



Mawlana Bhashani Science And Technology University

Lab-Report

Lab Report No: 08

Lab Report Name: Install and use wireshark on linux operating system

Group member ID: IT-18013 and IT-18028

Date of Performance: 30-06-2021

Date of Submission: 30-06-2021

Submitted by

Name: Anjom Nour Anika & Fatema-tuz-jannat

ID: IT-18013 & IT-18028.

3rd Year 2nd Semester

Session: 2017-2018

Dept of ICT MBSTU

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

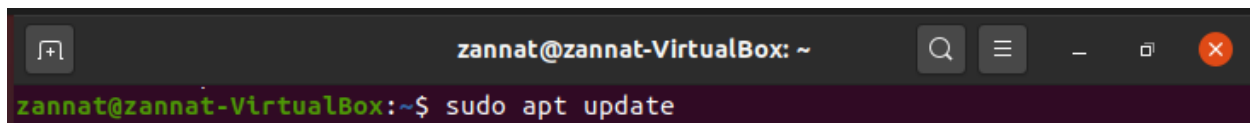
MBSTU.

Installing WIRESHARK:

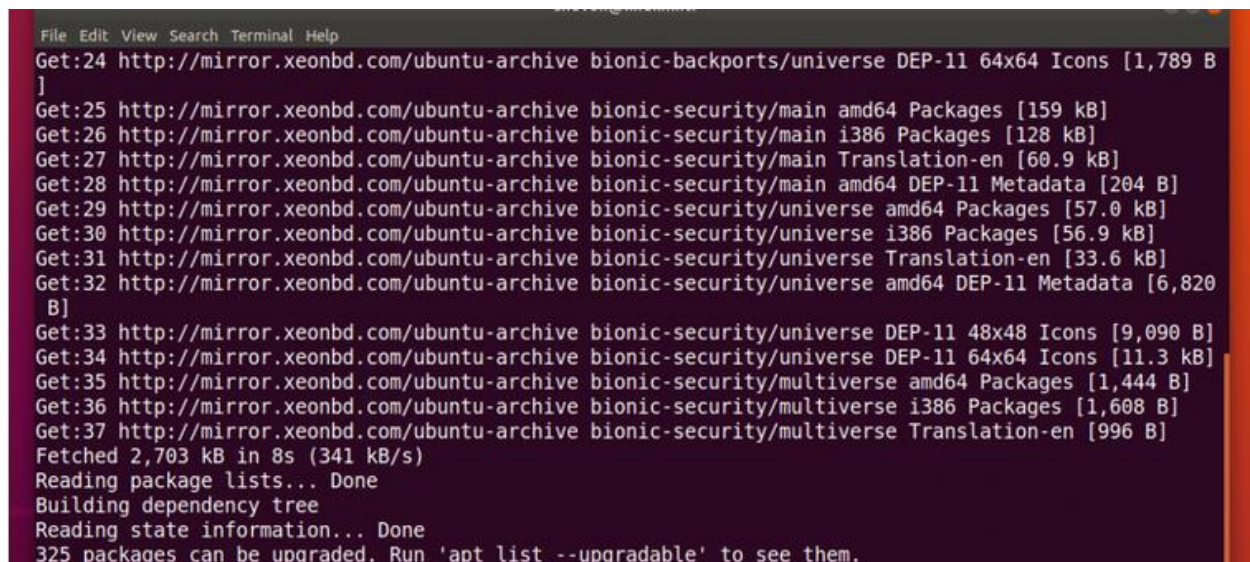
Wireshark is available in the official package repository of Ubuntu 14.04 LTS and later. So it is really easy to install.

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

```
$ sudo apt update
```

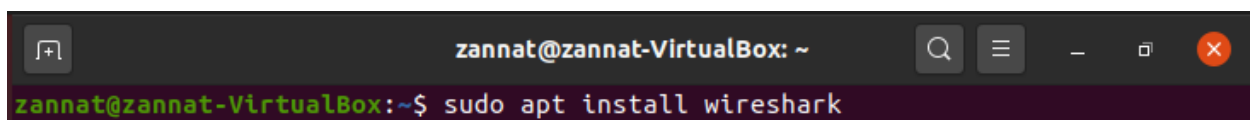
A terminal window titled 'zannat@zannat-VirtualBox: ~' with a search icon, menu icon, and window control buttons. The command 'zannat@zannat-VirtualBox:~\$ sudo apt update' is entered and executed.

The APT package repository cache should be updated.

A terminal window showing the output of the 'sudo apt update' command. The output lists various package repositories and their contents, including bionic-backports/universe, bionic-security/main, bionic-security/universe, and bionic-security/multiverse. It shows the number of packages and their sizes for each repository. The total size of the updated cache is 2,703 kB. The terminal also shows the status of the package lists and the dependency tree.

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

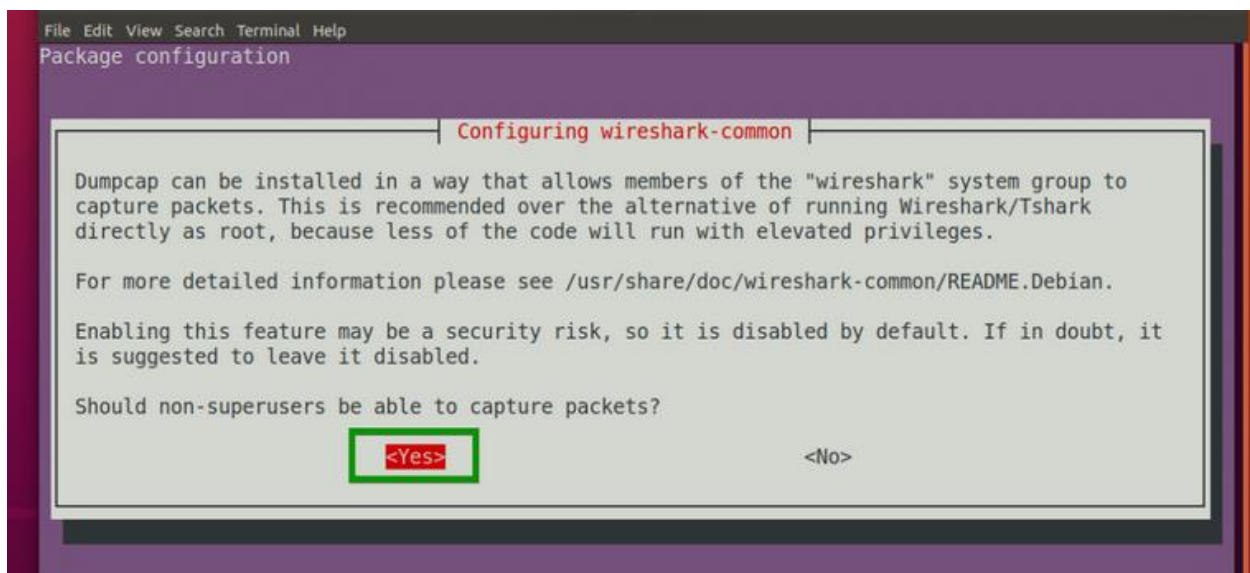
```
$ sudo apt install wireshark
```

A terminal window titled 'zannat@zannat-VirtualBox: ~' with a search icon, menu icon, and window control buttons. The command 'zannat@zannat-VirtualBox:~\$ sudo apt install wireshark' is entered and executed.

Now press **y** and then press **<Enter>**.

```
The following NEW packages will be installed:
geoip-database-extra javascript-common libc-ares2 libdouble-conversion1 libjs-openlayers
liblua5.2-0 libnl-route-3-200 libqt5core5a libqt5dbus5 libqt5gui5 libqt5multimedia5
libqt5network5 libqt5printsupport5 libqt5svg5 libqt5widgets5 libsmi2ldbl libsnappy1v5
libspandsp2 libssh-gcrypt-4 libwireshark-data libwireshark10 libwiretap7 libwscodecs1 libwsutil8
libxcb-xinerama0 qt5-gtk-platformtheme qttranslations5-l10n wireshark wireshark-common
wireshark-qt
0 upgraded, 30 newly installed, 0 to remove and 325 not upgraded.
Need to get 41.0 MB of archives.
After this operation, 181 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

By default, Wireshark must be started as root privileges in order to work. If you want to run wireshark without root privileges or without sudo, then select <Yes> and print <Enter>.



Wireshark should be installed.

```
File Edit View Search Terminal Help
Processing triggers for gnome-menus (3.13.3-1ubuntu1) ...
Setting up javascript-common (11) ...
Setting up libwscodecsl:amd64 (2.4.5-1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Setting up libc-ares2:amd64 (1.14.0-1) ...
Setting up libqt5core5a:amd64 (5.9.5+dfsg-0ubuntu1) ...
Setting up libqt5dbus5:amd64 (5.9.5+dfsg-0ubuntu1) ...
Setting up libqt5network5:amd64 (5.9.5+dfsg-0ubuntu1) ...
Setting up libqt5gui5:amd64 (5.9.5+dfsg-0ubuntu1) ...
Setting up libwireshark10:amd64 (2.4.5-1) ...
Setting up qt5-gtk-platformtheme:amd64 (5.9.5+dfsg-0ubuntu1) ...
Setting up libqt5widgets5:amd64 (5.9.5+dfsg-0ubuntu1) ...
Setting up wireshark-common (2.4.5-1) ...
Setting up libqt5sprintsupport5:amd64 (5.9.5+dfsg-0ubuntu1) ...
Setting up libqt5multimedia5:amd64 (5.9.5-0ubuntu1) ...
Setting up libqt5svg5:amd64 (5.9.5-0ubuntu1) ...
Setting up wireshark-qt (2.4.5-1) ...
Setting up wireshark (2.4.5-1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
```

Now if you selected **<Yes>** in the earlier section to run Wireshark without root access, then run the following command to add your user to the **wireshark** group:

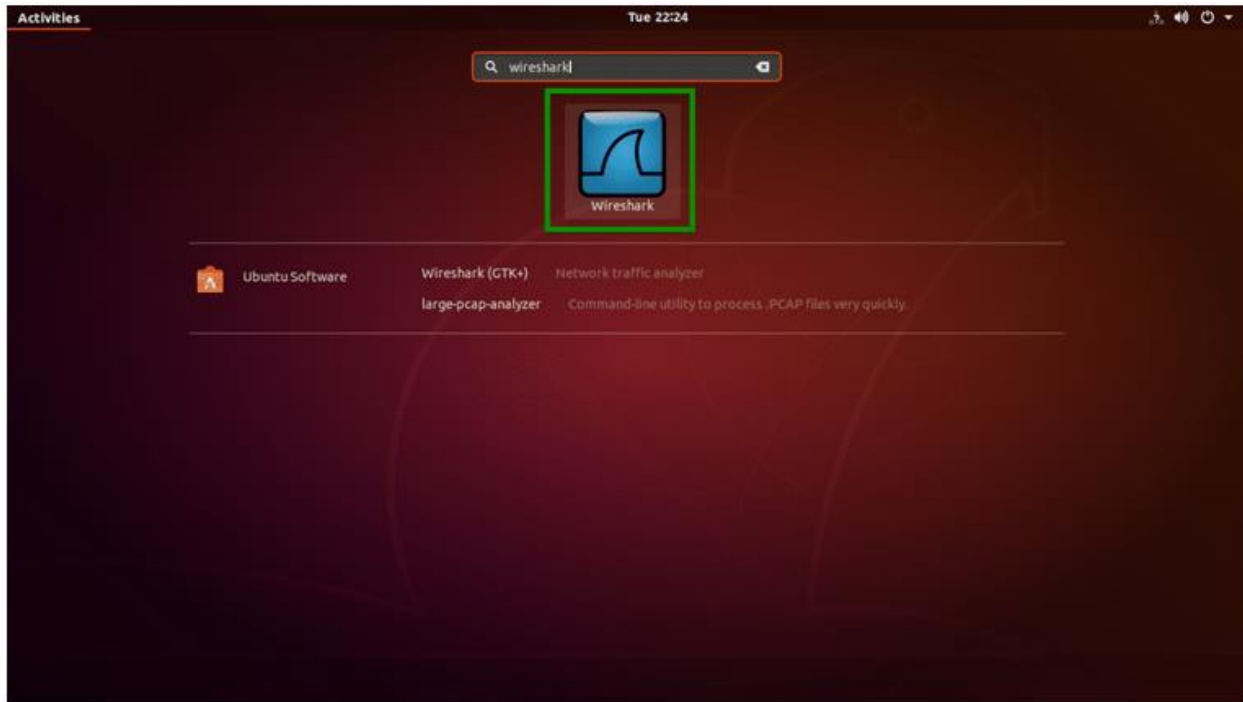
```
$ sudo usermod -aG wireshark $(whoami)
```

Finally, reboot your computer with the following command:

```
$ sudo reboot
```

Starting Wireshark:

Now that Wireshark is installed, you can start Wireshark from the **Application Menu** of Ubuntu.



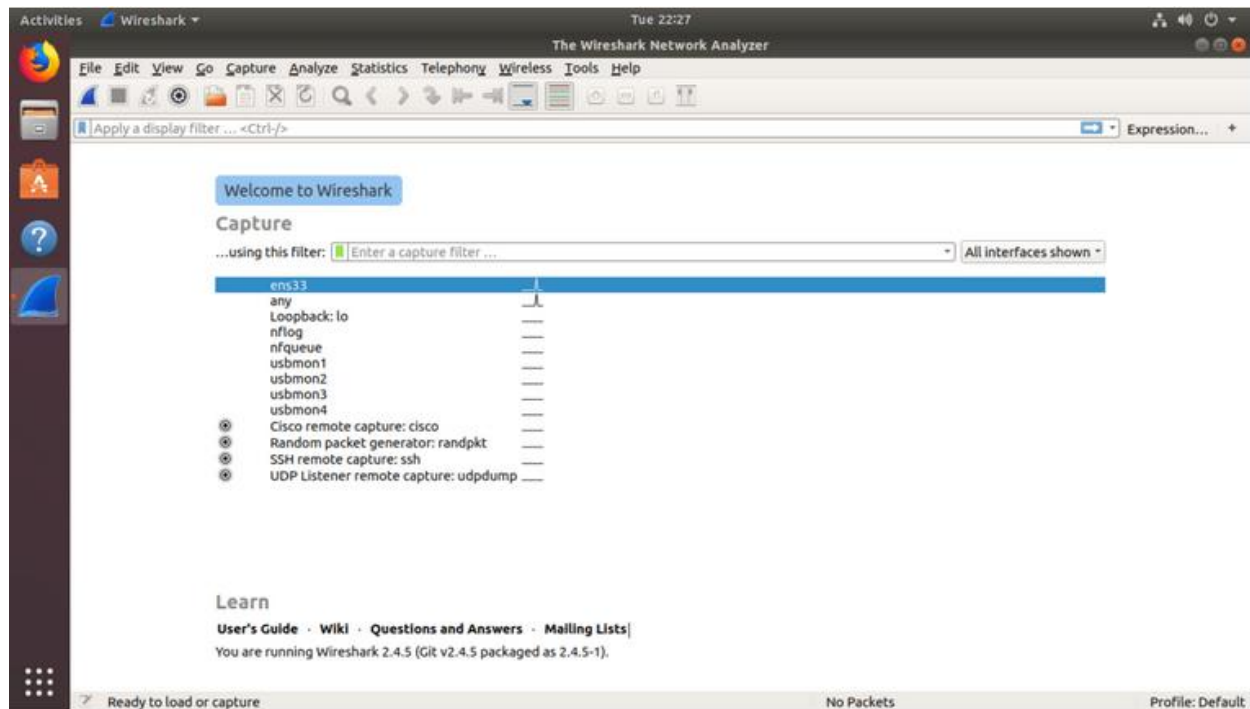
You can also run the following command to start Wireshark from the Terminal:

