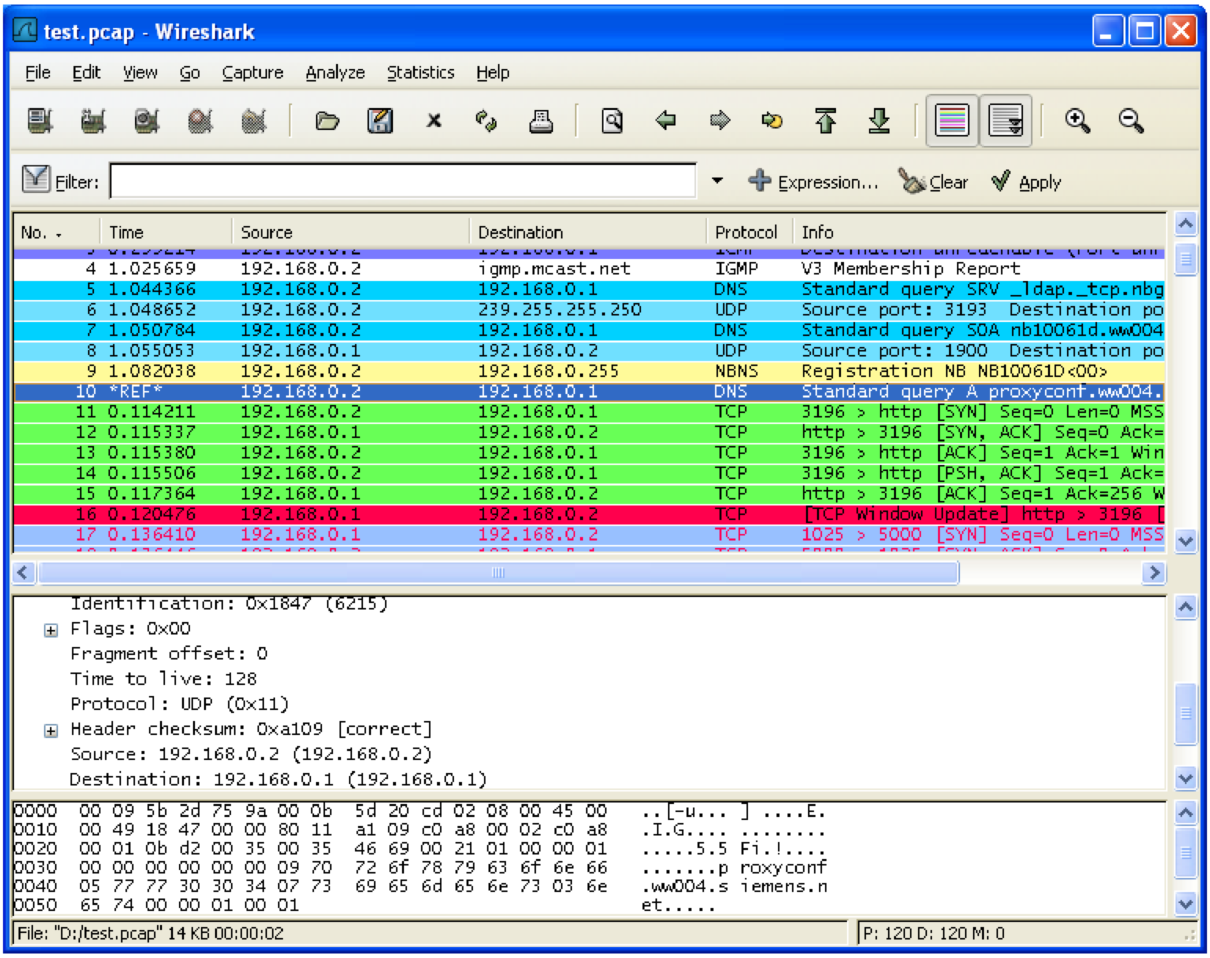
Wireshark

Capture output



No.	Time	Source	Destination	Protocol	Info
4	1.025659	192.168.0.2	igmp.mcast.net	IGMP	V3 Membership Report
5	1.044366	192.168.0.2	192.168.0.1	DNS	Standard query SRV _ldap._tcp.nbg
6	1.048652	192.168.0.2	239.255.255.250	UDP	Source port: 3193 Destination po
7	1.050784	192.168.0.2	192.168.0.1	DNS	Standard query SOA nb10061d.ww004
8	1.055053	192.168.0.1	192.168.0.2	UDP	Source port: 1900 Destination po
9	1.082038	192.168.0.2	192.168.0.255	NBNS	Registration NB NB10061D<00>
10	*REF*	192.168.0.2	192.168.0.1	DNS	Standard query A proxyconf.ww004.
11	0.114211	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0 Len=0 MSS
12	0.115337	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] Seq=0 Ack=
13	0.115380	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=1 Ack=1 Win
14	0.115506	192.168.0.2	192.168.0.1	TCP	3196 > http [PSH, ACK] Seq=1 Ack=
15	0.117364	192.168.0.1	192.168.0.2	TCP	http > 3196 [ACK] Seq=1 Ack=256 W
16	0.120476	192.168.0.1	192.168.0.2	TCP	[TCP Window Update] http > 3196 [
17	0.136410	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [SYN] Seq=0 Len=0 MSS

```
    Identification: 0x1847 (6215)
 ⊞  Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (0x11)
 ⊞  Header checksum: 0xa109 [correct]
    Source: 192.168.0.2 (192.168.0.2)
    Destination: 192.168.0.1 (192.168.0.1)
