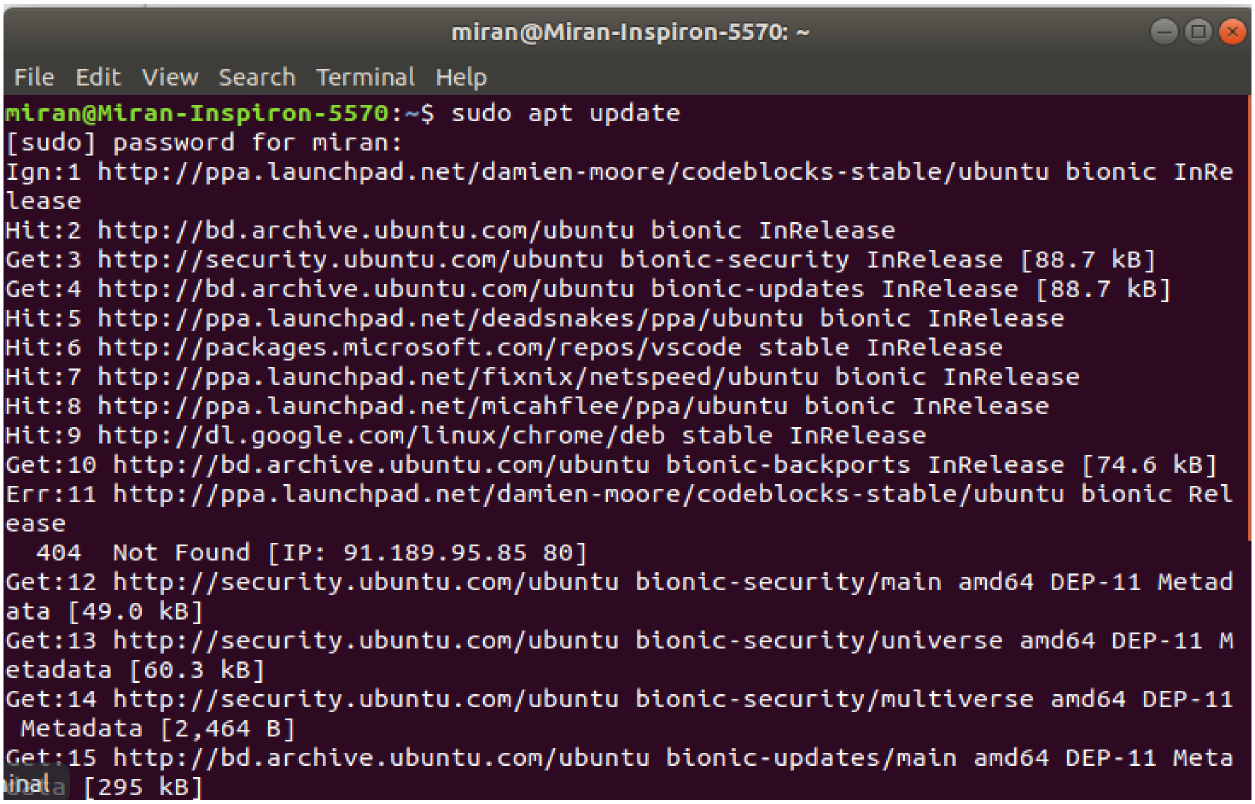
INSTALLING WIRESHARK:

Wireshark is a network packet analyzer. It captures every packet getting in or out of a network interface and shows them in a nicely formatted text. It is used by Network Engineers all over the world. How to install Wireshark is given below step by step:

First update the APT package repository cache with the following command:

$ sudo apt update

The APT package repository cache should be updated.


miran@Miran-Inspiron-5570: ~
File Edit View Search Terminal Help

```
miran@Miran-Inspiron-5570:~$ sudo apt update
[sudo] password for miran:
Ign:1 http://ppa.launchpad.net/damien-moore/codeblocks-stable/ubuntu bionic InRe
lease
Hit:2 http://bd.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://bd.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Hit:5 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Hit:6 http://packages.microsoft.com/repos/vscode stable InRelease
Hit:7 http://ppa.launchpad.net/fixnix/netspeed/ubuntu bionic InRelease
Hit:8 http://ppa.launchpad.net/micahflee/ppa/ubuntu bionic InRelease
Hit:9 http://dl.google.com/linux/chrome/deb stable InRelease
Get:10 http://bd.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Err:11 http://ppa.launchpad.net/damien-moore/codeblocks-stable/ubuntu bionic Rel
ease
  404  Not Found [IP: 91.189.95.85 80]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metad
ata [49.0 kB]
Get:13 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 M
etadata [60.3 kB]
Get:14 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11
 Metadata [2,464 B]
Get:15 http://bd.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Meta
data [295 kB]
```

Now, Run the following command to install Wireshark on your Ubuntu machine:

$ sudo apt get install wireshark

Wireshark should be installed.

Run the following command to add your user to the Wireshark group:

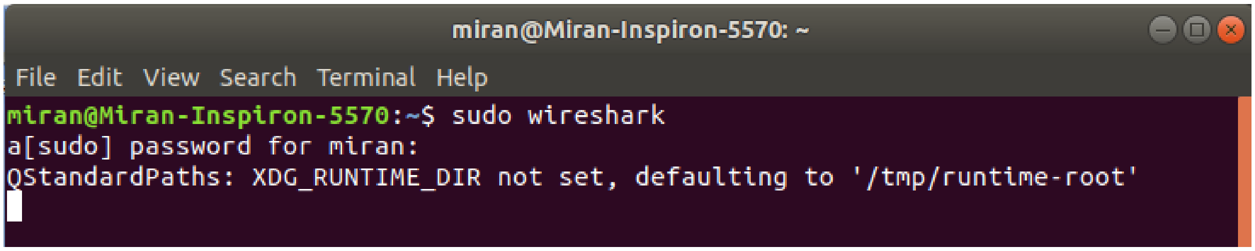
$ sudo usermod -aG wireshark $(whoami)

Now reboot your computer with the following command:

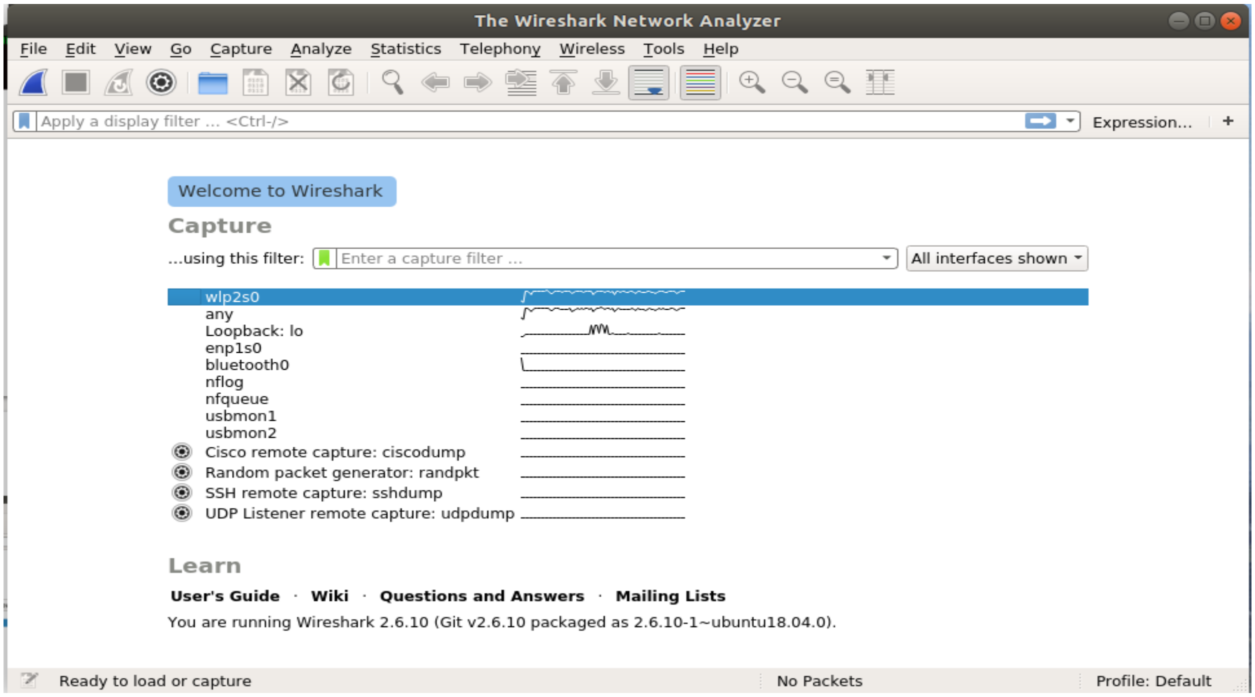
$ sudo reboot

Now run Wireshark using the following command:

$ sudo wireshark


miran@Miran-Inspiron-5570: ~
File Edit View Search Terminal Help

```
miran@Miran-Inspiron-5570:~$ sudo wireshark
a[sudo] password for miran:
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+
Welcome to Wireshark
Capture
…using this filter:
Enter a capture filter …
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

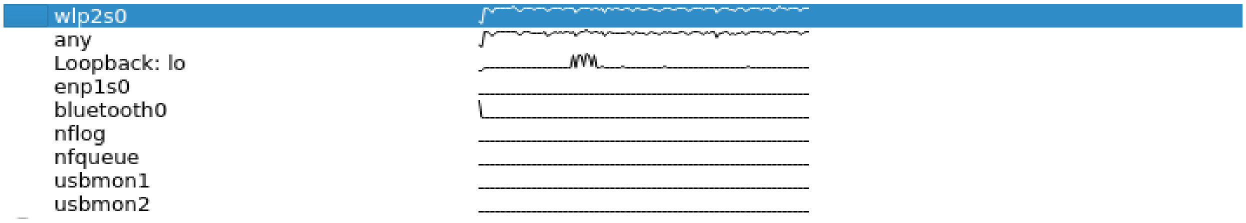
```
wlp2s0
any
Loopback: lo
enp1s0
bluetooth0
nflog
nfqueue
usbmon1
usbmon2
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
```

You are running Wireshark 2.6.10 (Git v2.6.10 packaged as 2.6.10-1~ubuntu18.04.0).

Ready to load or capture | No Packets | Profile: Default

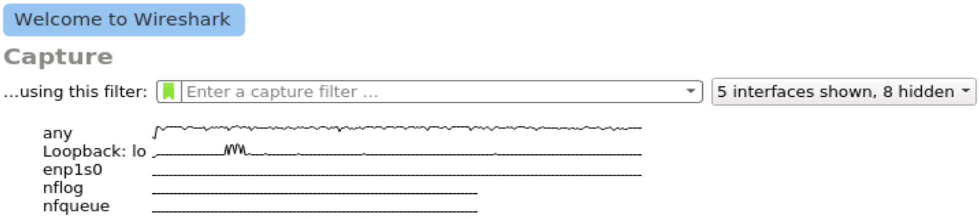
Now we will capture packages using Wireshark.

When you start Wireshark, you will see a list of interfaces that you can capture packets to and from.



```
wlp2s0
any
Loopback: lo
enp1s0
bluetooth0
nflog
nfqueue
usbmon1
usbmon2
```

There are many types of interfaces you can monitor using Wireshark, for example, **Wired, Wireless**, USB and many external devices. You can choose to show specific types of interfaces in the welcome screen from the marked section of the screenshot below.



Welcome to Wireshark

Capture

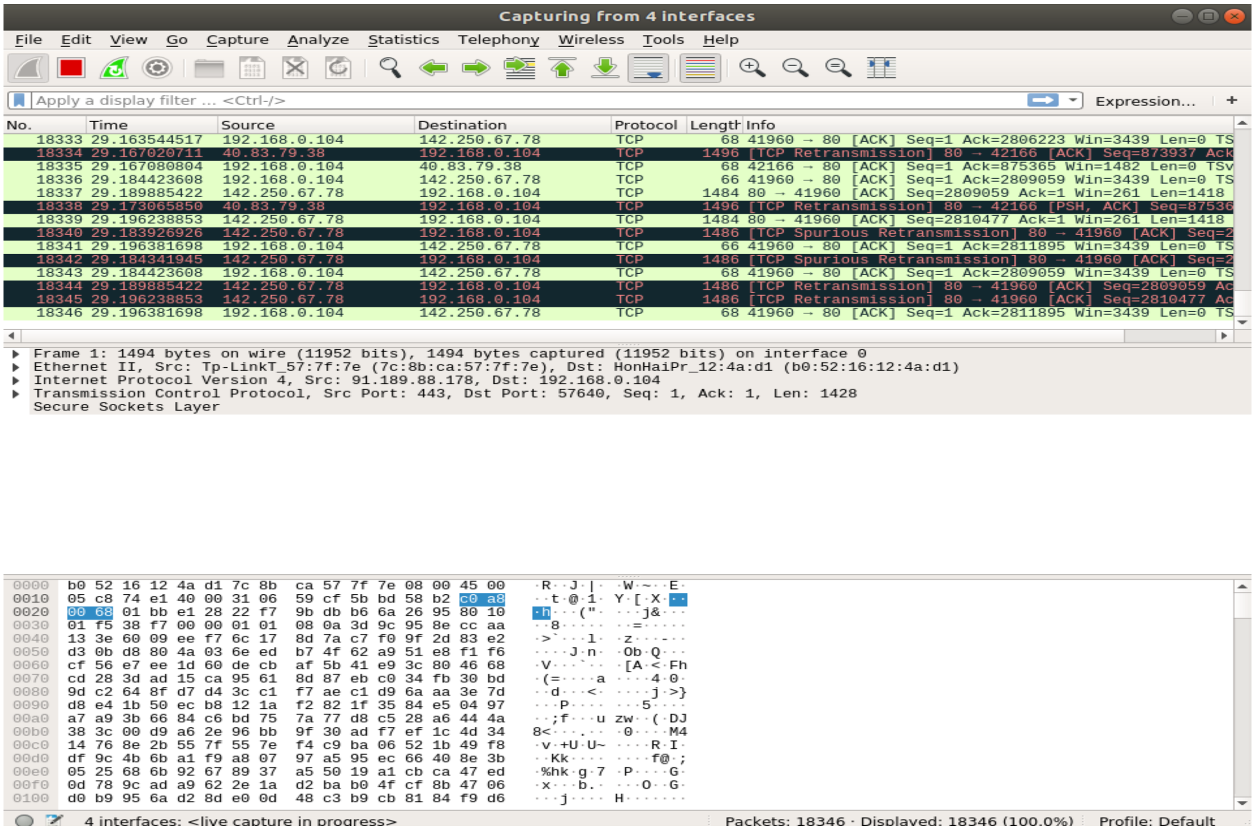
...using this filter: Enter a capture filter ... 5 interfaces shown, 8 hidden

