

# Lab 1 : Introduction to Wireshark and Packet Capture

Submission Deadline : 28<sup>th</sup> September, 2023

Lab adapted from Computer Networking: A Top-Down Approach, Kurose Ross, 8<sup>th</sup> edition.

Student name: \_\_\_\_\_ Section : \_\_\_\_\_ Student ID: \_\_\_\_\_

One's understanding of network protocols can often be greatly deepened by “seeing protocols in action” and by “playing around with protocols” – observing the sequence of messages exchanged between two protocol entities, delving down into the details of protocol operation, and causing protocols to perform certain actions and then observing these actions and their consequences. This can be done in simulated scenarios or in a “real” network environment such as the Internet. In the Wireshark labs you'll be doing in this course, you'll be running various network applications in different scenarios using your own computer. You'll observe the network protocols in your computer “in action,” interacting and exchanging messages with protocol entities executing elsewhere in the Internet. Thus, you and your computer will be an integral part of these “live” labs. You'll observe, and you'll learn, by doing.

In this first Wireshark lab, you'll get acquainted with Wireshark (GUI Tool), and make some simple packet captures and observations. Moreover, you will also be introduced to Tshark (CLI Tool) for recorded packet observations.

The basic tool for observing the messages exchanged between executing protocol entities is called a packet sniffer. As the name suggests, a packet sniffer captures (“sniffs”) messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent/received from/by application and protocols executing on your machine.

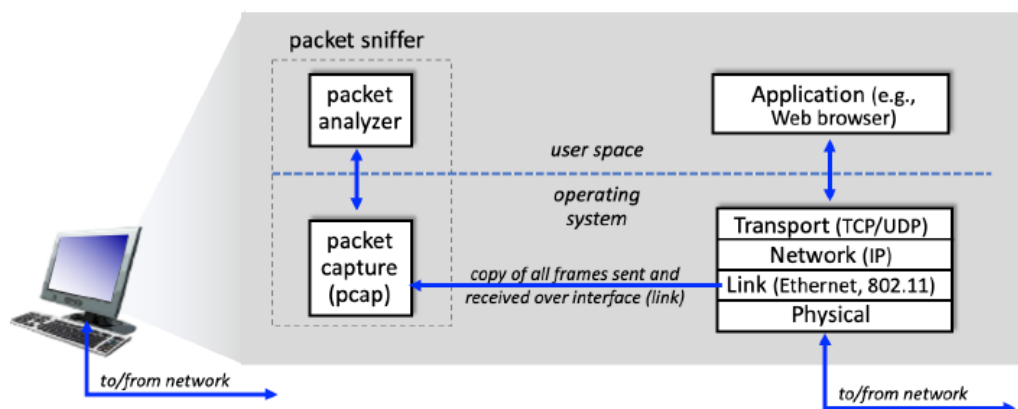


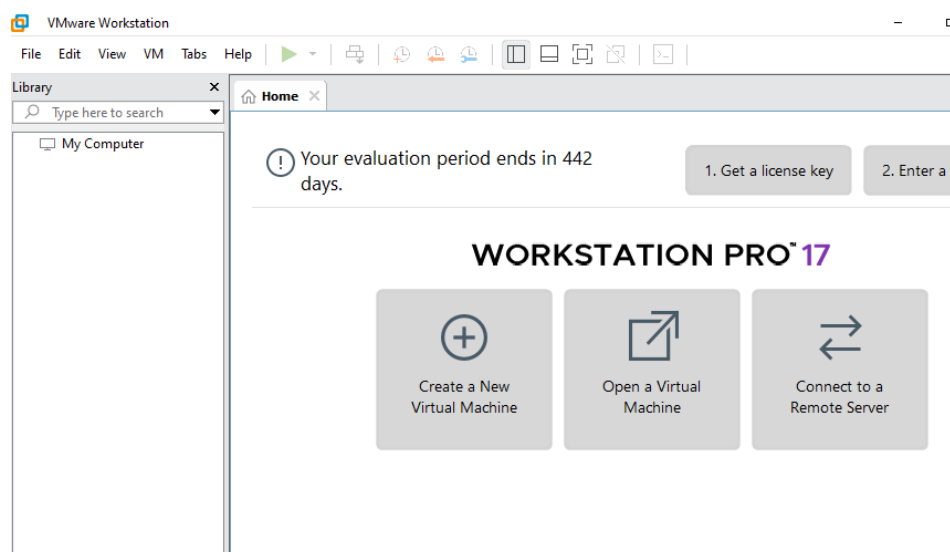
Figure 1.1 : Packet Sniffer Structure

The second component of a packet sniffer is the **packet analyzer**, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must “understand” the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in Figure 1. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string “GET,” “POST,” or “HEAD,”.