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 49 18 47 00 00 80 11  a1 09 c0 a8 00 02 c0 a8   .I.G.... ........
0020  00 01 0b d2 00 35 00 35  46 69 00 21 01 00 00 01   .....5.5 Fi.!....
0030  00 00 00 00 00 00 09 70  72 6f 78 79 63 6f 6e 66   .......p roxyconf
0040  05 77 77 30 30 34 07 73  69 65 6d 65 6e 73 03 6e   .ww004.s iemens.n
0050  65 74 00 00 01 00 01                               et.....
```

File: "D:/test.pcap" 14 KB 00:00:02 | P: 120 D: 120 M: 0

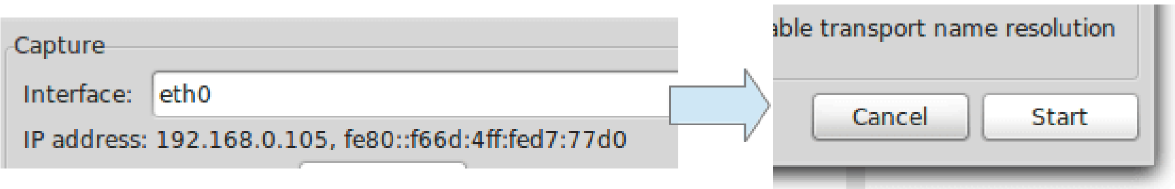
Setting Capture Options

The most useful capture options we will consider are:

1. **Network interface** – As we explained before, we will only analyze packets coming through **eth0**, either incoming or outcoming.
    2. **Capture filter** – This option allows us to indicate what kind of traffic we want to monitor by port, protocol, or type.

Before we proceed with the tips, it is important to note that some organizations forbid the use of **Wireshark** in their networks. That said, if you are not utilizing Wireshark for personal purposes make sure your organization allows its use.

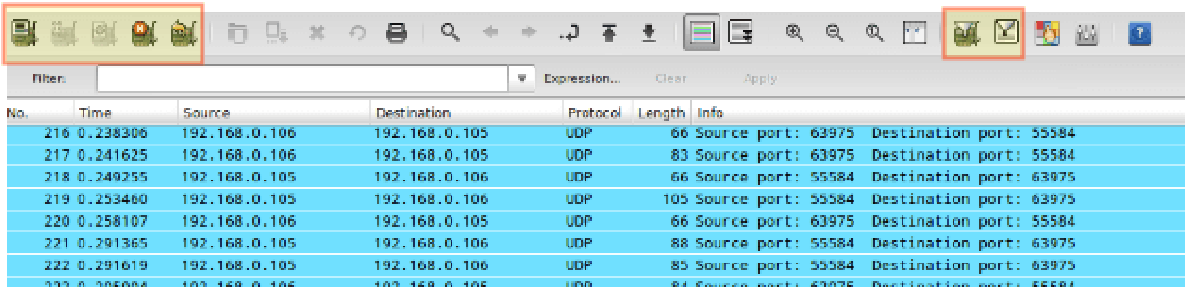
For the time being, just select `eth0` from the dropdown list and click **Start** at the button. You will start seeing all traffic passing through that interface. Not really useful for monitoring purposes due to the high amount of packets inspected, but it's a start.