```
$ wireshark
```

If you did not enable Wireshark to run without **root** privileges or **sudo**, then the command should be:

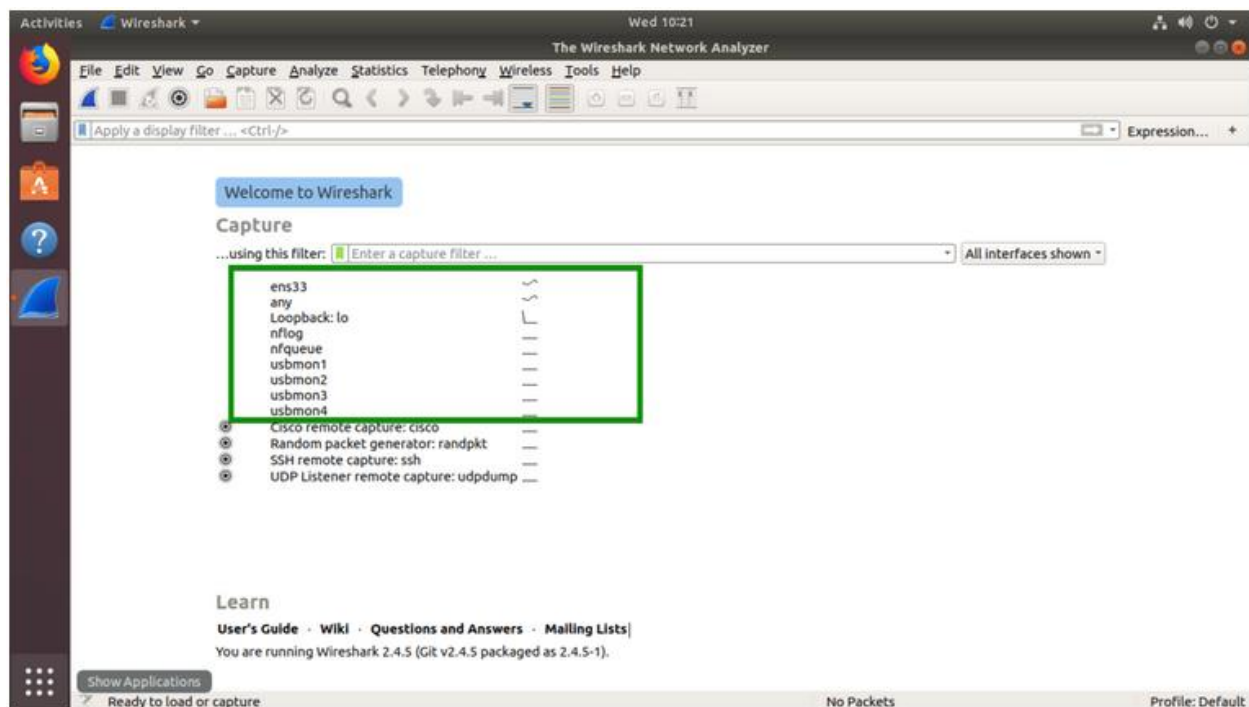
```
$ sudo wireshark
```

Wireshark should start.

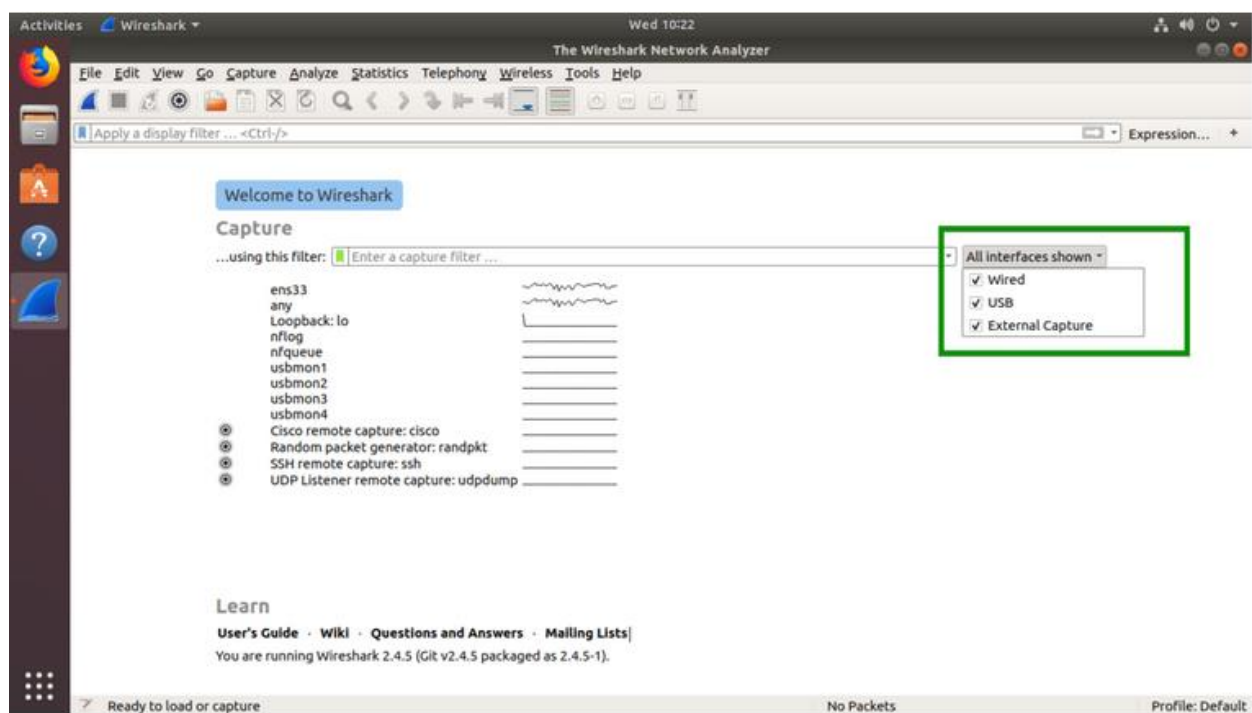


Capturing Packets Using Wireshark:

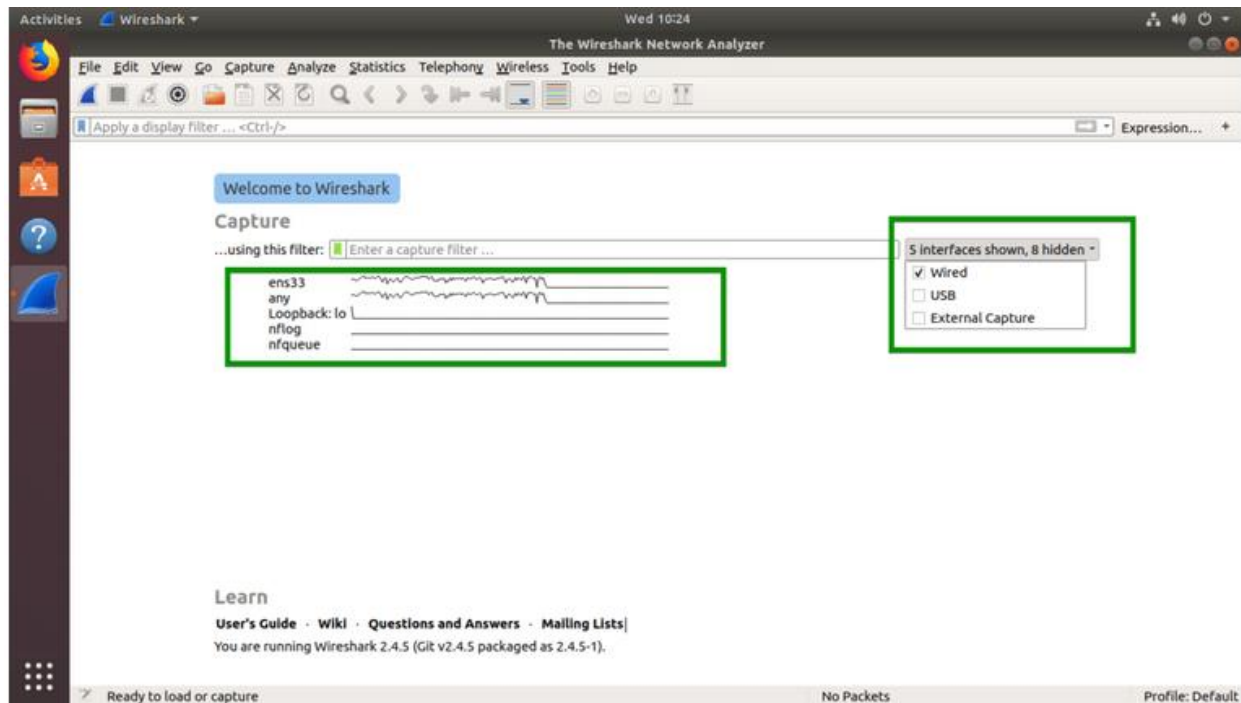
When you start Wireshark, you will see a list of interfaces that you can capture packets to and from.



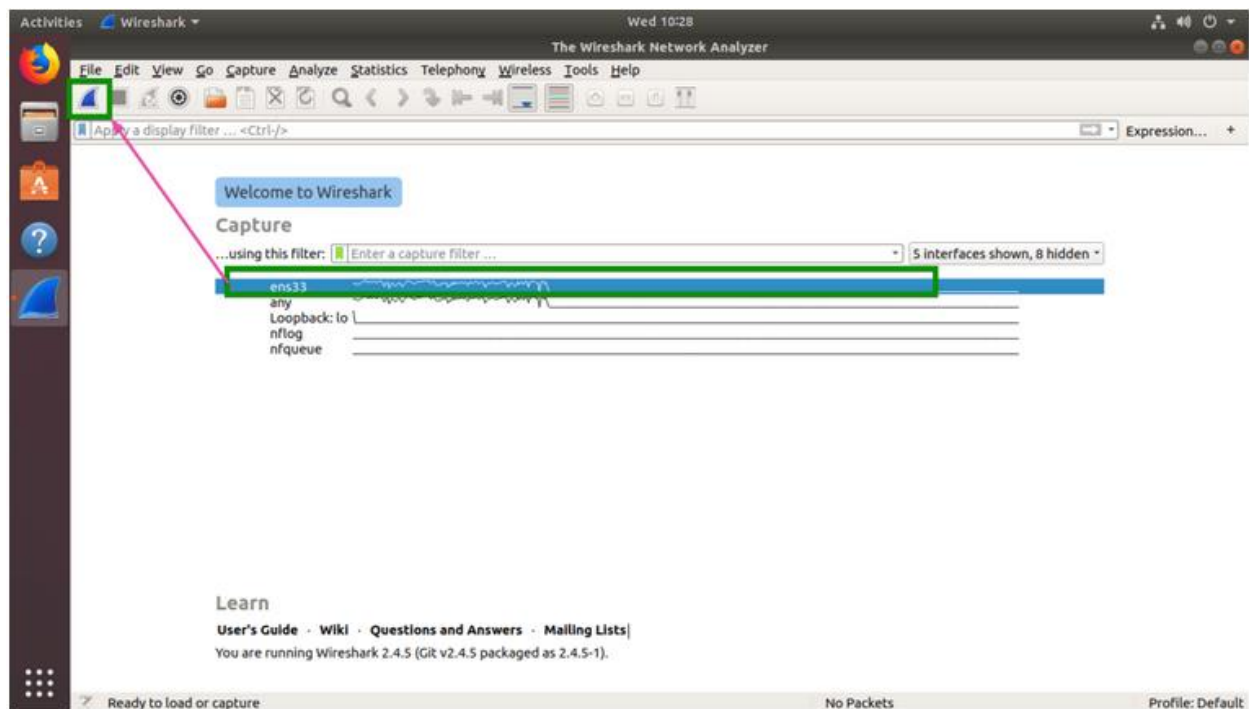
There are many types of interfaces you can monitor using Wireshark, for example, **Wired**, **Wireless**, USB and many external devices. You can choose to show specific types of interfaces in the welcome screen from the marked section of the screenshot below.



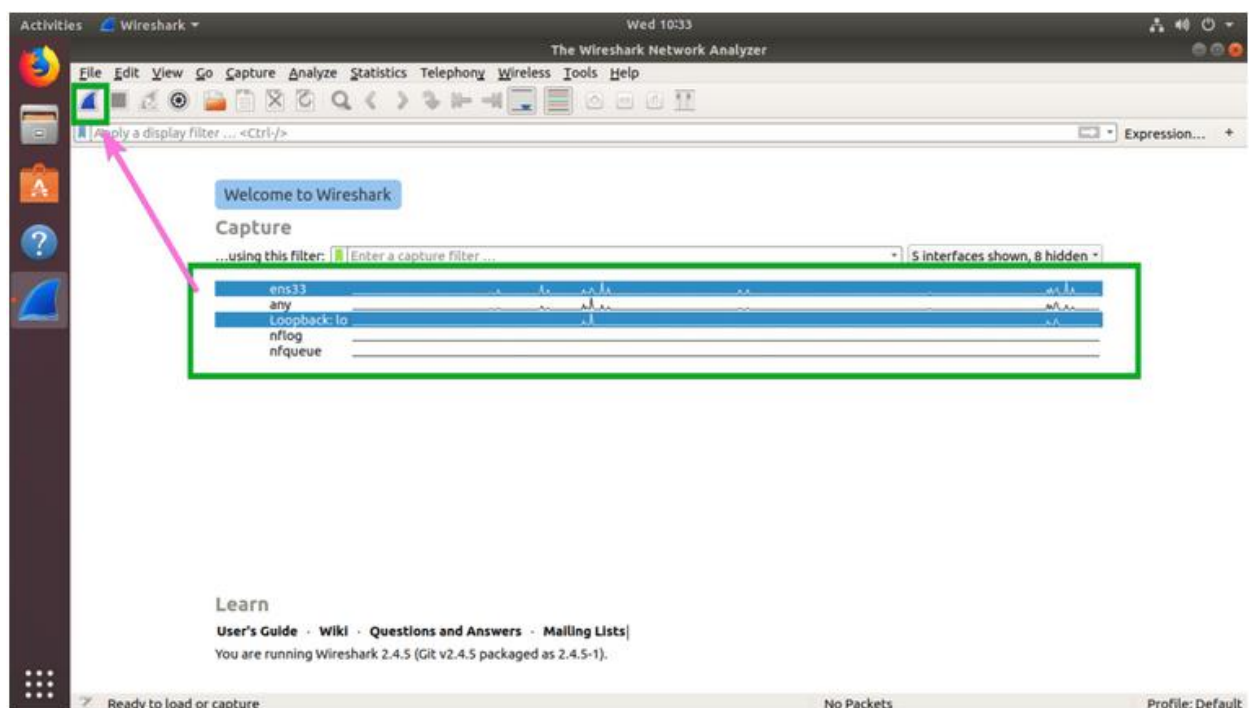
Here, I listed only the **Wired** network interfaces.



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 double click on the interface that you want to capture packets to and from to start capturing packets on that particular interface.

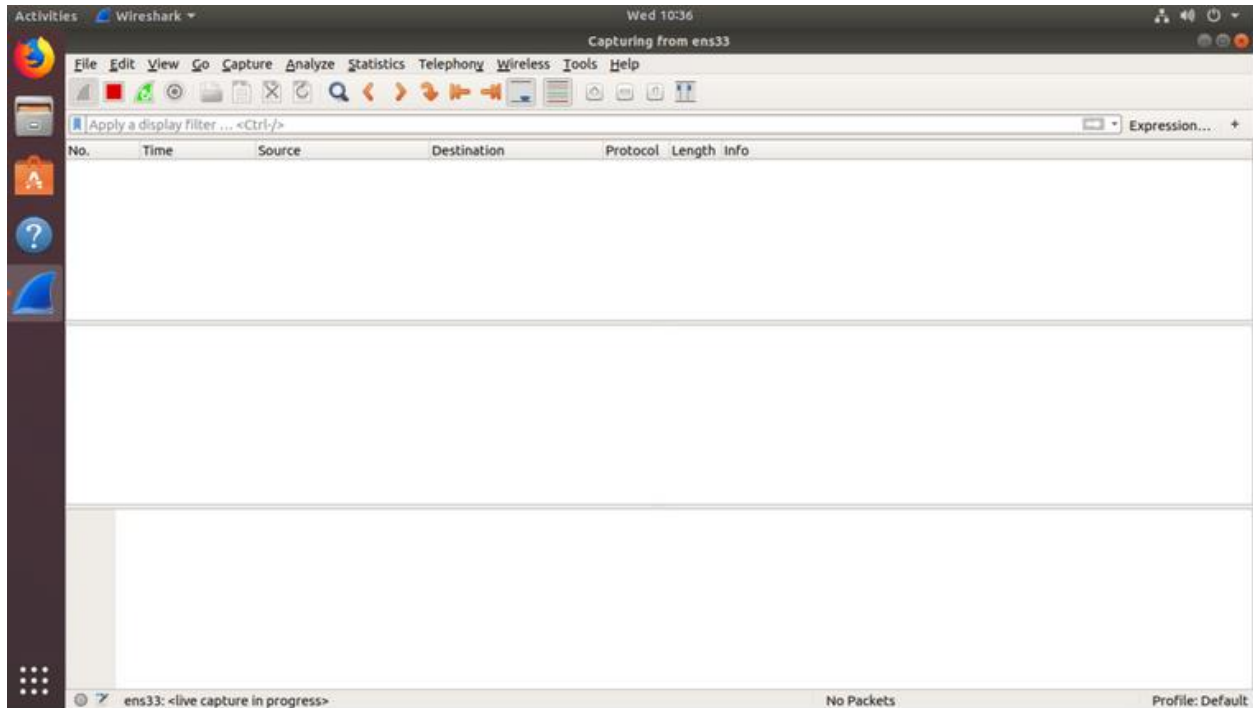