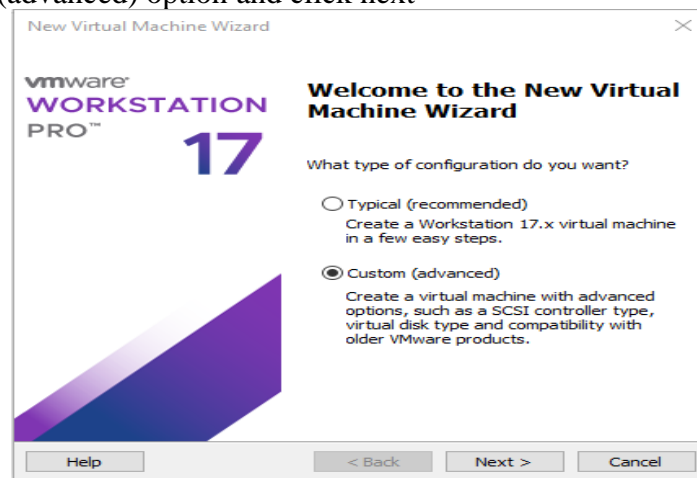
Figure 1.1 shows the structure of a packet sniffer. At the right of figure 1.1 are the protocols (in this case, Internet protocols) and applications (such as a web browser or email client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in Figure 1.1 is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer over a given interface (link layer, such as Ethernet or WiFi). Recall from the discussion of Chapter 1 slides that messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable or an 802.11 WiFi radio. Capturing all link-layer frames thus gives you all messages sent/received across the monitored link from/by all protocols and applications executing in your computer.

### Task 1: Setting up a virtual machine

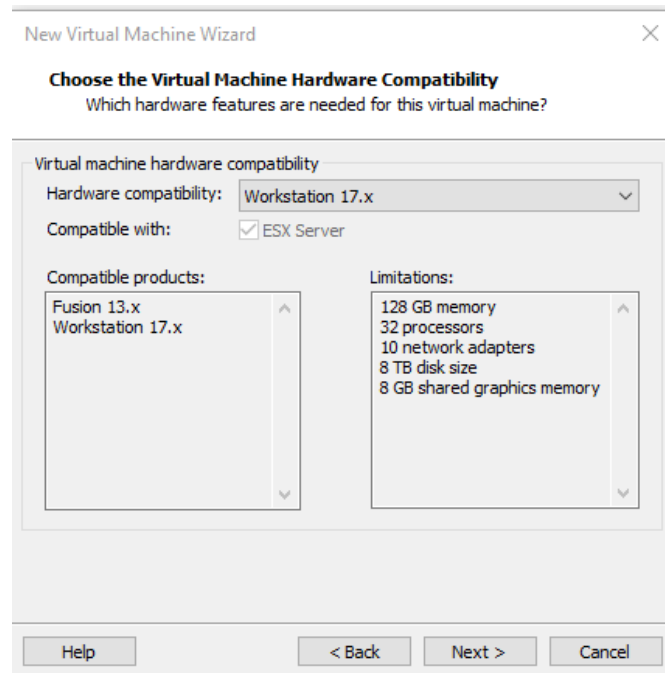
- We will use VMware workstation software to load Ubuntu virtual machine for our lab.
- Download the Ubuntu image from this link [Ubuntu 20.04.4 \(64bit\).vmdk](#)
- Open VMware workstation software from your machine and click on “Create a new Virtual Machine”.



- d. Chose Custom(advanced) option and click next



- e. Follow below screenshot steps and click next on each window with appropriate selections shown.



New Virtual Machine Wizard

**Guest Operating System Installation**

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

☐ Installer disc:

No drives available

☐ Installer disc image file (iso):

Browse...

☒ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

Help < Back Next > Cancel

New Virtual Machine Wizard

**Select a Guest Operating System**

Which operating system will be installed on this virtual machine?