Capture

Interface: eth0

IP address: 192.168.0.105, fe80::f66d:4ff:fed7:77d0



```
No.   Time      Source         Destination    Protocol Length Info
  216 0.238306  192.168.0.106  192.168.0.105  UDP          66 Source port: 63975  Destination port: 55584
  217 0.241625  192.168.0.106  192.168.0.105  UDP          83 Source port: 63975  Destination port: 55584
  218 0.249255  192.168.0.105  192.168.0.106  UDP          66 Source port: 55584  Destination port: 63975
  219 0.253460  192.168.0.105  192.168.0.106  UDP         105 Source port: 55584  Destination port: 63975
  220 0.258107  192.168.0.106  192.168.0.105  UDP          66 Source port: 63975  Destination port: 55584
  221 0.291365  192.168.0.105  192.168.0.106  UDP          88 Source port: 55584  Destination port: 63975
  222 0.291619  192.168.0.105  192.168.0.106  UDP          85 Source port: 55584  Destination port: 63975
```

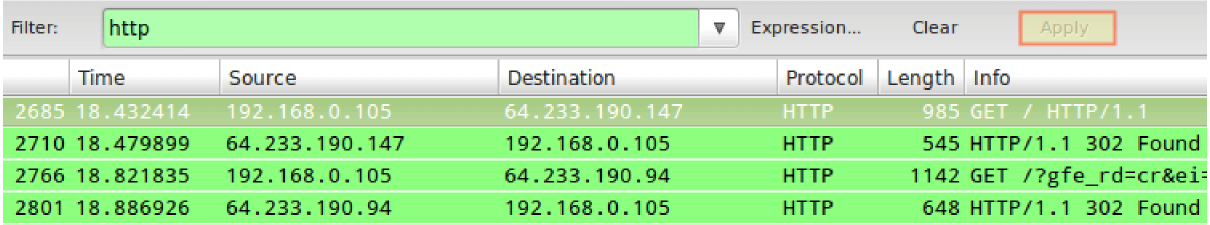
Monitor Network Interface Traffic

In the above image, we can also see the **icons** to list the available interfaces, to **stop** the current capture, and to **restart** it (red box on the **left**) and to configure and edit a filter (red box on the **right**). When you hover over one of these icons, a tooltip will be displayed to indicate what it does.

We will begin by illustrating capture options, whereas tips **#7** through **#10** will discuss how to do actually do something useful with a capture.

#1 – Inspect HTTP Traffic

Type `http` in the filter box and click **Apply**. Launch your browser and go to any site you wish:



Filter: http

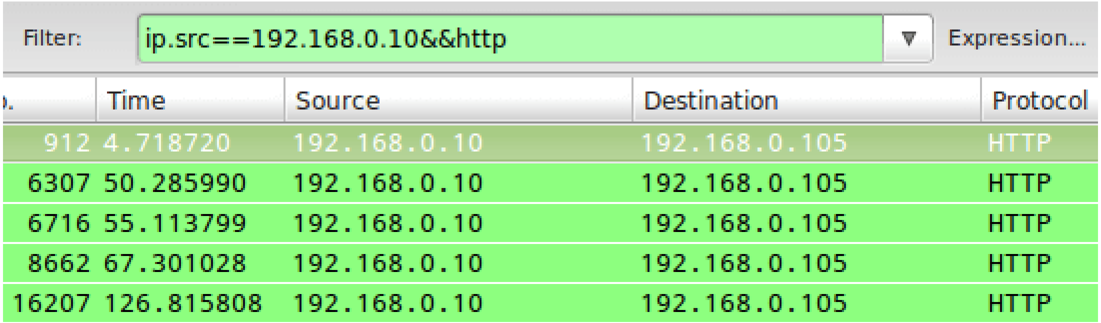
```
     Time       Source          Destination     Protocol Length Info
2685 18.432414  192.168.0.105   64.233.190.147  HTTP        985 GET / HTTP/1.1
2710 18.479899  64.233.190.147  192.168.0.105   HTTP        545 HTTP/1.1 302 Found
2766 18.821835  192.168.0.105   64.233.190.94   HTTP       1142 GET /?gfe_rd=cr&ei=
2801 18.886926  64.233.190.94   192.168.0.105   HTTP        648 HTTP/1.1 302 Found
```