You can also capture packets to and from multiple interfaces at the same time. Just press and hold **<Ctrl>** and click on the interfaces that you want to capture packets to and from and then click on the **Start capturing packets** icon as marked in the screenshot below.

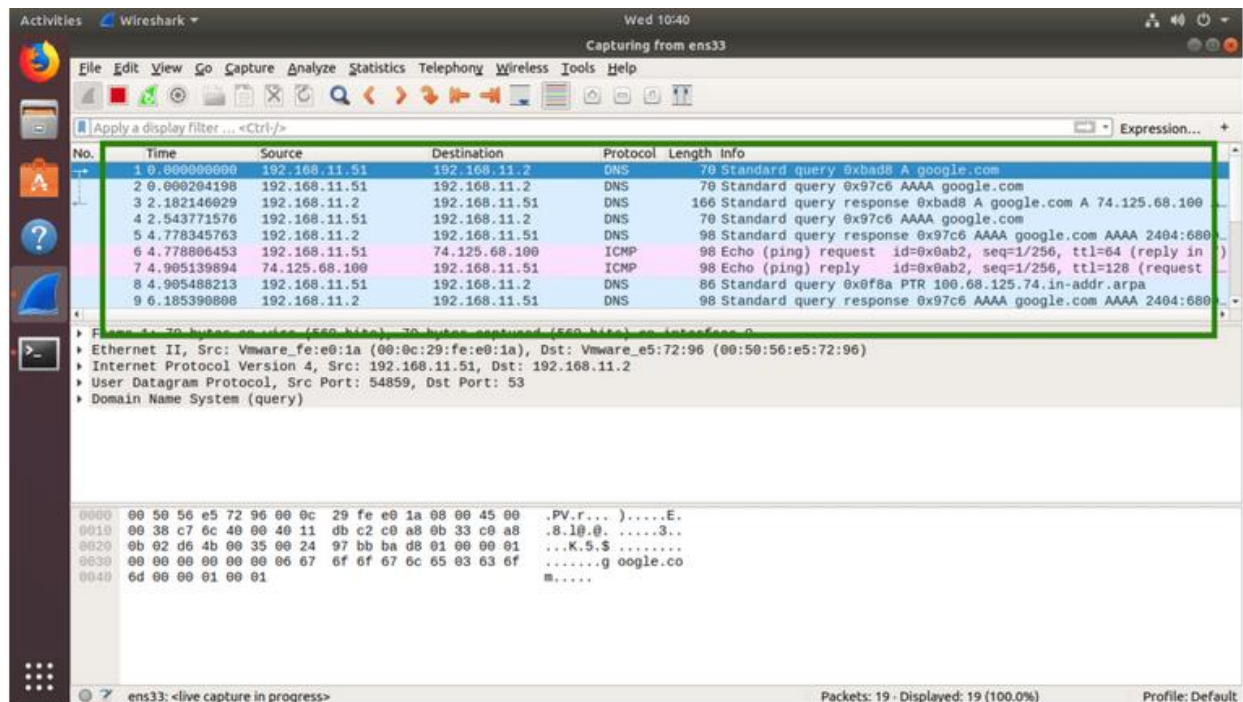


Using Wireshark on Ubuntu:

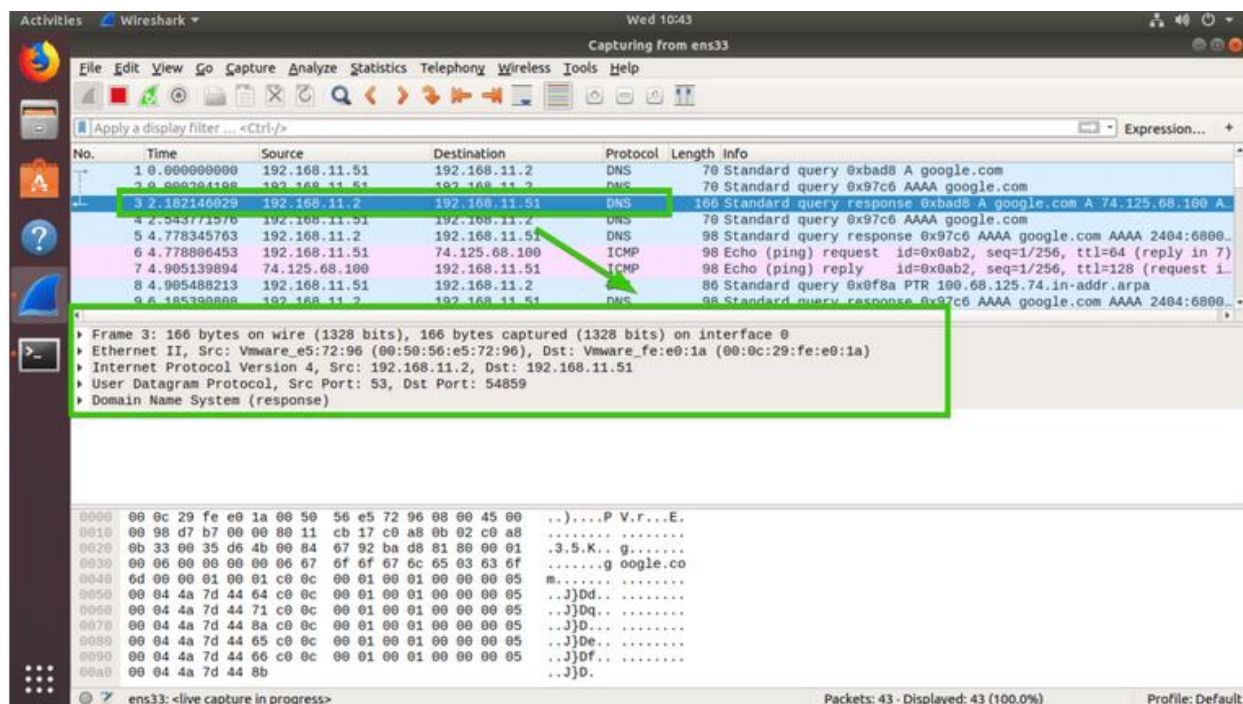
I am capturing packets on the **ens33** wired network interface as you can see in the screenshot below. Right now, I have no captured packets.



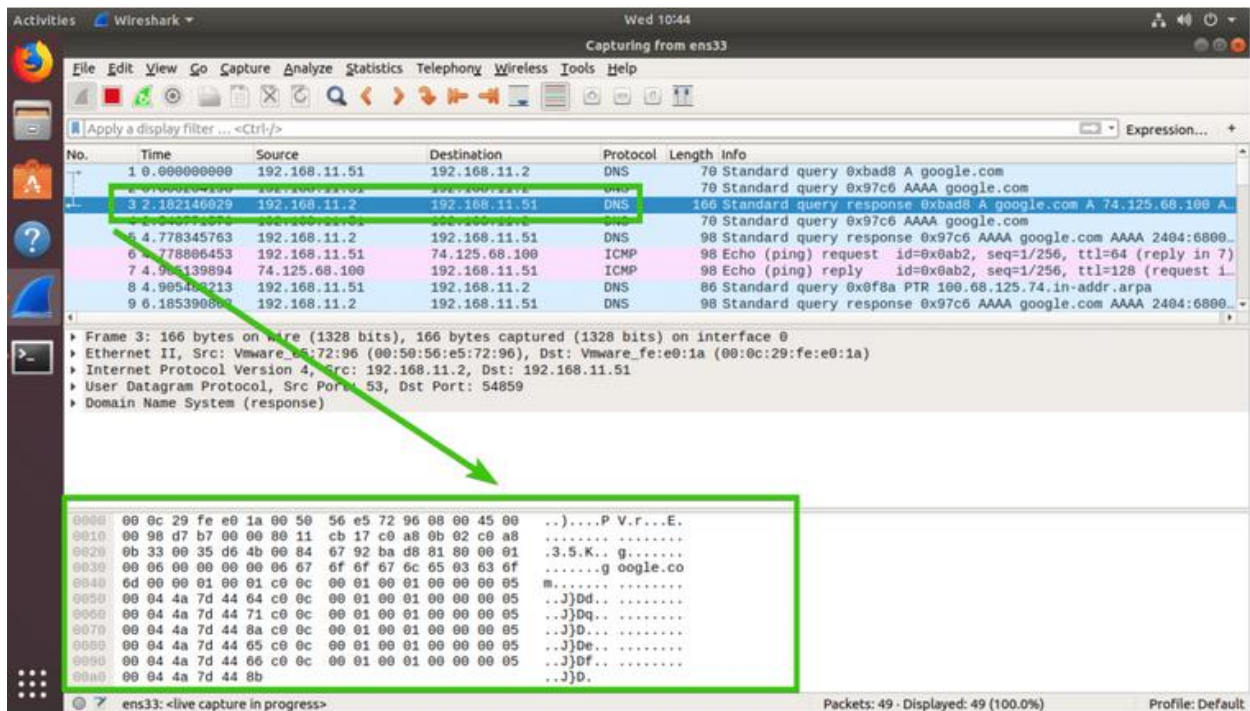
I pinged google.com from the terminal and as you can see, many packets were captured.



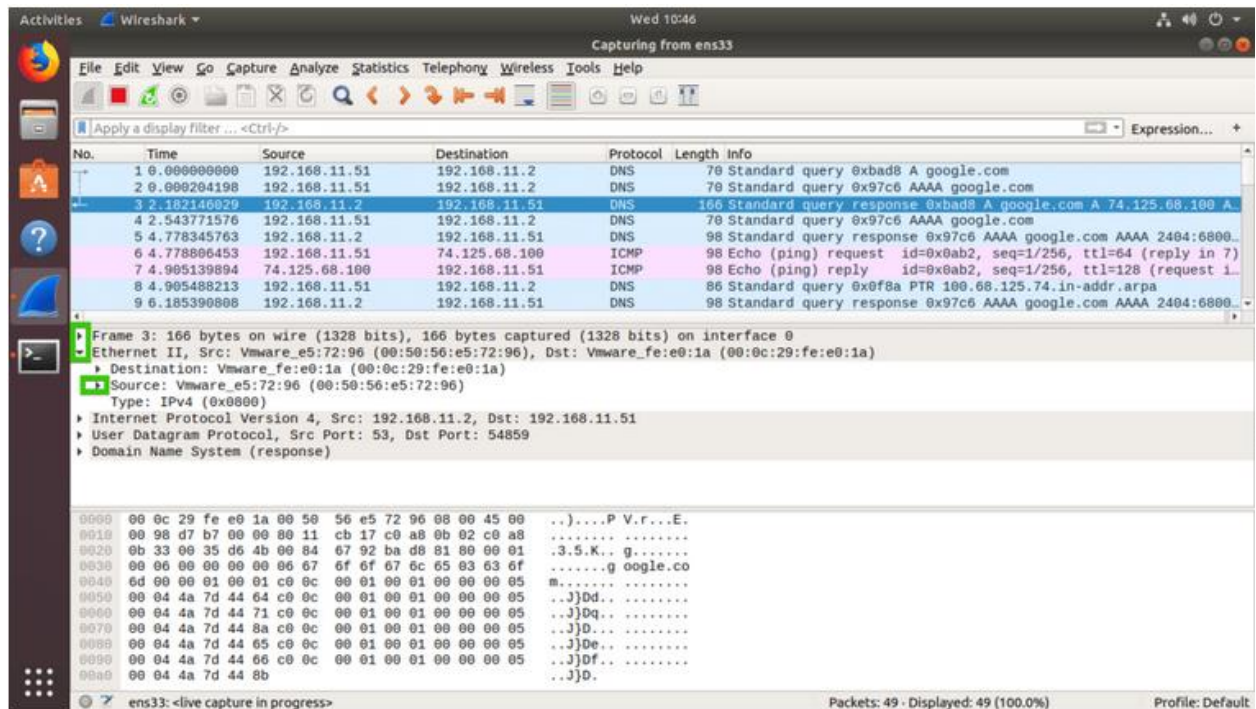
Now you can click on a packet to select it. Selecting a packet would show many information about that packet. As you can see, information about different layers of TCP/IP Protocol is listed.



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



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

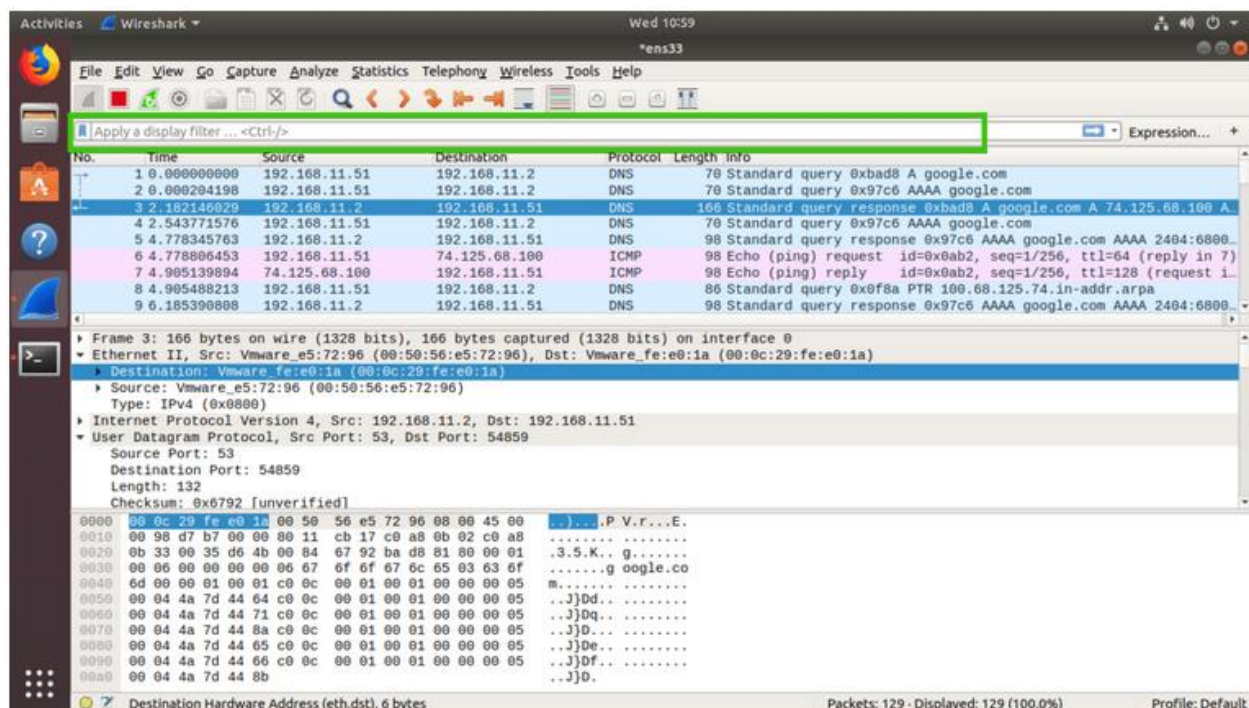


Filtering Packets Using Wireshark:

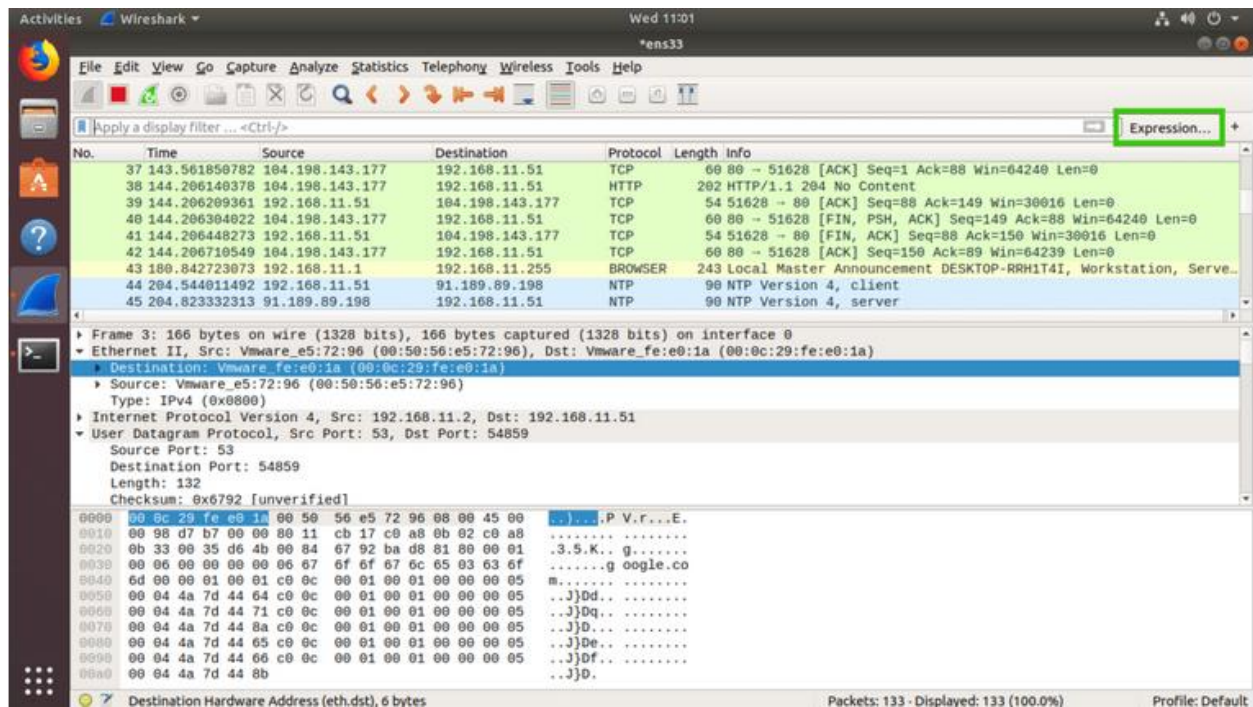