```
any
Loopback: lo
enp1s0
nflog
nfqueue
```

Now to start capturing packets, just select the interface (in my case interface ens33) and click on the **Start capturing packets** icon as marked in the screenshot below.

You can also capture packets to and from multiple interfaces at the same time. Just press and hold **<Ctrl>** and click on the interfaces that you want to capture packets to and from and then click on the **Start capturing packets** icon as marked in the screenshot below.

I pinged google.com from the terminal and many packets were captured



Capturing from 4 interfaces

Apply a display filter ... <Ctrl-/>

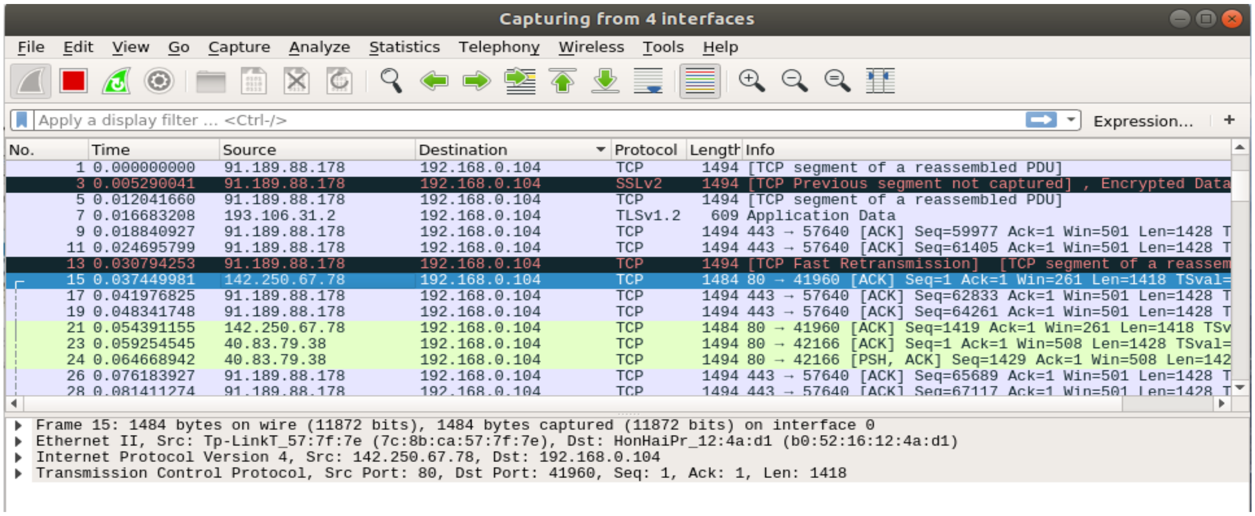
```
No.    Time         Source         Destination    Protocol Length Info
 18333 29.163544517 192.168.0.104  142.250.67.78  TCP        68 41960 → 80 [ACK] Seq=1 Ack=2806223 Win=3439 Len=0 TS
 18334 29.167020711 40.83.79.38    192.168.0.104  TCP      1496 [TCP Retransmission] 80 → 42166 [ACK] Seq=873937 Ack
 18335 29.167080804 192.168.0.104  40.83.79.38    TCP        68 42166 → 80 [ACK] Seq=1 Ack=875365 Win=1482 Len=0 TSv
 18336 29.184423608 192.168.0.104  142.250.67.78  TCP        66 41960 → 80 [ACK] Seq=1 Ack=2809059 Win=3439 Len=0 TS
 18337 29.189885422 142.250.67.78  192.168.0.104  TCP      1484 80 → 41960 [ACK] Seq=2809059 Ack=1 Win=261 Len=1418
 18338 29.173065850 40.83.79.38    192.168.0.104  TCP      1496 [TCP Retransmission] 80 → 42166 [PSH, ACK] Seq=87536
 18339 29.196238853 142.250.67.78  192.168.0.104  TCP      1484 80 → 41960 [ACK] Seq=2810477 Ack=1 Win=261 Len=1418
 18340 29.183926926 142.250.67.78  192.168.0.104  TCP      1486 [TCP Spurious Retransmission] 80 → 41960 [ACK] Seq=2
 18341 29.196381698 192.168.0.104  142.250.67.78  TCP        66 41960 → 80 [ACK] Seq=1 Ack=2811895 Win=3439 Len=0 TS
 18342 29.184341945 142.250.67.78  192.168.0.104  TCP      1486 [TCP Spurious Retransmission] 80 → 41960 [ACK] Seq=2
 18343 29.184423608 192.168.0.104  142.250.67.78  TCP        68 41960 → 80 [ACK] Seq=1 Ack=2809059 Win=3439 Len=0 TS
 18344 29.189885422 142.250.67.78  192.168.0.104  TCP      1486 [TCP Retransmission] 80 → 41960 [ACK] Seq=2809059 Ac
 18345 29.196238853 142.250.67.78  192.168.0.104  TCP      1486 [TCP Retransmission] 80 → 41960 [ACK] Seq=2810477 Ac
 18346 29.196381698 192.168.0.104  142.250.67.78  TCP        68 41960 → 80 [ACK] Seq=1 Ack=2811895 Win=3439 Len=0 TS
```

```
▸ Frame 1: 1494 bytes on wire (11952 bits), 1494 bytes captured (11952 bits) on interface 0
▸ Ethernet II, Src: Tp-LinkT_57:7f:7e (7c:8b:ca:57:7f:7e), Dst: HonHaiPr_12:4a:d1 (b0:52:16:12:4a:d1)
▸ Internet Protocol Version 4, Src: 91.189.88.178, Dst: 192.168.0.104
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 57640, Seq: 1, Ack: 1, Len: 1428
  Secure Sockets Layer
```

```
0000  b0 52 16 12 4a d1 7c 8b  ca 57 7f 7e 08 00 45 00   ·R··J·|· ·W·~··E·
0010  05 c8 74 e1 40 00 31 06  59 cf 5b bd 58 b2 c0 a8   ··t·@·1· Y·[·X···
0020  00 68 01 bb e1 28 22 f7  9b db b6 6a 26 95 80 10   ·h···("· ···j&···
0030  01 f5 38 f7 00 00 01 01  08 0a 3d 9c 95 8e cc aa   ··8····· ··=·····
0040  13 3e 60 09 ee f7 6c 17  8d 7a c7 f0 9f 2d 83 e2   ·>`···l· ·z···-··
0050  d3 0b d8 80 4a 03 6e ed  b7 4f 62 a9 51 e8 f1 f6   ····J·n· ·Ob·Q···
0060  cf 56 e7 ee 1d 60 de cb  af 5b 41 e9 3c 80 46 68   ·V···`·· ·[A·<·Fh
0070  cd 28 3d ad 15 ca 95 61  8d 87 eb c0 34 fb 30 bd   ·(=····a ····4·0·
0080  9d c2 64 8f d7 d4 3c c1  f7 ae c1 d9 6a aa 3e 7d   ··d···<· ····j·>}
0090  d8 e4 1b 50 ec b8 12 1a  f2 82 1f 35 84 e5 04 97   ···P···· ···5····
00a0  a7 a9 3b 66 84 c6 bd 75  7a 77 d8 c5 28 a6 44 4a   ··;f···u zw··(·DJ
00b0  38 3c 00 d9 a6 2e 96 bb  9f 30 ad f7 ef 1c 4d 34   8<···.·· ·0····M4
00c0  14 76 8e 2b 55 7f 55 7e  f4 c9 ba 06 52 1b 49 f8   ·v·+U·U~ ····R·I·
00d0  df 9c 4b 6b a1 f9 a8 07  97 a5 95 ec 66 40 8e 3b   ··Kk···· ····f@·;
00e0  05 25 68 6b 92 67 89 37  a5 50 19 a1 cb ca 47 ed   ·%hk·g·7 ·P····G·
00f0  0d 78 9c ad a9 62 2e 1a  d2 ba b0 4f cf 8b 47 06   ·x···b.· ···O··G·
0100  d0 b9 95 6a d2 8d e0 0d  48 c3 b9 cb 81 84 f9 d6   ···j···· H·······
```

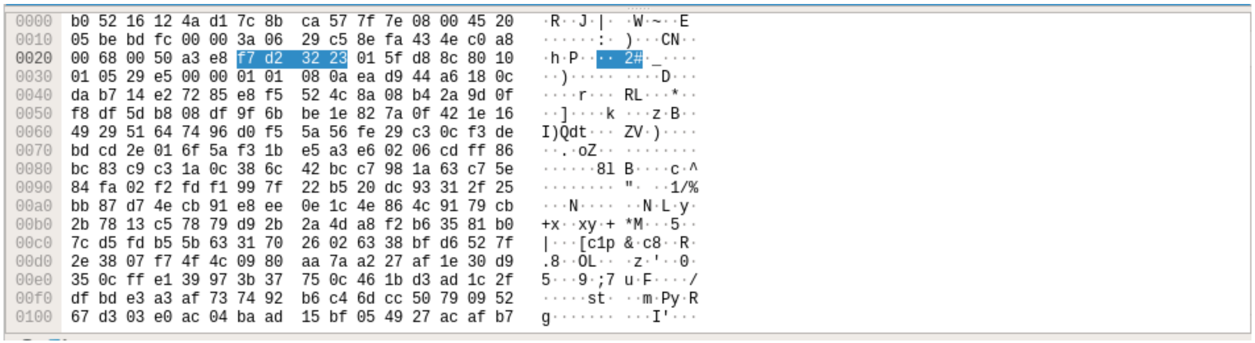
4 interfaces: <live capture in progress>   Packets: 18346 · Displayed: 18346 (100.0%)   Profile: Default

Now you can click on a packet to select it. Selecting a packet would show many information about that packet. As you can see, information about different layers of TCP/IP Protocol is listed.



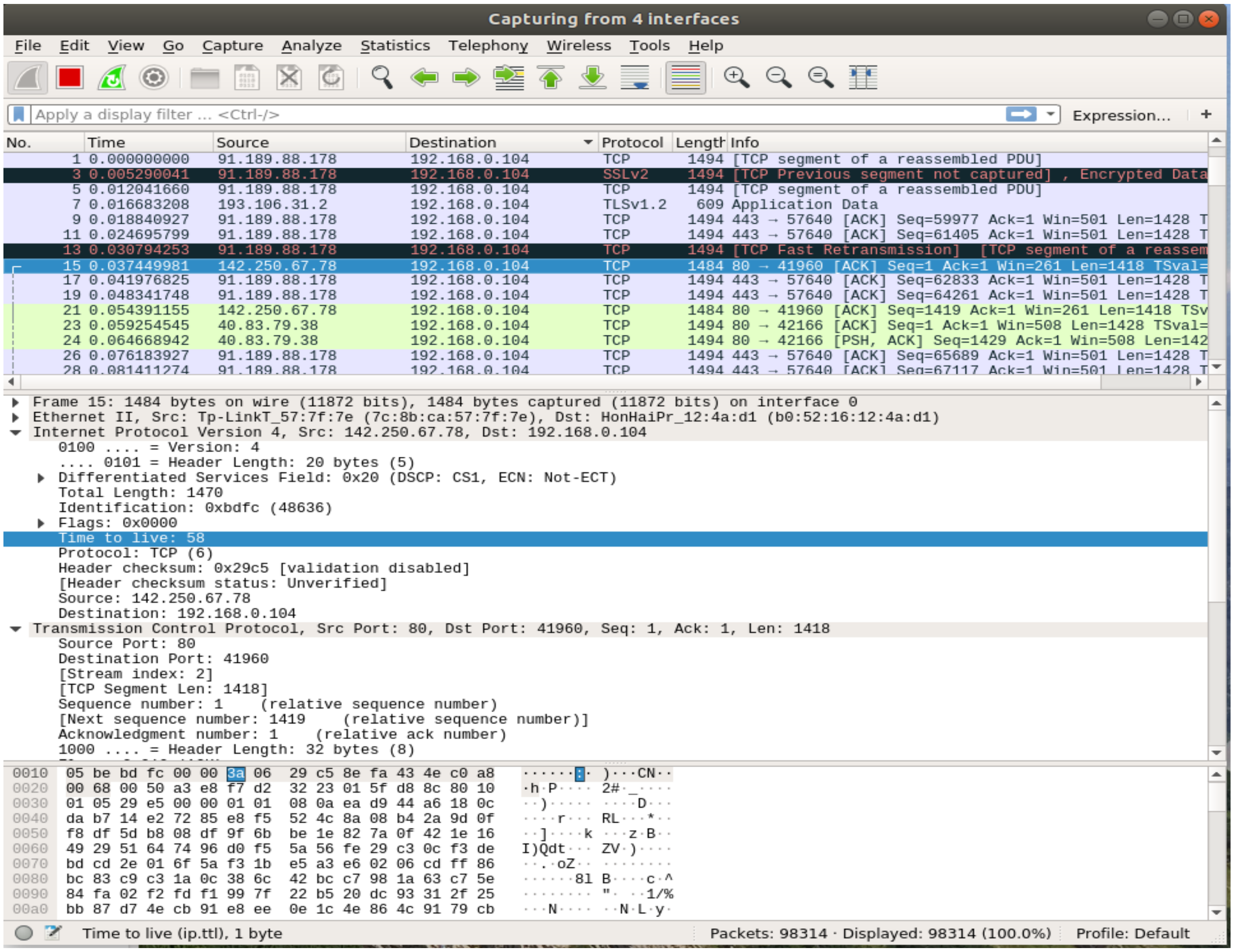
```
No.   Time          Source          Destination     Protocol Lengtł Info
    1 0.000000000   91.189.88.178   192.168.0.104   TCP      1494 [TCP segment of a reassembled PDU]
    3 0.005290041   91.189.88.178   192.168.0.104   SSLv2    1494 [TCP Previous segment not captured] , Encrypted Data
    5 0.012041660   91.189.88.178   192.168.0.104   TCP      1494 [TCP segment of a reassembled PDU]
    7 0.016683208   193.106.31.2    192.168.0.104   TLSv1.2   609 Application Data
    9 0.018840927   91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=59977 Ack=1 Win=501 Len=1428 T
   11 0.024695799   91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=61405 Ack=1 Win=501 Len=1428 T
   13 0.030794253   91.189.88.178   192.168.0.104   TCP      1494 [TCP Fast Retransmission]  [TCP segment of a reassem
   15 0.037449981   142.250.67.78   192.168.0.104   TCP      1484 80 → 41960 [ACK] Seq=1 Ack=1 Win=261 Len=1418 TSval=
   17 0.041976825   91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=62833 Ack=1 Win=501 Len=1428 T
   19 0.048341748   91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=64261 Ack=1 Win=501 Len=1428 T
   21 0.054391155   142.250.67.78   192.168.0.104   TCP      1484 80 → 41960 [ACK] Seq=1419 Ack=1 Win=261 Len=1418 TSv
   23 0.059254545   40.83.79.38     192.168.0.104   TCP      1494 80 → 42166 [ACK] Seq=1 Ack=1 Win=508 Len=1428 TSval=
   24 0.064668942   40.83.79.38     192.168.0.104   TCP      1494 80 → 42166 [PSH, ACK] Seq=1429 Ack=1 Win=508 Len=142
   26 0.076183927   91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=65689 Ack=1 Win=501 Len=1428 T
   28 0.081411274   91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=67117 Ack=1 Win=501 Len=1428 T