Inspect HTTP Network Traffic

To begin every subsequent tip, stop the live capture and edit the capture filter.

#2 – Inspect HTTP Traffic from a Given IP Address

In this particular tip, we will prepend `ip==192.168.0.10&&` to the filter stanza to monitor HTTP traffic between the local computer and **192.168.0.10**:



Filter: ip.src==192.168.0.10&&http

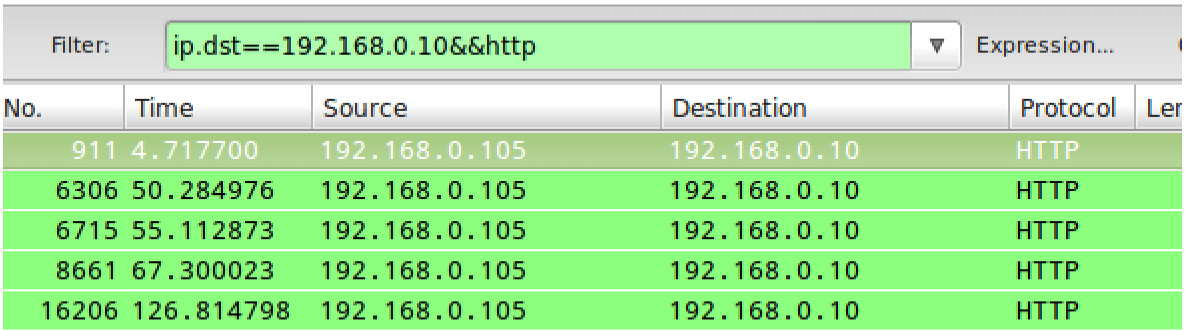
).	Time	Source	Destination	Protocol
912	4.718720	192.168.0.10	192.168.0.105	HTTP
6307	50.285990	192.168.0.10	192.168.0.105	HTTP
6716	55.113799	192.168.0.10	192.168.0.105	HTTP
8662	67.301028	192.168.0.10	192.168.0.105	HTTP
16207	126.815808	192.168.0.10	192.168.0.105	HTTP

Inspect HTTP Traffic on IP Address

#3 – Inspect HTTP Traffic to a Given IP Address

Closely related with **#2**, in this case, we will use `ip.dst` as part of the capture filter as follows:

```
ip.dst==192.168.0.10&&http
```



Filter: ip.dst==192.168.0.10&&http

No.	Time	Source	Destination	Protocol	Ler
911	4.717700	192.168.0.105	192.168.0.10	HTTP	
6306	50.284976	192.168.0.105	192.168.0.10	HTTP	
6715	55.112873	192.168.0.105	192.168.0.10	HTTP	
8661	67.300023	192.168.0.105	192.168.0.10	HTTP	
16206	126.814798	192.168.0.105	192.168.0.10	HTTP	

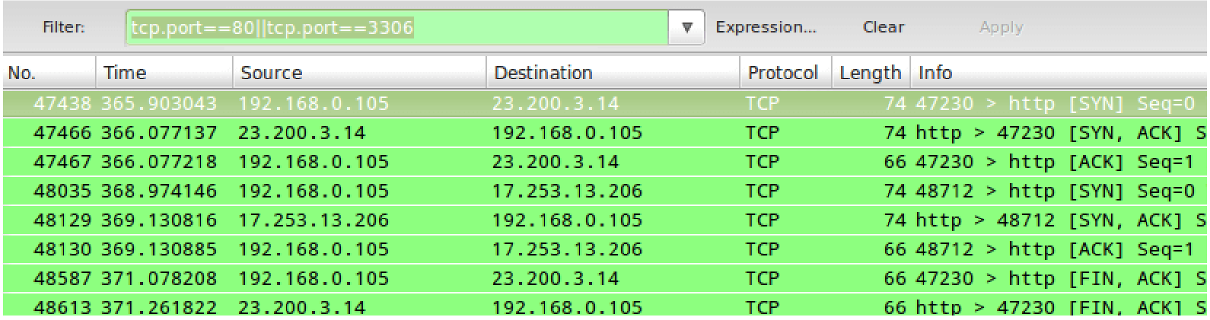
Monitor HTTP Network Traffic to IP Address