Guest operating system

☐ Microsoft Windows

☒ Linux

☐ VMware ESX

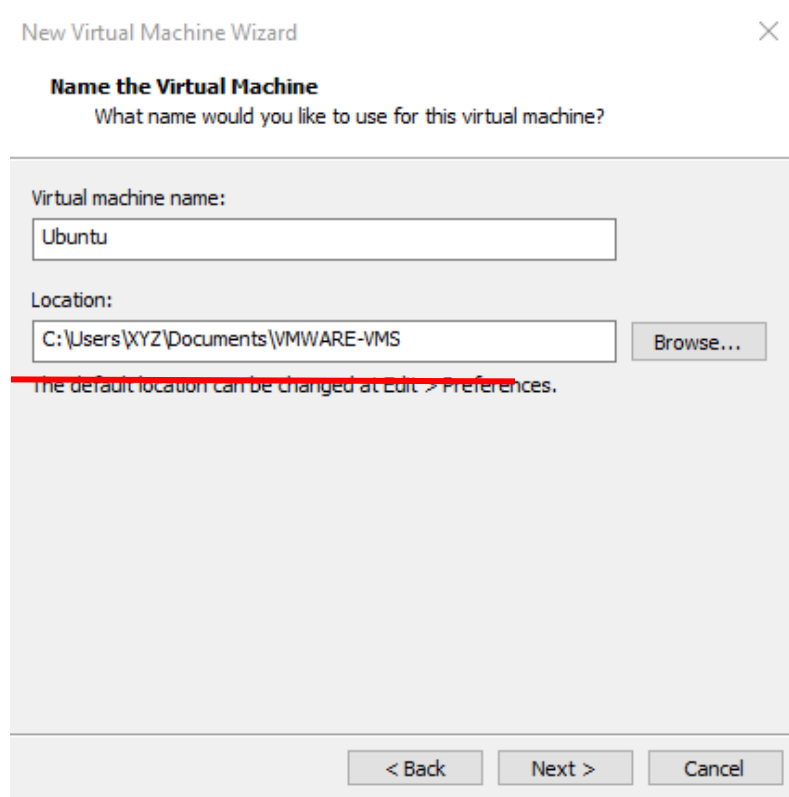
☐ Other

Version

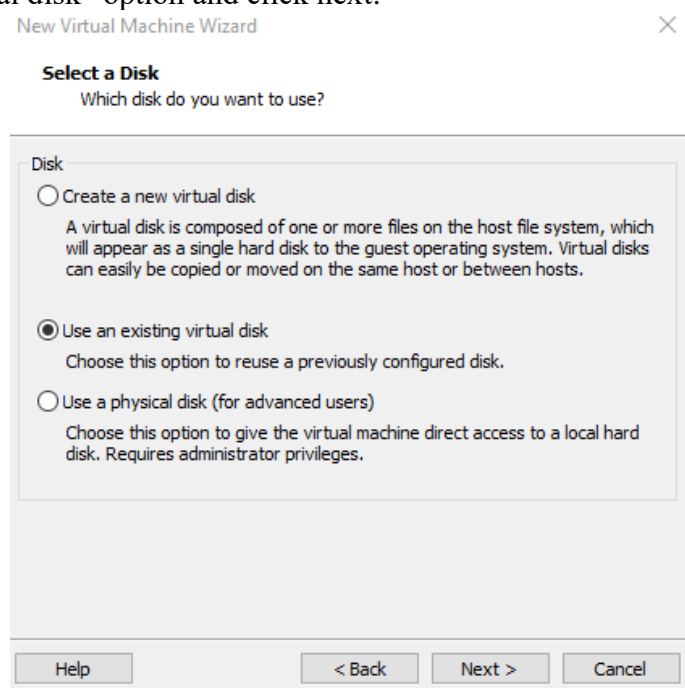
Ubuntu

Help < Back Next > Cancel

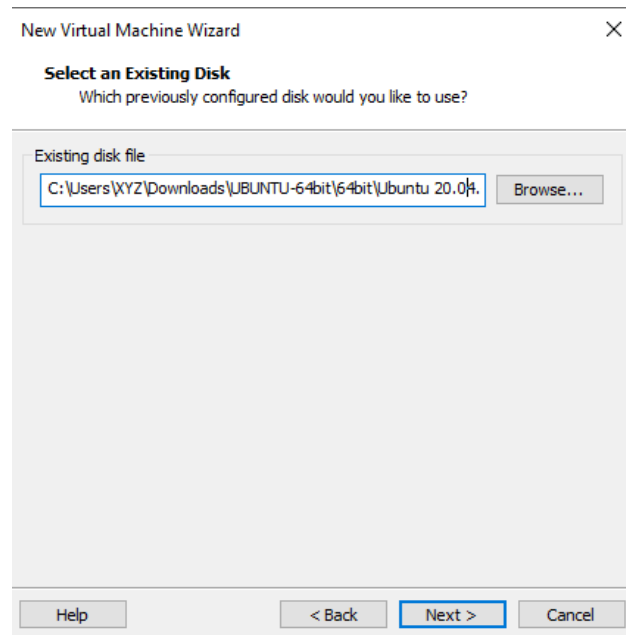
- f. In the next step, choose a location where you want to install the virtual machine and click next.