▸ Frame 15: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits) on interface 0
▸ Ethernet II, Src: Tp-LinkT_57:7f:7e (7c:8b:ca:57:7f:7e), Dst: HonHaiPr_12:4a:d1 (b0:52:16:12:4a:d1)
▸ Internet Protocol Version 4, Src: 142.250.67.78, Dst: 192.168.0.104
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 41960, Seq: 1, Ack: 1, Len: 1418
```

You can also see the RAW data of that particular packet.



```
0000  b0 52 16 12 4a d1 7c 8b  ca 57 7f 7e 08 00 45 20   ·R··J·|· ·W·~··E 
0010  05 be bd fc 00 00 3a 06  29 c5 8e fa 43 4e c0 a8   ······:· )···CN··
0020  00 68 00 50 a3 e8 f7 d2  32 23 01 5f d8 8c 80 10   ·h·P···· 2#·_····
0030  01 05 29 e5 00 00 01 01  08 0a ea d9 44 a6 18 0c   ··)····· ····D···
0040  da b7 14 e2 72 85 e8 f5  52 4c 8a 08 b4 2a 9d 0f   ····r··· RL···*··
0050  f8 df 5d b8 08 df 9f 6b  be 1e 82 7a 0f 42 1e 16   ··]····k ···z·B··
0060  49 29 51 64 74 96 d0 f5  5a 56 fe 29 c3 0c f3 de   I)Qdt··· ZV·)····
0070  bd cd 2e 01 6f 5a f3 1b  e5 a3 e6 02 06 cd ff 86   ··.·oZ·· ········
0080  bc 83 c9 c3 1a 0c 38 6c  42 bc c7 98 1a 63 c7 5e   ······8l B····c·^
0090  84 fa 02 f2 fd f1 99 7f  22 b5 20 dc 93 31 2f 25   ········ "· ··1/%
00a0  bb 87 d7 4e cb 91 e8 ee  0e 1c 4e 86 4c 91 79 cb   ···N···· ··N·L·y·
00b0  2b 78 13 c5 78 79 d9 2b  2a 4d a8 f2 b6 35 81 b0   +x··xy·+ *M···5··
00c0  7c d5 fd b5 5b 63 31 70  26 02 63 38 bf d6 52 7f   |···[c1p &·c8··R·
00d0  2e 38 07 f7 4f 4c 09 80  aa 7a a2 27 af 1e 30 d9   .8··OL·· ·z·'··0·
00e0  35 0c ff e1 39 97 3b 37  75 0c 46 1b d3 ad 1c 2f   5···9·;7 u·F····/
00f0  df bd e3 a3 af 73 74 92  b6 c4 6d cc 50 79 09 52   ·····st· ··m·Py·R
0100  67 d3 03 e0 ac 04 ba ad  15 bf 05 49 27 ac af b7   g······· ···I'···
```

You can also click on the arrows to expand packet data for a particular TCP/IP Protocol Layer.



Capturing from 4 interfaces
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+

```
No.   Time         Source          Destination     Protocol Length Info
    1 0.000000000  91.189.88.178   192.168.0.104   TCP      1494 [TCP segment of a reassembled PDU]
    3 0.005290041  91.189.88.178   192.168.0.104   SSLv2    1494 [TCP Previous segment not captured] , Encrypted Data
    5 0.012041660  91.189.88.178   192.168.0.104   TCP      1494 [TCP segment of a reassembled PDU]
    7 0.016683208  193.106.31.2    192.168.0.104   TLSv1.2   609 Application Data
    9 0.018840927  91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=59977 Ack=1 Win=501 Len=1428 T
   11 0.024695799  91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=61405 Ack=1 Win=501 Len=1428 T
   13 0.030794253  91.189.88.178   192.168.0.104   TCP      1494 [TCP Fast Retransmission] [TCP segment of a reassem
   15 0.037449981  142.250.67.78   192.168.0.104   TCP      1484 80 → 41960 [ACK] Seq=1 Ack=1 Win=261 Len=1418 TSval=
   17 0.041976825  91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=62833 Ack=1 Win=501 Len=1428 T
   19 0.048341748  91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=64261 Ack=1 Win=501 Len=1428 T
   21 0.054391155  142.250.67.78   192.168.0.104   TCP      1484 80 → 41960 [ACK] Seq=1419 Ack=1 Win=261 Len=1418 TSv
   23 0.059254545  40.83.79.38     192.168.0.104   TCP      1494 80 → 42166 [ACK] Seq=1 Ack=1 Win=508 Len=1428 TSval=
   24 0.064668942  40.83.79.38     192.168.0.104   TCP      1494 80 → 42166 [PSH, ACK] Seq=1429 Ack=1 Win=508 Len=142
   26 0.076183927  91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=65689 Ack=1 Win=501 Len=1428 T
   28 0.081411274  91.189.88.178   192.168.0.104   TCP      1494 443 → 57640 [ACK] Seq=67117 Ack=1 Win=501 Len=1428 T
```