To combine tips **#2** and **#3**, you can use `ip.addr` in the filter rule instead of `ip.src` or `ip.dst`.

#4 – Monitor Apache and MySQL Network Traffic

Sometimes you will be interested in inspecting traffic that matches either (or both) conditions whatsoever. For example, to monitor traffic on TCP ports **80** (webserver) and **3306** (MySQL / MariaDB database server), you can use an `OR` condition in the capture filter:

```
tcp.port==80||tcp.port==3306
```



Filter: tcp.port==80||tcp.port==3306

No.	Time	Source	Destination	Protocol	Length	Info
47438	365.903043	192.168.0.105	23.200.3.14	TCP	74	47230 > http [SYN] Seq=0
47466	366.077137	23.200.3.14	192.168.0.105	TCP	74	http > 47230 [SYN, ACK] S
47467	366.077218	192.168.0.105	23.200.3.14	TCP	66	47230 > http [ACK] Seq=1
48035	368.974146	192.168.0.105	17.253.13.206	TCP	74	48712 > http [SYN] Seq=0
48129	369.130816	17.253.13.206	192.168.0.105	TCP	74	http > 48712 [SYN, ACK] S
48130	369.130885	192.168.0.105	17.253.13.206	TCP	66	48712 > http [ACK] Seq=1
48587	371.078208	192.168.0.105	23.200.3.14	TCP	66	47230 > http [FIN, ACK] S
48613	371.261822	23.200.3.14	192.168.0.105	TCP	66	http > 47230 [FIN, ACK] S

Monitor Apache and MySQL Traffic

In tips **#2** and **#3**, `||` and the word **or** produce the same results. Same with `&&` and the word **and**.

TIP #5 – Reject Packets to Given IP Address

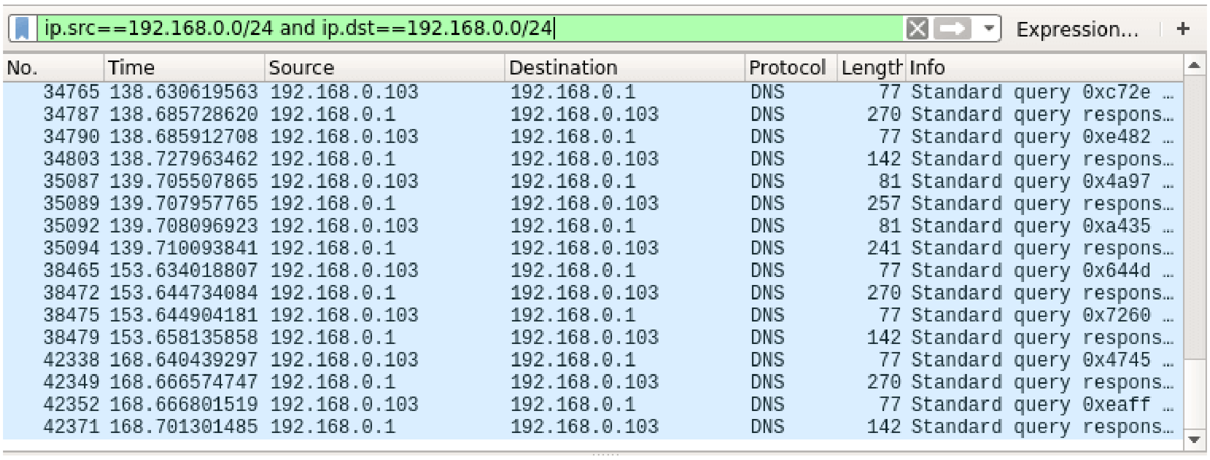
To exclude packets not matching the filter rule, use `!` and enclose the rule within parentheses. For example, to exclude packages originating from or being directed to a given IP address, you can use:

```
!(ip.addr == 192.168.0.10)
```