On a busy network thousands or millions of packets will be captured each second. So the list will be so long that it will be nearly impossible to scroll through the list and search for certain type of packet.

The good thing is, in Wireshark, you can filter the packets and see only the packets that you need.

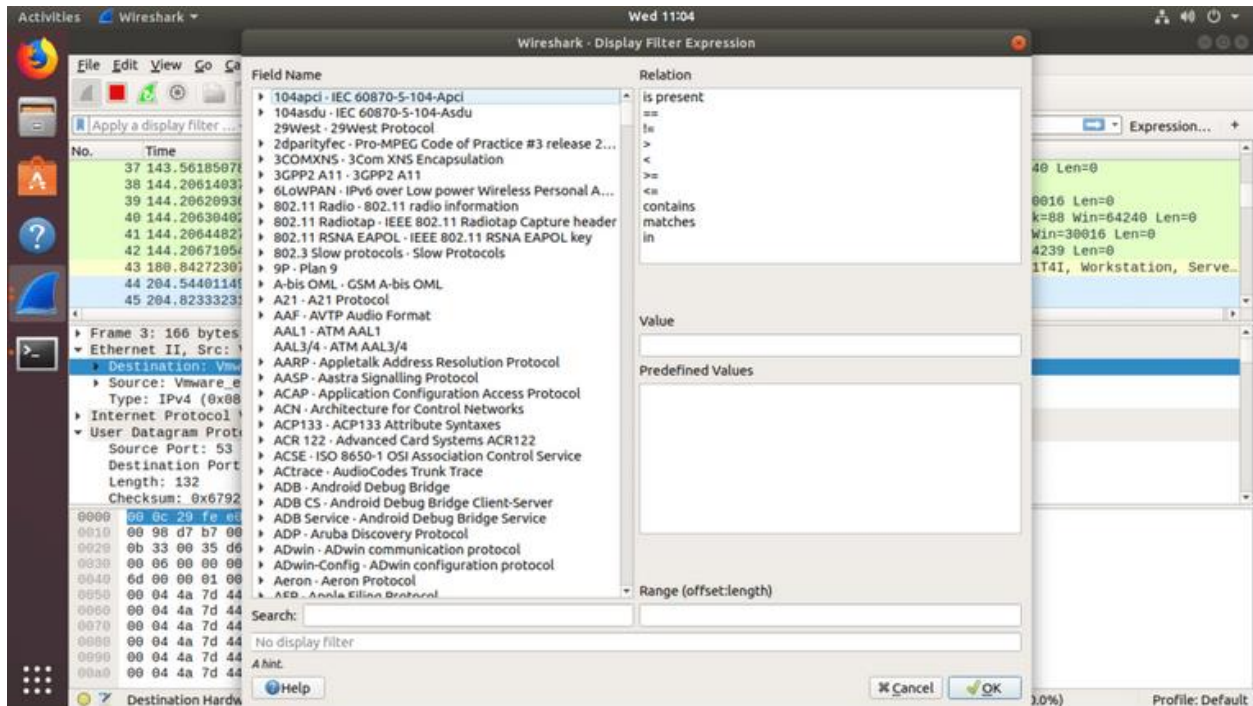
To filter packets, you can directly type in the filter expression in the textbox as marked in the screenshot below.



You can also filter packets captured by Wireshark graphically. To do that, click on the **Expression...** button as marked in the screenshot below.

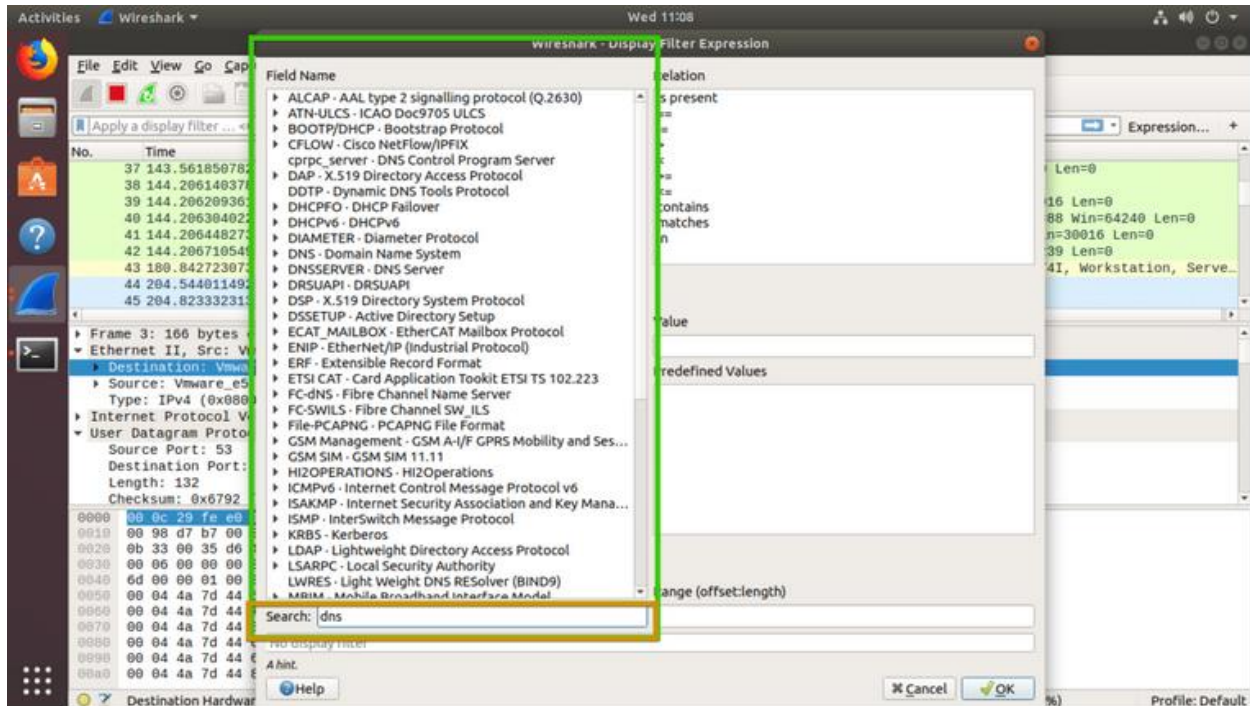


A new window should open as shown in the screenshot below. From here you can create filter expression to search packets very specifically.

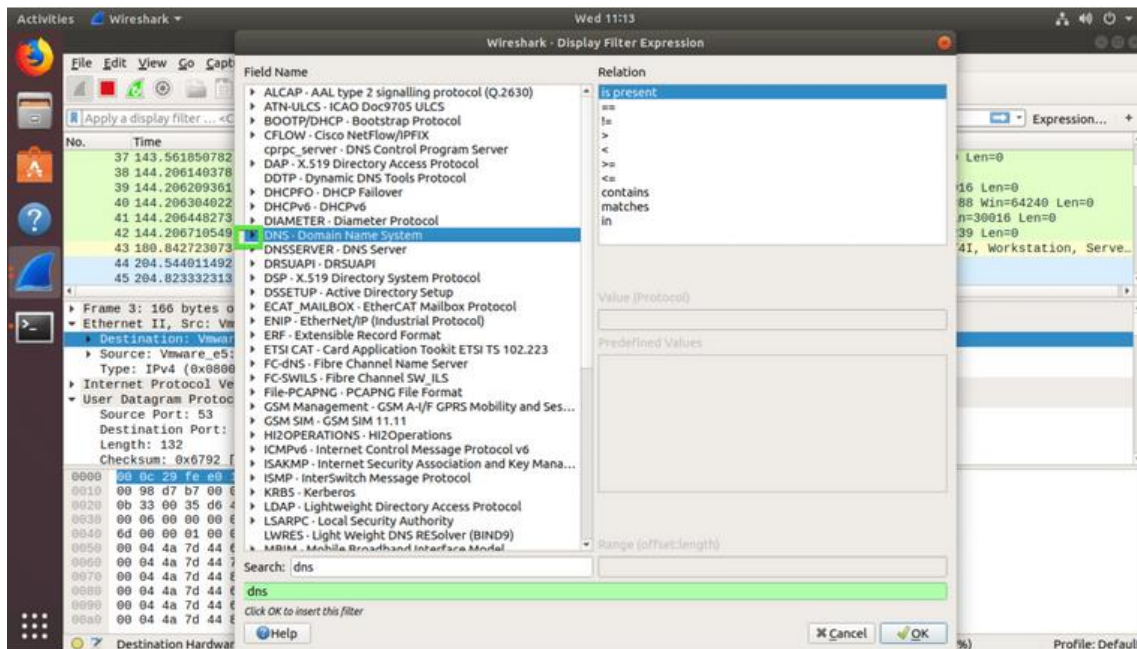


In the **Field Name** section almost all the networking protocols are listed. The list is huge. You can type in what protocol you're looking for in the

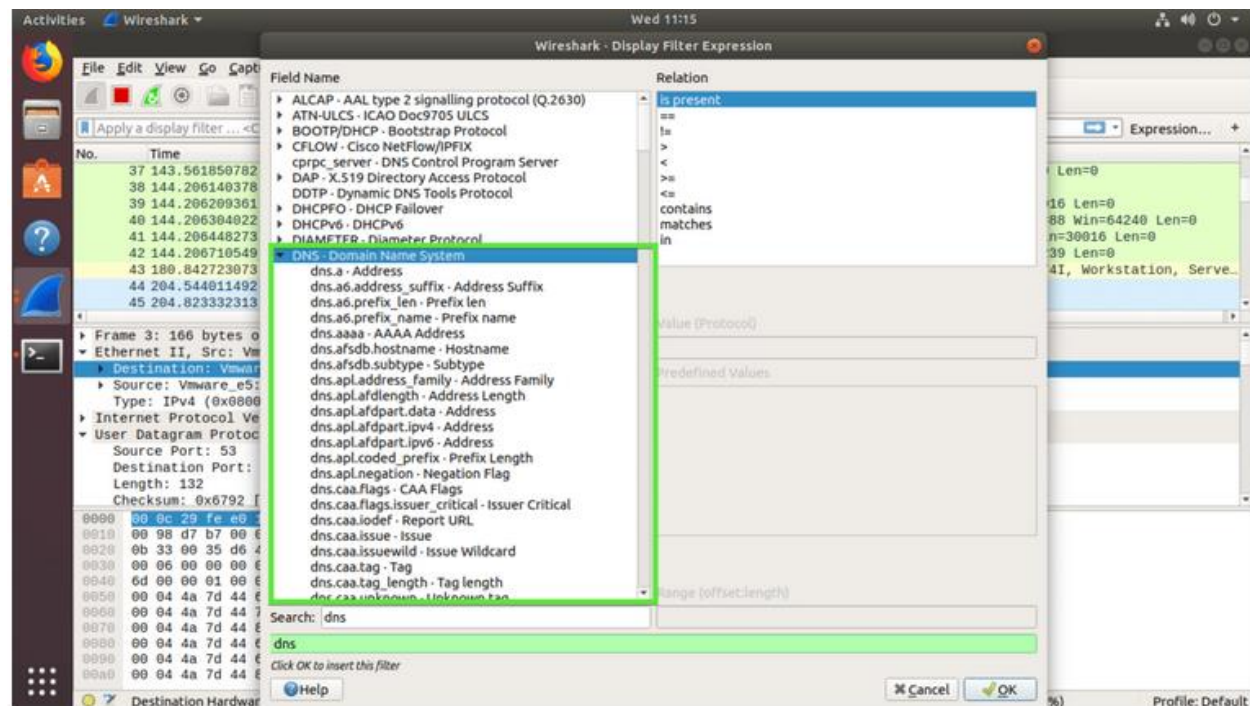
Search textbox and the **Field Name** section would show the ones that matched.