```
▸ Frame 15: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits) on interface 0
▸ Ethernet II, Src: Tp-LinkT_57:7f:7e (7c:8b:ca:57:7f:7e), Dst: HonHaiPr_12:4a:d1 (b0:52:16:12:4a:d1)
▾ Internet Protocol Version 4, Src: 142.250.67.78, Dst: 192.168.0.104
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x20 (DSCP: CS1, ECN: Not-ECT)
    Total Length: 1470
    Identification: 0xbdfc (48636)
  ▸ Flags: 0x0000
    Time to live: 58
    Protocol: TCP (6)
    Header checksum: 0x29c5 [validation disabled]
    [Header checksum status: Unverified]
    Source: 142.250.67.78
    Destination: 192.168.0.104
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 41960, Seq: 1, Ack: 1, Len: 1418
    Source Port: 80
    Destination Port: 41960
    [Stream index: 2]
    [TCP Segment Len: 1418]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1419    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

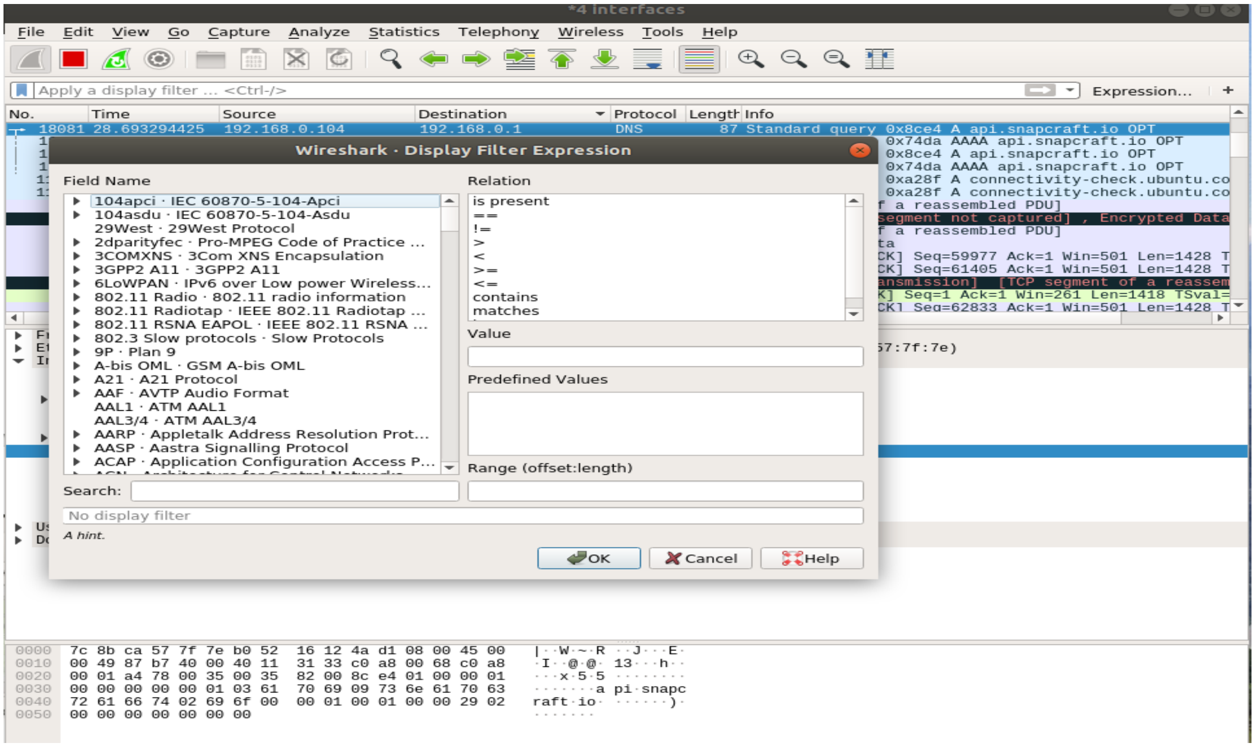
```
0010  05 be bd fc 00 00 3a 06  29 c5 8e fa 43 4e c0 a8   ······:· )···CN··
0020  00 68 00 50 a3 e8 f7 d2  32 23 01 5f d8 8c 80 10   ·h·P···· 2#·_····
0030  01 05 29 e5 00 00 01 01  08 0a ea d9 44 a6 18 0c   ··)····· ····D···
0040  da b7 14 e2 72 85 e8 f5  52 4c 8a 08 b4 2a 9d 0f   ····r··· RL···*··
0050  f8 df 5d b8 08 df 9f 6b  be 1e 82 7a 0f 42 1e 16   ··]····k ···z·B··
0060  49 29 51 64 74 96 d0 f5  5a 56 fe 29 c3 0c f3 de   I)Qdt··· ZV·)····
0070  bd cd 2e 01 6f 5a f3 1b  e5 a3 e6 02 06 cd ff 86   ··.·oZ·· ········
0080  bc 83 c9 c3 1a 0c 38 6c  42 bc c7 98 1a 63 c7 5e   ······8l B····c·^
0090  84 fa 02 f2 fd f1 99 7f  22 b5 20 dc 93 31 2f 25   ········ "· ··1/%
00a0  bb 87 d7 4e cb 91 e8 ee  0e 1c 4e 86 4c 91 79 cb   ···N···· ··N·L·y·
```

Time to live (ip.ttl), 1 byte   Packets: 98314 · Displayed: 98314 (100.0%)   Profile: Default

To filter packets, you can directly type in the filter expression in the textbox as marked in the screenshot below.

A new window should open as shown in the screenshot below. From here you can create filter expression to search packets very specifically.

In the **Field Name** section almost all the networking protocols are listed. The list is huge. You can type in what protocol you're looking for in the **Search** textbox and the **Field Name** section would show the ones that matched.