TIP #6 – Monitor Local Network Traffic (192.168.0.0/24)

The following filter rule will display only local traffic and exclude packets going to and coming from the Internet:

```
ip.src==192.168.0.0/24 and ip.dst==192.168.0.0/24
```



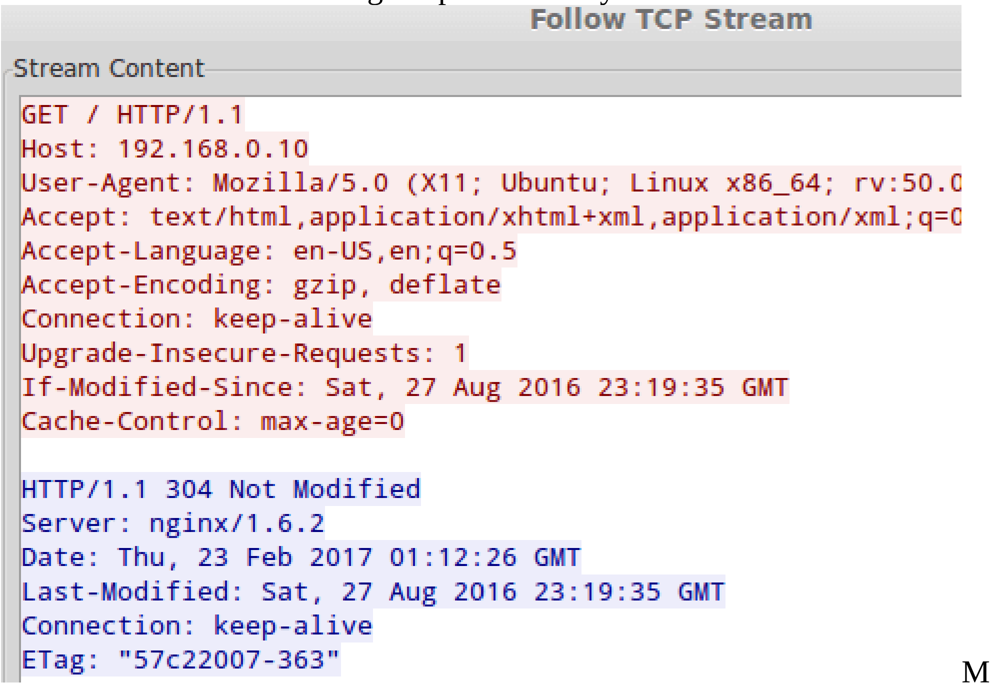
```
ip.src==192.168.0.0/24 and ip.dst==192.168.0.0/24
```

```
No.   Time          Source          Destination     Protocol Length Info
34765 138.630619563 192.168.0.103   192.168.0.1     DNS       77 Standard query 0xc72e …
34787 138.685728620 192.168.0.1     192.168.0.103   DNS      270 Standard query respons…
34790 138.685912708 192.168.0.103   192.168.0.1     DNS       77 Standard query 0xe482 …
34803 138.727963462 192.168.0.1     192.168.0.103   DNS      142 Standard query respons…
35087 139.705507865 192.168.0.103   192.168.0.1     DNS       81 Standard query 0x4a97 …
35089 139.707957765 192.168.0.1     192.168.0.103   DNS      257 Standard query respons…
35092 139.708096923 192.168.0.103   192.168.0.1     DNS       81 Standard query 0xa435 …
35094 139.710093841 192.168.0.1     192.168.0.103   DNS      241 Standard query respons…
38465 153.634018807 192.168.0.103   192.168.0.1     DNS       77 Standard query 0x644d …
38472 153.644734084 192.168.0.1     192.168.0.103   DNS      270 Standard query respons…
38475 153.644904181 192.168.0.103   192.168.0.1     DNS       77 Standard query 0x7260 …
38479 153.658135858 192.168.0.1     192.168.0.103   DNS      142 Standard query respons…
42338 168.640439297 192.168.0.103   192.168.0.1     DNS       77 Standard query 0x4745 …
42349 168.666574747 192.168.0.1     192.168.0.103   DNS      270 Standard query respons…
42352 168.666801519 192.168.0.103   192.168.0.1     DNS       77 Standard query 0xeaff …
42371 168.701301485 192.168.0.1     192.168.0.103   DNS      142 Standard query respons…
```