Now, 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

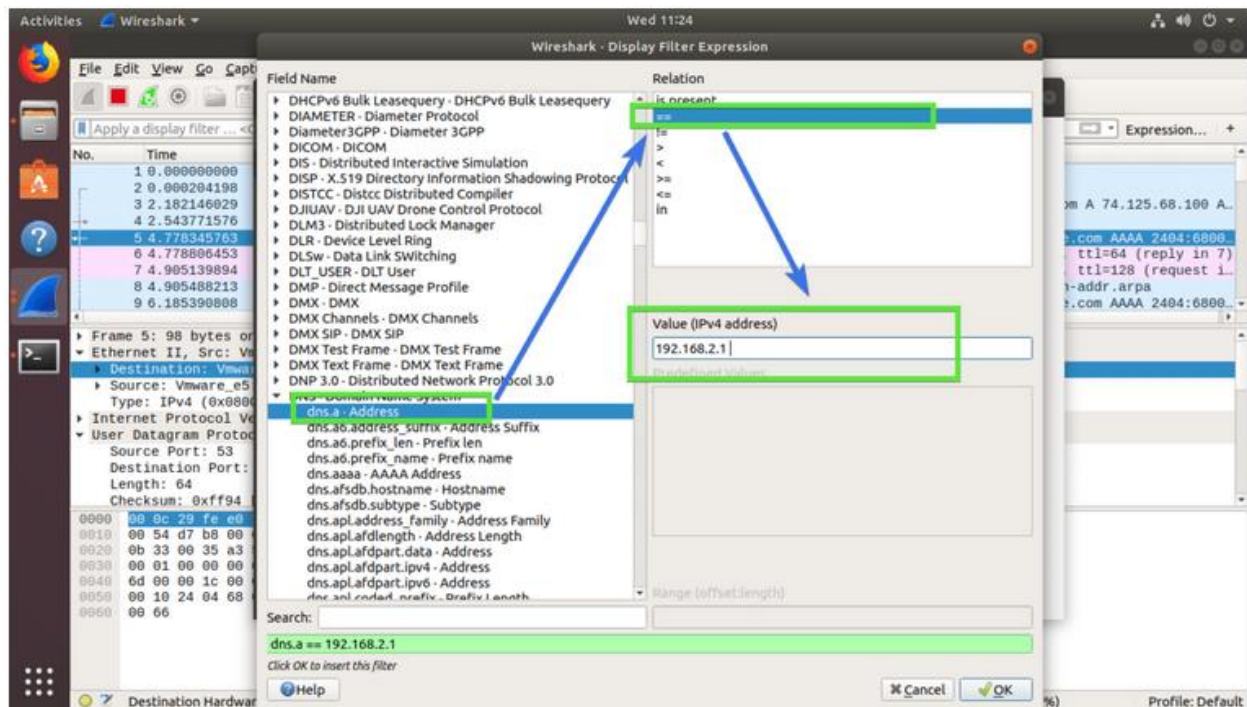


And make your selection more specific.

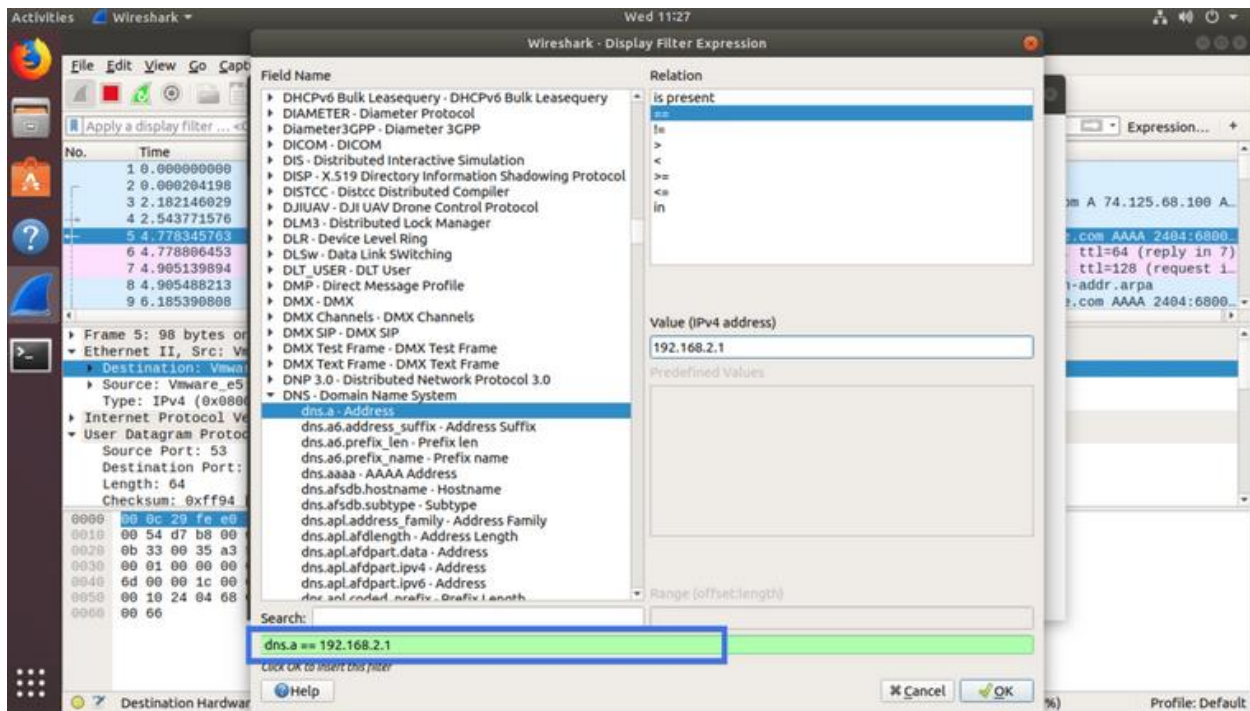


You can also use relational operators to test whether some field is equal to, not equal to, great than or less than some value. I searched

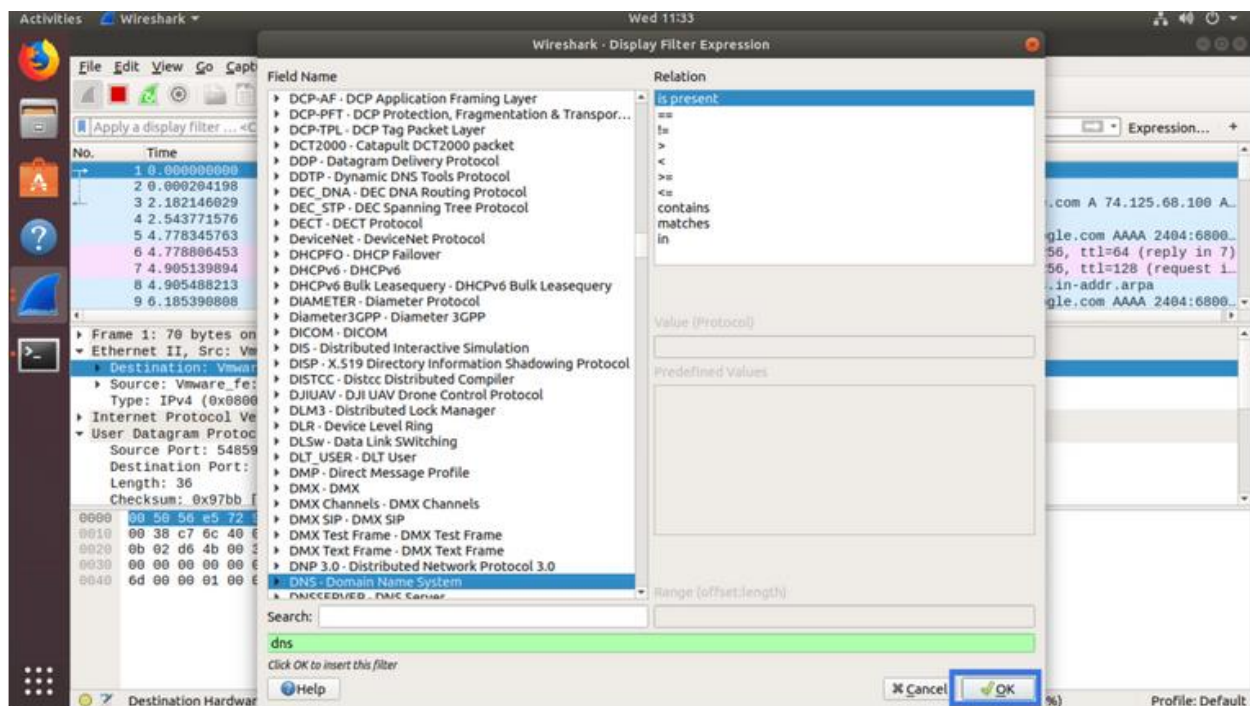
for all the **DNS IPv4** address which is equal to **192.168.2.1** as you can see in the screenshot below.



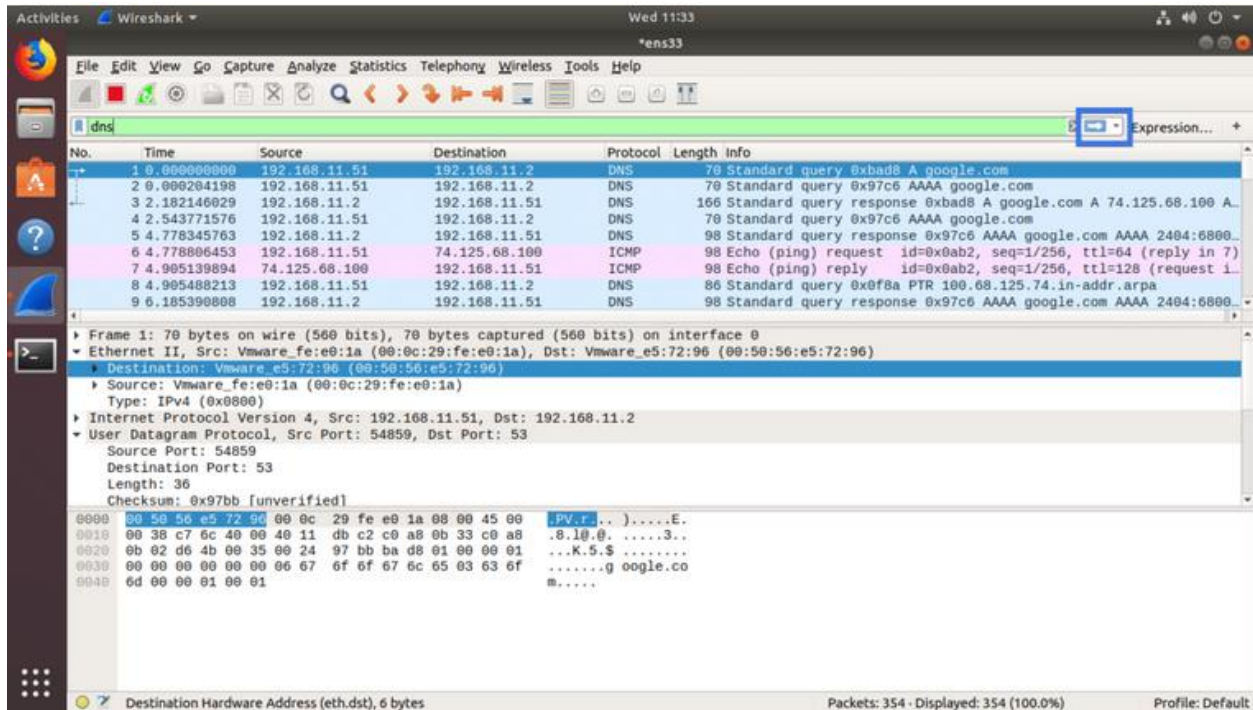
The filter expression is also shown in the marked section of the screenshot below. This is a great way to learn how to write filter expression in Wireshark.



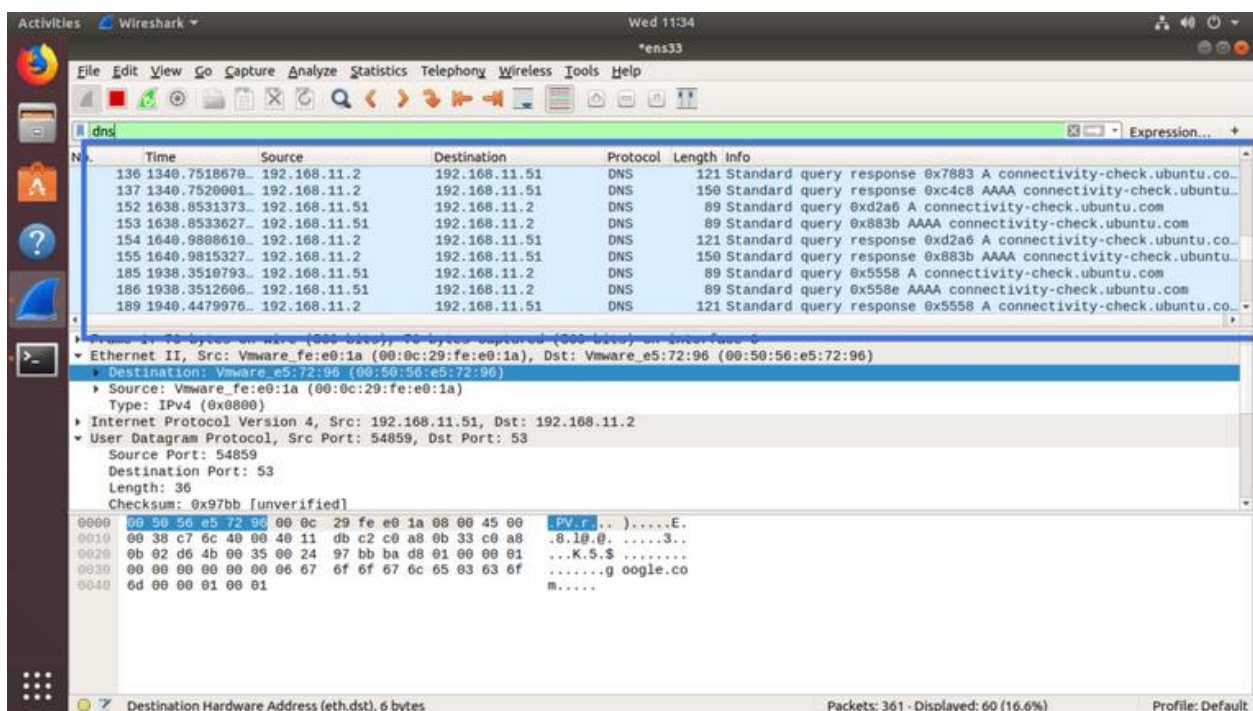
Once you're done, just click on **OK**.



Now click on the marked icon to Apply the filter.

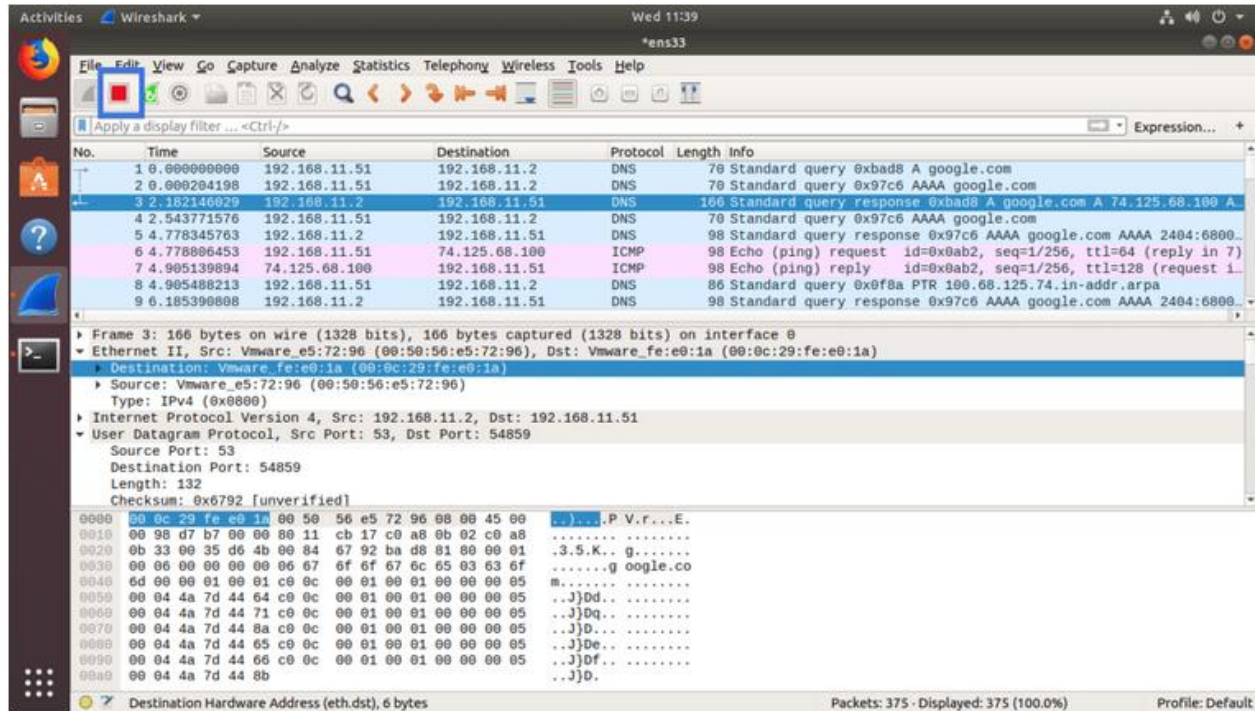


As you can see, only the DNS protocol packets are shown.



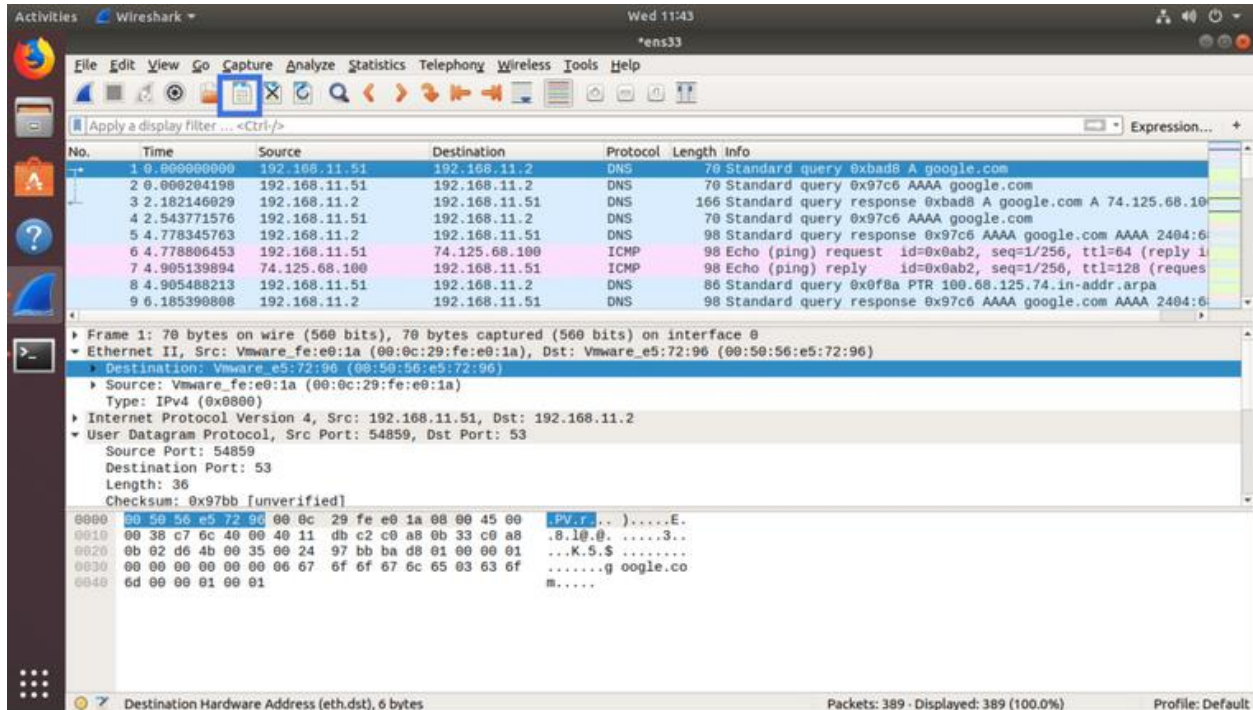
Stopping Packet Capture in Wireshark:

You can click on the red icon as marked in the screenshot below to stop capturing Wireshark packets

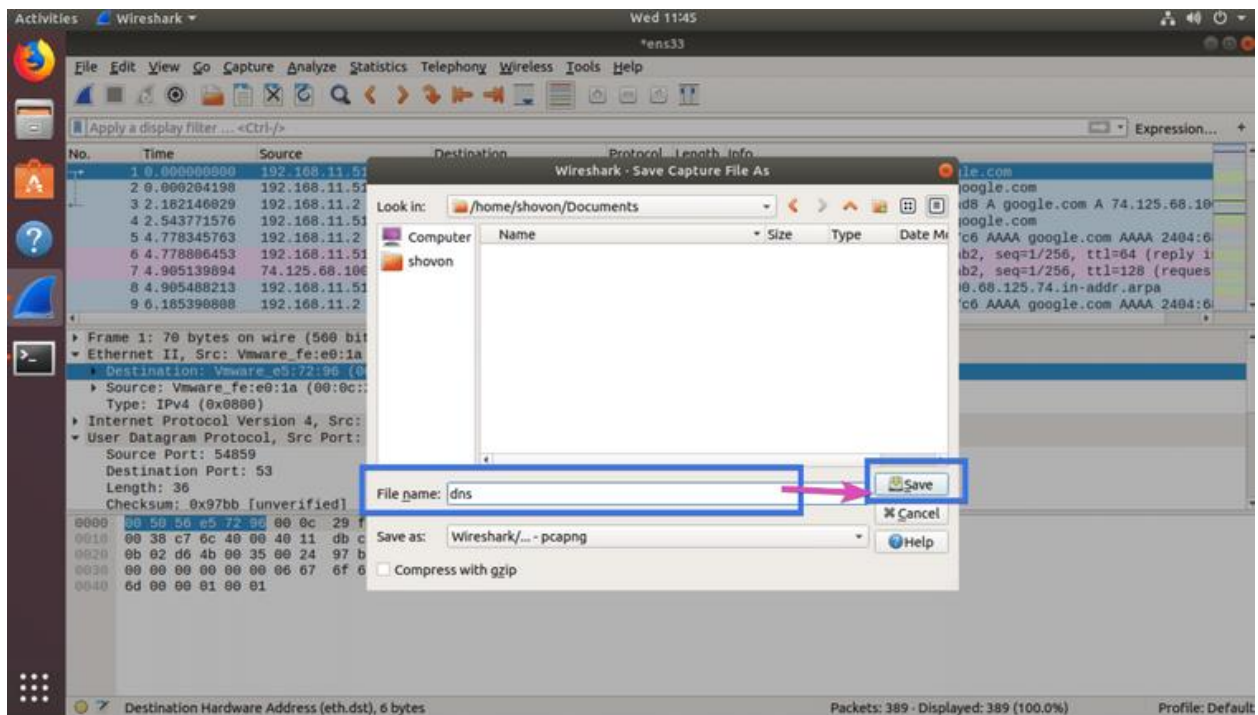