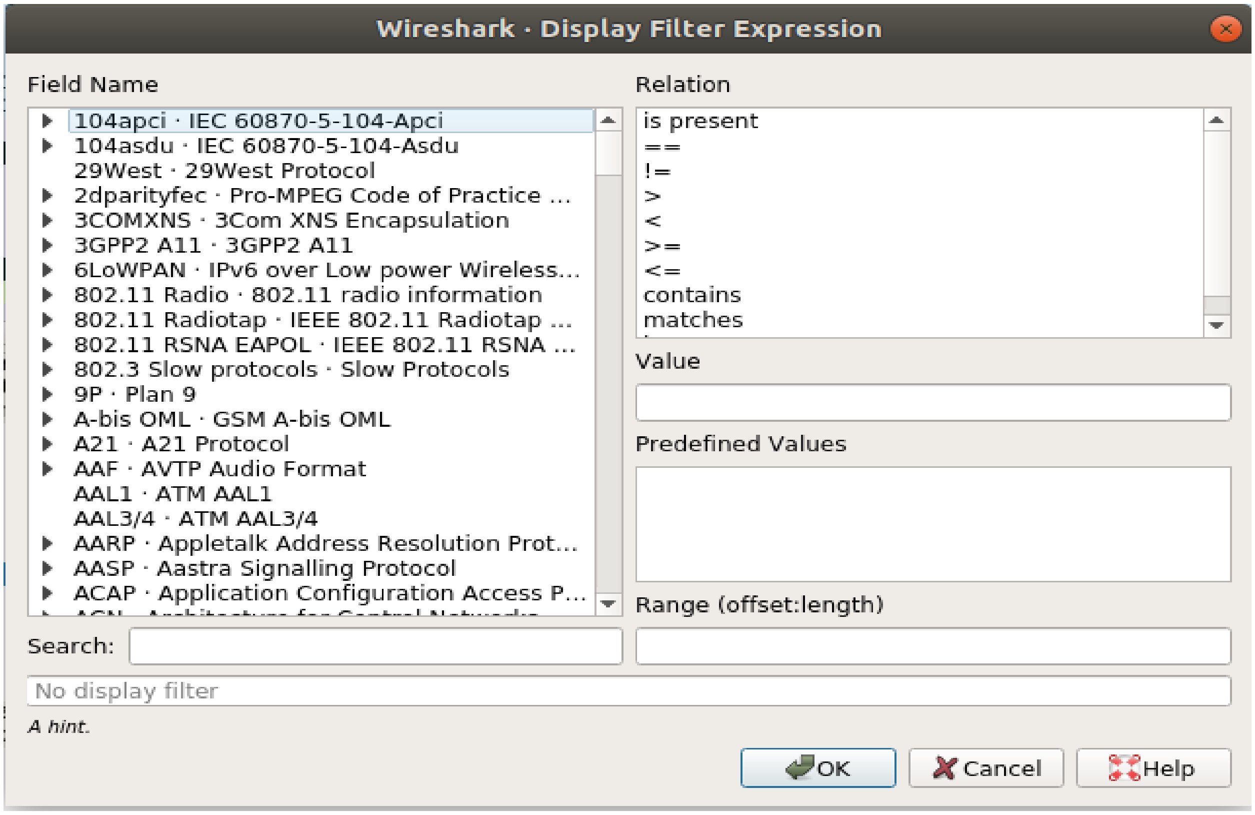

*4 interfaces
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter … <Ctrl-/>
Expression…
+
Wireshark · Display Filter Expression
Field Name
104apci · IEC 60870-5-104-Apci
104asdu · IEC 60870-5-104-Asdu
29West · 29West Protocol
2dparityfec · Pro-MPEG Code of Practice …
3COMXNS · 3Com XNS Encapsulation
3GPP2 A11 · 3GPP2 A11
6LoWPAN · IPv6 over Low power Wireless…
802.11 Radio · 802.11 radio information
802.11 Radiotap · IEEE 802.11 Radiotap …
802.11 RSNA EAPOL · IEEE 802.11 RSNA …
802.3 Slow protocols · Slow Protocols
9P · Plan 9
A-bis OML · GSM A-bis OML
A21 · A21 Protocol
AAF · AVTP Audio Format
AAL1 · ATM AAL1
AAL3/4 · ATM AAL3/4
AARP · Appletalk Address Resolution Prot…
AASP · Aastra Signalling Protocol
ACAP · Application Configuration Access P…
Relation
is present
==
!=
>
<
>=
<=
contains
matches
Value
Predefined Values
Range (offset:length)
Search:
No display filter
A hint.
OK
Cancel
Help

```
No.   Time          Source         Destination   Protocol Length Info
18081 28.693294425  192.168.0.104  192.168.0.1   DNS         87 Standard query 0x8ce4 A api.snapcraft.io OPT
                                                               0x74da AAAA api.snapcraft.io OPT
                                                               0x8ce4 A api.snapcraft.io OPT
                                                               0x74da AAAA api.snapcraft.io OPT
                                                               0xa28f A connectivity-check.ubuntu.co
                                                               0xa28f A connectivity-check.ubuntu.co
                                                              f a reassembled PDU]
                                                              segment not captured] , Encrypted Data
                                                              f a reassembled PDU]
                                                              ta
                                                              CK] Seq=59977 Ack=1 Win=501 Len=1428 T
                                                              CK] Seq=61405 Ack=1 Win=501 Len=1428 T
                                                              ansmission]  [TCP segment of a reassem
                                                              K] Seq=1 Ack=1 Win=261 Len=1418 TSval=
                                                              CK] Seq=62833 Ack=1 Win=501 Len=1428 T
```

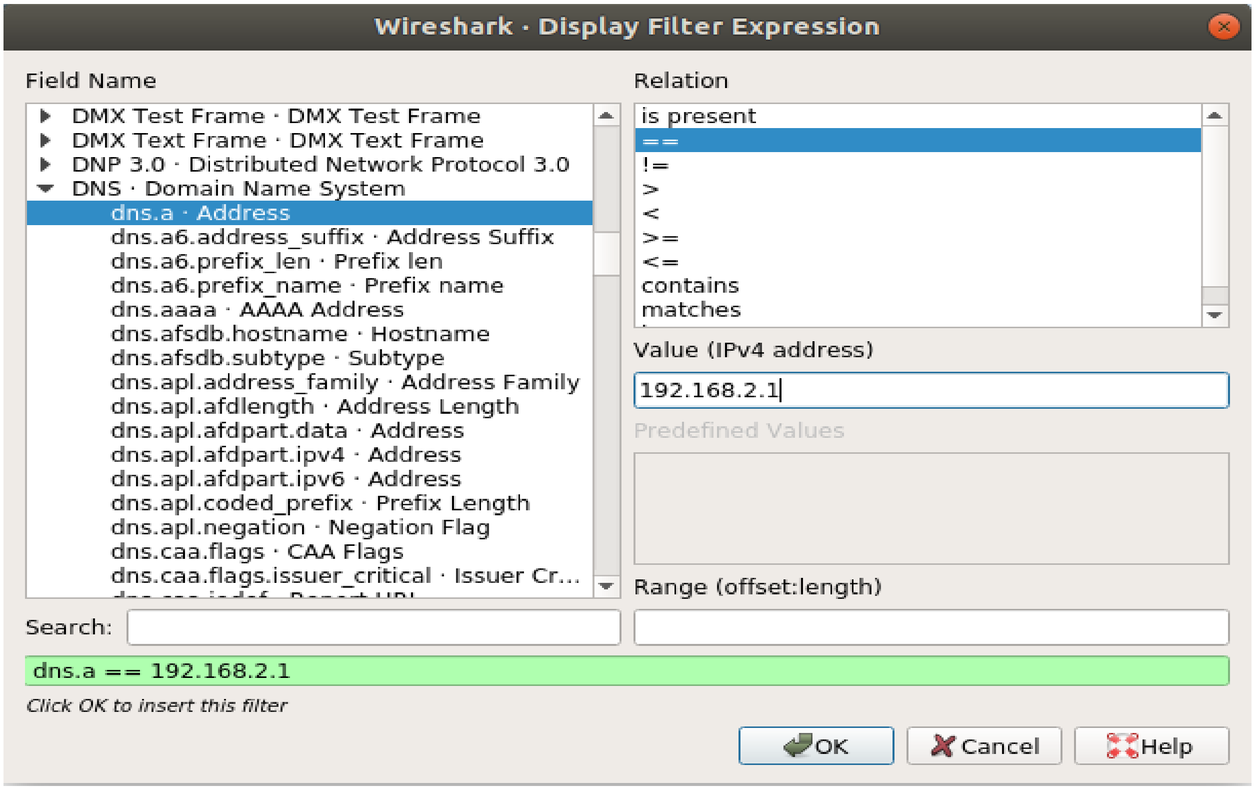
```
57:7f:7e)
```

```
0000  7c 8b ca 57 7f 7e b0 52  16 12 4a d1 08 00 45 00   |··W·~·R ··J···E·
0010  00 49 87 b7 40 00 40 11  31 33 c0 a8 00 68 c0 a8   ·I··@·@· 13···h··
0020  00 01 a4 78 00 35 00 35  82 00 8c e4 01 00 00 01   ···x·5·5 ········
0030  00 00 00 00 00 01 03 61  70 69 09 73 6e 61 70 63   ·······a pi·snapc
0040  72 61 66 74 02 69 6f 00  00 01 00 01 00 00 29 02   raft·io· ······)·
0050  00 00 00 00 00 00 00                               ·······
```

I am going to filter out all the DNS packets. So I selected **DNS Domain Name System** from the **Field Name** list. You can also click on the arrow on any protocol.


Wireshark · Display Filter Expression
Field Name
104apci · IEC 60870-5-104-Apci
104asdu · IEC 60870-5-104-Asdu
29West · 29West Protocol
2dparityfec · Pro-MPEG Code of Practice …
3COMXNS · 3Com XNS Encapsulation
3GPP2 A11 · 3GPP2 A11
6LoWPAN · IPv6 over Low power Wireless…
802.11 Radio · 802.11 radio information
802.11 Radiotap · IEEE 802.11 Radiotap …
802.11 RSNA EAPOL · IEEE 802.11 RSNA …
802.3 Slow protocols · Slow Protocols
9P · Plan 9
A-bis OML · GSM A-bis OML
A21 · A21 Protocol
AAF · AVTP Audio Format
AAL1 · ATM AAL1
AAL3/4 · ATM AAL3/4
AARP · Appletalk Address Resolution Prot…
AASP · Aastra Signalling Protocol
ACAP · Application Configuration Access P…
Relation
is present
==
!=
>
<
>=
<=
contains
matches
Value
Predefined Values
Range (offset:length)
Search:
No display filter
A hint.
OK
Cancel
Help

You can also use relational operators to test whether some field is equal to, not equal to, great than or less than some value. I searched for all the **DNS IPv4** address which is equal to **192.168.2.1** as you can see in the screenshot below.