Monitor Local Network Traffic

TIP #7 – Monitor the Contents of a TCP Conversation

To inspect the contents of a **TCP** conversation (data exchange), right-click on a given packet and choose Follow **TCP** stream. A window will pop-up with the content of the conversation. This will include **HTTP** headers if we are inspecting web traffic, and also any plain text credentials transmitted during the process if any.



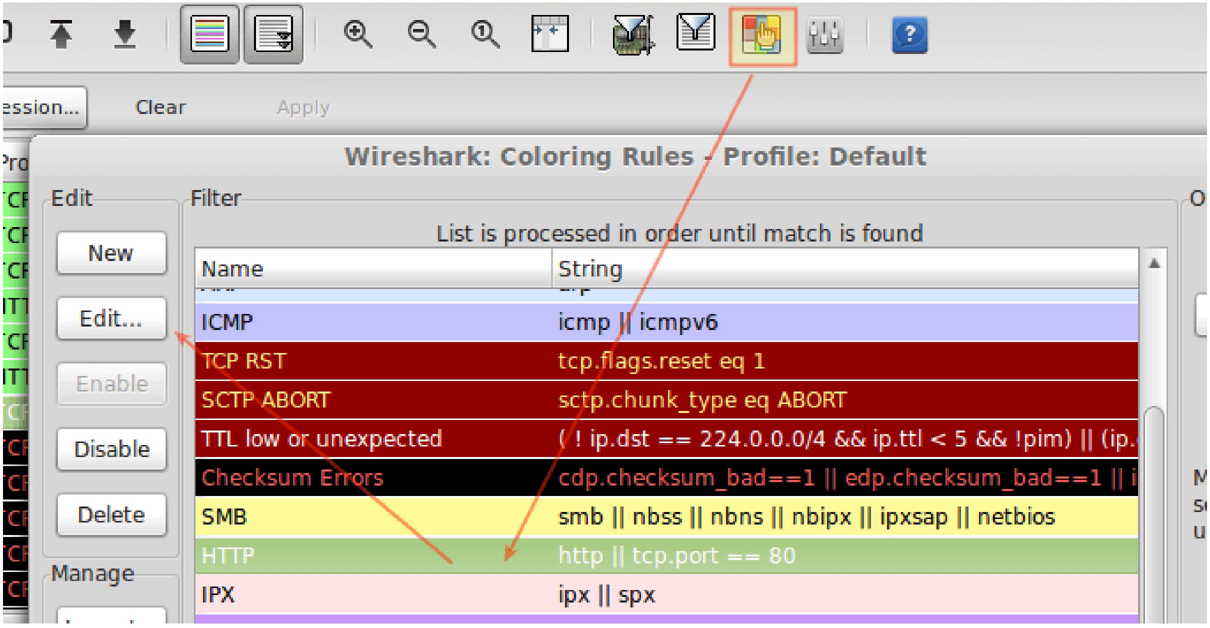
```
GET / HTTP/1.1
Host: 192.168.0.10
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Sat, 27 Aug 2016 23:19:35 GMT
Cache-Control: max-age=0

HTTP/1.1 304 Not Modified
Server: nginx/1.6.2
Date: Thu, 23 Feb 2017 01:12:26 GMT
Last-Modified: Sat, 27 Aug 2016 23:19:35 GMT
Connection: keep-alive
ETag: "57c22007-363"
```

Monitor TCP Conversation

TIP #8 – Edit Coloring Rules

By now I am sure you already noticed that each row in the capture window is colored. By default, **HTTP** traffic appears in the **green** background with black text,
whereas **checksum** errors are shown in **red** text with a black background.
If you wish to change these settings, click the **Edit** coloring rules icon, choose a given filter, and click **Edit**.