Saving Captured Packets to a File:

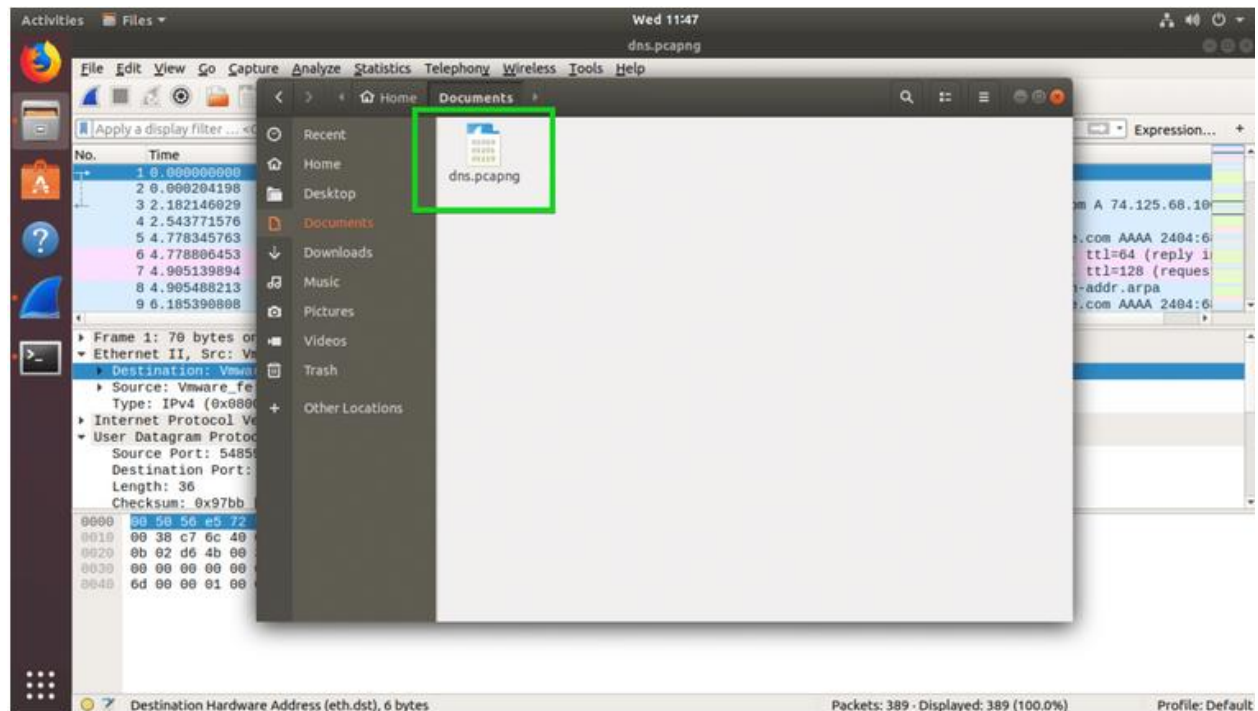
You can click on the marked icon to save captured packets to a file for future use.



Now select a destination folder, type in the file name and click on **Save**.



The file should be saved.



That's how we install and use Wireshark in Linux.

Conclusion:

Wireshark is very similar to tcpdump, but has a graphical front-end and integrated sorting and filtering options.

Wireshark lets the user put network interface controllers into promiscuous mode (if supported by the network interface controller), so they can see all the traffic visible on that interface including unicast traffic not sent to that network interface controller's MAC address. However, when capturing with a packet analyzer in promiscuous mode on a port on a network switch, not all traffic through the switch is necessarily sent to the port where the capture is done, so capturing in promiscuous mode is not necessarily sufficient to see all network traffic. Port mirroring or various network taps extend capture to any point on the network. Simple passive taps are extremely resistant to tampering.

Web debugging :

Debugging is the process of finding and fixing errors within a script. ... All modern browsers and most other environments support debugging tools – a special UI in developer tools that makes debugging much easier. It also allows to trace the code step by step to see what exactly is going on. Some debugging tools are:

- Firefox Developer Tools. ...
- Chrome Developer Tools. ...
- Web Developer. ...
- Safari Developer Tools. ...
- Internet Explorer Web Edge (Developer) Toolbar. ...
- Fiddler. ...
- Open Dragonfly. ...
- DebugBar.

In software development, the debugging process begins when a developer locates a code error in a computer program and is able to reproduce it. ... For example, an engineer might run a JTAG connection test to debug connections on an integrated circuit.

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. The fun fact is that you can use those channels for your own benefit.

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 quantities 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 netwise out of it.