Field Name

```
▶ DMX Test Frame · DMX Test Frame
▶ DMX Text Frame · DMX Text Frame
▶ DNP 3.0 · Distributed Network Protocol 3.0
▼ DNS · Domain Name System
    dns.a · Address
    dns.a6.address_suffix · Address Suffix
    dns.a6.prefix_len · Prefix len
    dns.a6.prefix_name · Prefix name
    dns.aaaa · AAAA Address
    dns.afsdb.hostname · Hostname
    dns.afsdb.subtype · Subtype
    dns.apl.address_family · Address Family
    dns.apl.afdlength · Address Length
    dns.apl.afdpart.data · Address
    dns.apl.afdpart.ipv4 · Address
    dns.apl.afdpart.ipv6 · Address
    dns.apl.coded_prefix · Prefix Length
    dns.apl.negation · Negation Flag
    dns.caa.flags · CAA Flags
    dns.caa.flags.issuer_critical · Issuer Cr...
```

Relation

```
is present
==
!=
>
<
>=
<=
contains
matches
```

Value (IPv4 address): 192.168.2.1

Predefined Values

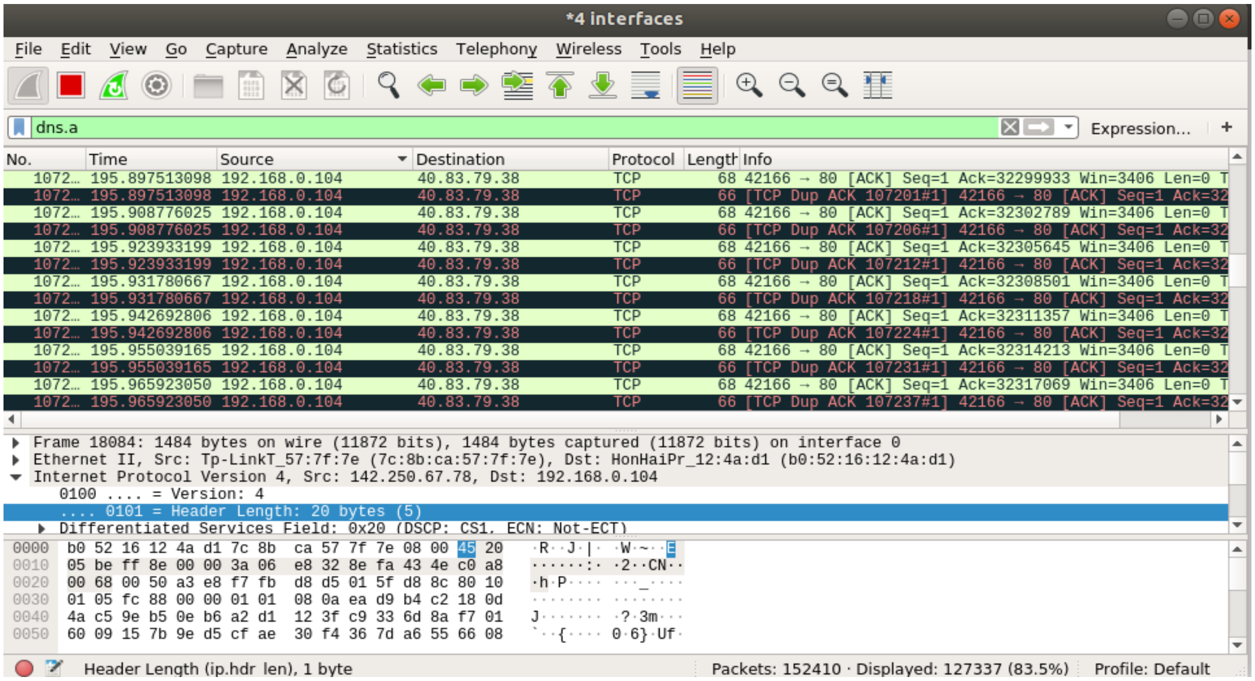
Range (offset:length)

Search:

dns.a == 192.168.2.1

Click OK to insert this filter

As you can see, only the DNS protocol packets are shown.



dns.a

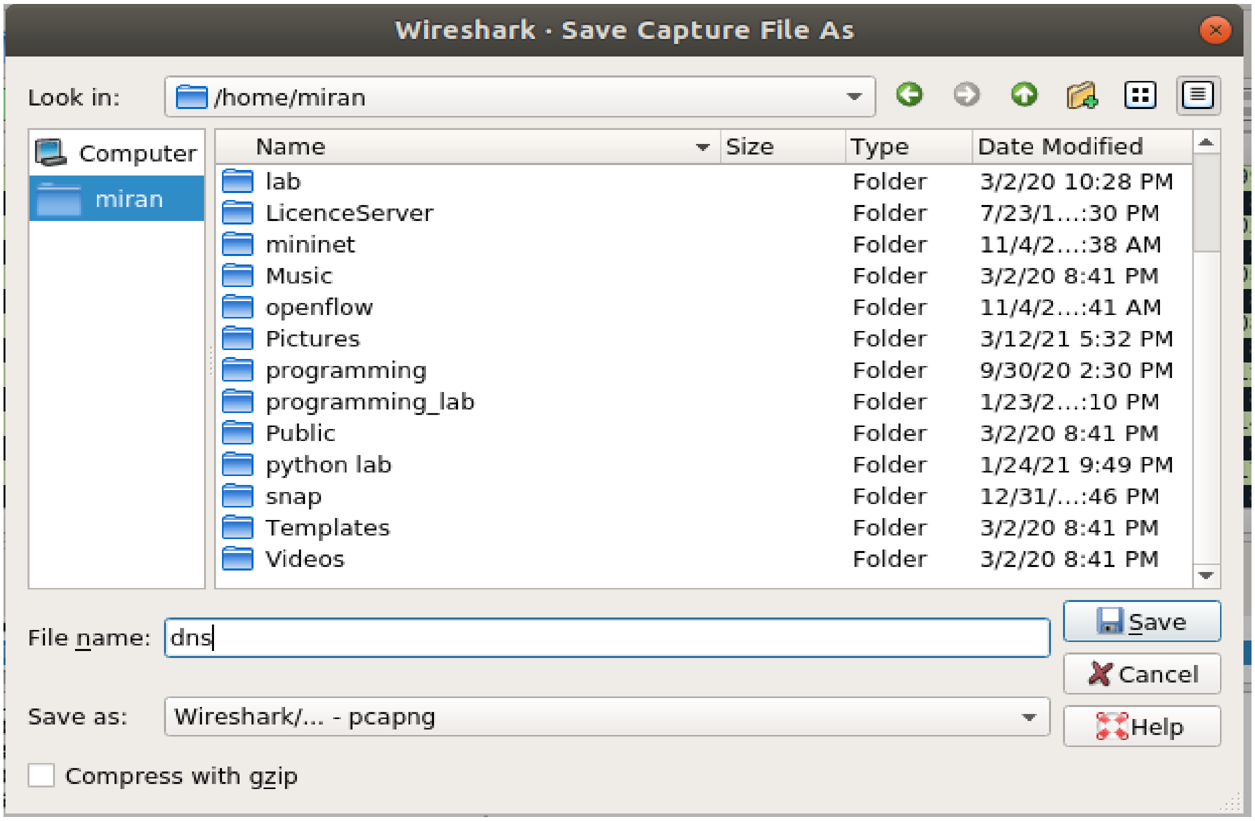
```
No.     Time           Source          Destination    Protocol Length Info
 1072… 195.897513098 192.168.0.104    40.83.79.38    TCP       68 42166 → 80 [ACK] Seq=1 Ack=32299933 Win=3406 Len=0 T
 1072… 195.897513098 192.168.0.104    40.83.79.38    TCP       66 [TCP Dup ACK 107201#1] 42166 → 80 [ACK] Seq=1 Ack=32
 1072… 195.908776025 192.168.0.104    40.83.79.38    TCP       68 42166 → 80 [ACK] Seq=1 Ack=32302789 Win=3406 Len=0 T
 1072… 195.908776025 192.168.0.104    40.83.79.38    TCP       66 [TCP Dup ACK 107206#1] 42166 → 80 [ACK] Seq=1 Ack=32
 1072… 195.923933199 192.168.0.104    40.83.79.38    TCP       68 42166 → 80 [ACK] Seq=1 Ack=32305645 Win=3406 Len=0 T
 1072… 195.923933199 192.168.0.104    40.83.79.38    TCP       66 [TCP Dup ACK 107212#1] 42166 → 80 [ACK] Seq=1 Ack=32
 1072… 195.931780667 192.168.0.104    40.83.79.38    TCP       68 42166 → 80 [ACK] Seq=1 Ack=32308501 Win=3406 Len=0 T
 1072… 195.931780667 192.168.0.104    40.83.79.38    TCP       66 [TCP Dup ACK 107218#1] 42166 → 80 [ACK] Seq=1 Ack=32
 1072… 195.942692806 192.168.0.104    40.83.79.38    TCP       68 42166 → 80 [ACK] Seq=1 Ack=32311357 Win=3406 Len=0 T
 1072… 195.942692806 192.168.0.104    40.83.79.38    TCP       66 [TCP Dup ACK 107224#1] 42166 → 80 [ACK] Seq=1 Ack=32
 1072… 195.955039165 192.168.0.104    40.83.79.38    TCP       68 42166 → 80 [ACK] Seq=1 Ack=32314213 Win=3406 Len=0 T
 1072… 195.955039165 192.168.0.104    40.83.79.38    TCP       66 [TCP Dup ACK 107231#1] 42166 → 80 [ACK] Seq=1 Ack=32
 1072… 195.965923050 192.168.0.104    40.83.79.38    TCP       68 42166 → 80 [ACK] Seq=1 Ack=32317069 Win=3406 Len=0 T
 1072… 195.965923050 192.168.0.104    40.83.79.38    TCP       66 [TCP Dup ACK 107237#1] 42166 → 80 [ACK] Seq=1 Ack=32
```

```
▶ Frame 18084: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits) on interface 0
▶ Ethernet II, Src: Tp-LinkT_57:7f:7e (7c:8b:ca:57:7f:7e), Dst: HonHaiPr_12:4a:d1 (b0:52:16:12:4a:d1)
▼ Internet Protocol Version 4, Src: 142.250.67.78, Dst: 192.168.0.104
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x20 (DSCP: CS1, ECN: Not-ECT)
```

```
0000  b0 52 16 12 4a d1 7c 8b  ca 57 7f 7e 08 00 45 20   ·R··J·|· ·W·~··E
0010  05 be ff 8e 00 00 3a 06  e8 32 8e fa 43 4e c0 a8   ······:· ·2··CN··
0020  00 68 00 50 a3 e8 f7 fb  d8 d5 01 5f d8 8c 80 10   ·h·P···· ···_····
0030  01 05 fc 88 00 00 01 01  08 0a ea d9 b4 c2 18 0d   ········ ········
0040  4a c5 9e b5 0e b6 a2 d1  12 3f c9 33 6d 8a f7 01   J······· ·?·3m···
0050  60 09 15 7b 9e d5 cf ae  30 f4 36 7d a6 55 66 08   `··{···· 0·6}·Uf·
```