Wireshark: Coloring Rules - Profile: Default

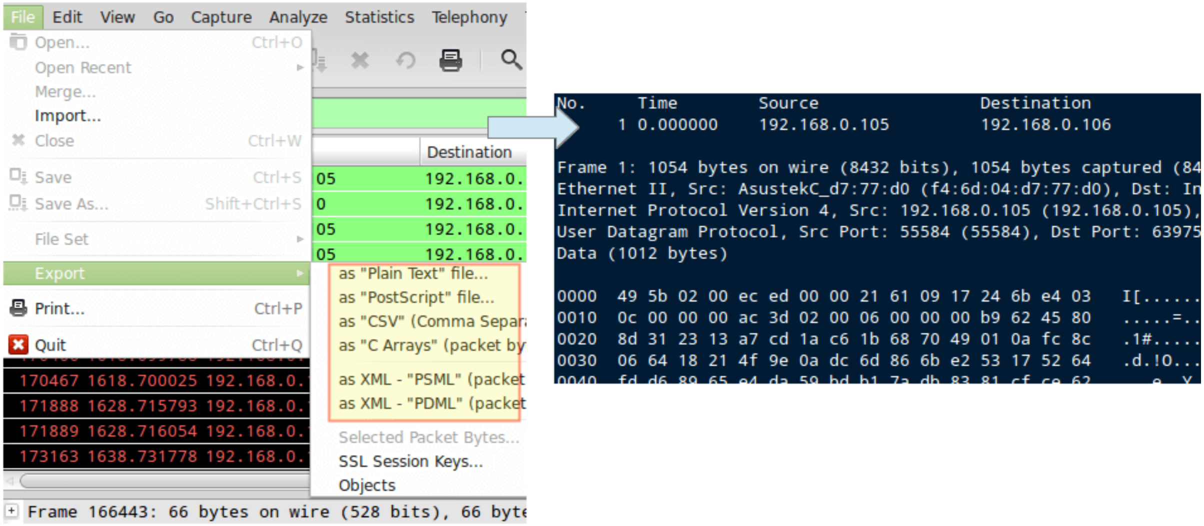
List is processed in order until match is found

Name	String
ICMP	icmp || icmpv6
TCP RST	tcp.flags.reset eq 1
SCTP ABORT	sctp.chunk_type eq ABORT
TTL low or unexpected	(! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim) || (ip.
Checksum Errors	cdp.checksum_bad==1 || edp.checksum_bad==1 || i
SMB	smb || nbss || nbns || nbipx || ipxsap || netbios
HTTP	http || tcp.port == 80
IPX	ipx || spx

Customize Wireshark Output in Colors

TIP #9 – Save the Capture to a File

Saving the contents of capture will allow us to be able to inspect it with greater detail. To do this, go to **File** → **Export** and choose an export format from the list:



```
05  192.168.0.
0   192.168.0.
05  192.168.0.
05  192.168.0.
170467 1618.700025 192.168.0.
171888 1628.715793 192.168.0.
171889 1628.716054 192.168.0.
173163 1638.731778 192.168.0.
Frame 166443: 66 bytes on wire (528 bits), 66 byte
```

```
No.     Time          Source                Destination
      1 0.000000      192.168.0.105         192.168.0.106

Frame 1: 1054 bytes on wire (8432 bits), 1054 bytes captured (84
Ethernet II, Src: AsustekC_d7:77:d0 (f4:6d:04:d7:77:d0), Dst: In
Internet Protocol Version 4, Src: 192.168.0.105 (192.168.0.105),
User Datagram Protocol, Src Port: 55584 (55584), Dst Port: 63975
Data (1012 bytes)

0000  49 5b 02 00 ec ed 00 00 21 61 09 17 24 6b e4 03   I[......
0010  0c 00 00 00 ac 3d 02 00 06 00 00 00 b9 62 45 80   .....=..
0020  8d 31 23 13 a7 cd 1a c6 1b 68 70 49 01 0a fc 8c   .1#.....
0030  06 64 18 21 4f 9e 0a dc 6d 86 6b e2 53 17 52 64   .d.!O...
0040  fd d6 89 65 e4 da 59 bd b1 7a db 83 81 cf ce 62
```