Header Length (ip.hdr_len), 1 byte · Packets: 152410 · Displayed: 127337 (83.5%) · Profile: Default

You can click on the red icon as red marked in the screenshot below to stop capturing Wireshark packets.
You can click on the saved marked icon to save captured packets to a file for future use.
Now select a destination folder, type in the file name and click on **Save**.



Wireshark · Save Capture File As

Look in: /home/miran

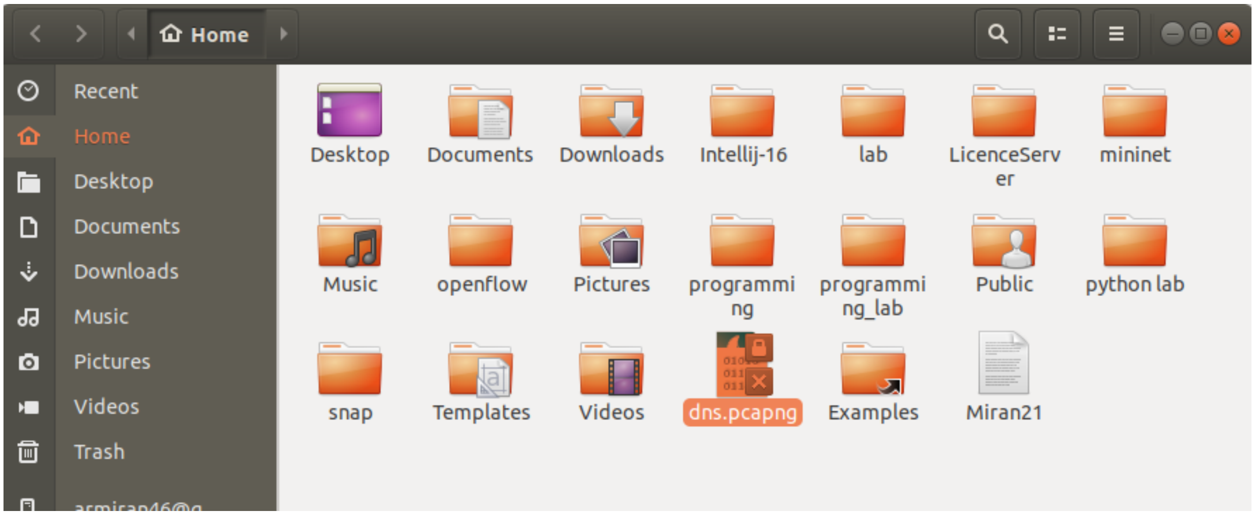
Name	Size	Type	Date Modified
lab		Folder	3/2/20 10:28 PM
LicenceServer		Folder	7/23/1...:30 PM
mininet		Folder	11/4/2...:38 AM
Music		Folder	3/2/20 8:41 PM
openflow		Folder	11/4/2...:41 AM
Pictures		Folder	3/12/21 5:32 PM
programming		Folder	9/30/20 2:30 PM
programming_lab		Folder	1/23/2...:10 PM
Public		Folder	3/2/20 8:41 PM
python lab		Folder	1/24/21 9:49 PM
snap		Folder	12/31/...:46 PM
Templates		Folder	3/2/20 8:41 PM
Videos		Folder	3/2/20 8:41 PM

File name: dns

Save as: Wireshark/... - pcapng

☐ Compress with gzip

The file should be saved.



That’s how you install and use Wireshark in Linux.

Conclusion: WireShark - a network protocol analyzer For Windows and *nix systems. And although I never used it on anything but my Windows PC, it's a really interesting tool to have installed - both for developers and curious minds. So what are some uses that might potentially benefit the people who decided to install it? Personally I have three main reasons, and i am describing those below.

Web debugging

Have you ever had issues controlling the traffic flow - maybe you were calling a service but weren't sure that the requests went the right way? What about malformed HTTP requests? If any (or both) of the situations are familiar to you, then WireShark is there to help. Although it might seem that the initial returned data set is quite complicated (and trust me, there is a lot of un-needed junk captured), you can easily set specific filters to only see what you need. In many cases it reveals details I did not expect to see. Detailed data might include source port, target port, target and source IPs as well as some details about the actual physical network controller handling the processing. Packet data is also really interesting to look at, especially if it goes through SSL -WireShark has pretty decent tools to cover those details too (just look at the hex table).

Capture interesting stuff

For example Windows Phone applications that come as XAP packages. Some time to set up an environment and you will be ready to intercept incoming content. Also, I found out some interesting stuff about an undocumented Zune API - also through inspecting existing transfer logs. It's really cool to see how a lot of content that is used on various web sites and application is in fact transmitted through open channels without any authentication necessary (even if that is present in the application itself). The fun fact is that you can use those channels for your own benefit (e.g. build third party clients for specific services).

Making sure that the right applications access the right resources
From time to time I want to make sure that every application I use, that has access to the Internet, only accesses resources it should. WireShark pretty much covers every transfer layer - of course, sometimes it is hard to see what data is passed between machines due to the fact that it is encrypted, but nonetheless, it is interesting to keep track at least where the HTTP traffic is targeted.

If you go through some packets and HTTP POST requests, you will be able to see what information is sent from your device to a remote server. For example, Rafael Riviera was able to track down the data transmitted from a Windows Phone 7 device to the Software Quality Management server in a similar manner.

WireShark is not that big and doesn't consume enormous quanitites of resources, so it runs pretty well in the background while other processes are running. I would definitely recommend to try it out, even just for fun, to see what you can get net-wise out of it.