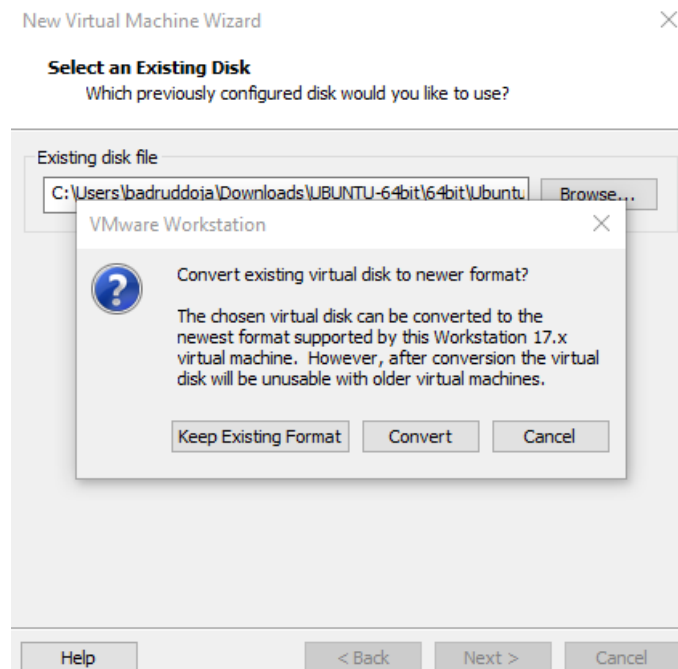
- g. Click next on all pages until you reach the below window where you will choose “Use an existing virtual disk” option and click next.



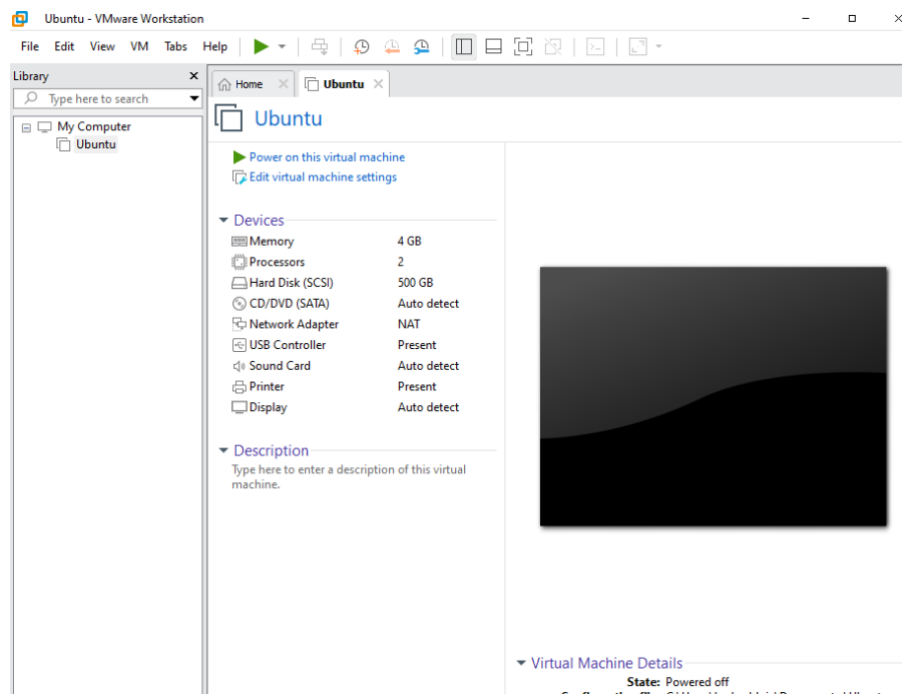
- h. Now browse and show the path where you downloaded the virtual machine vmdk file from link given earlier and click next.



- i. Chose “Keep Existing Format” in the next window

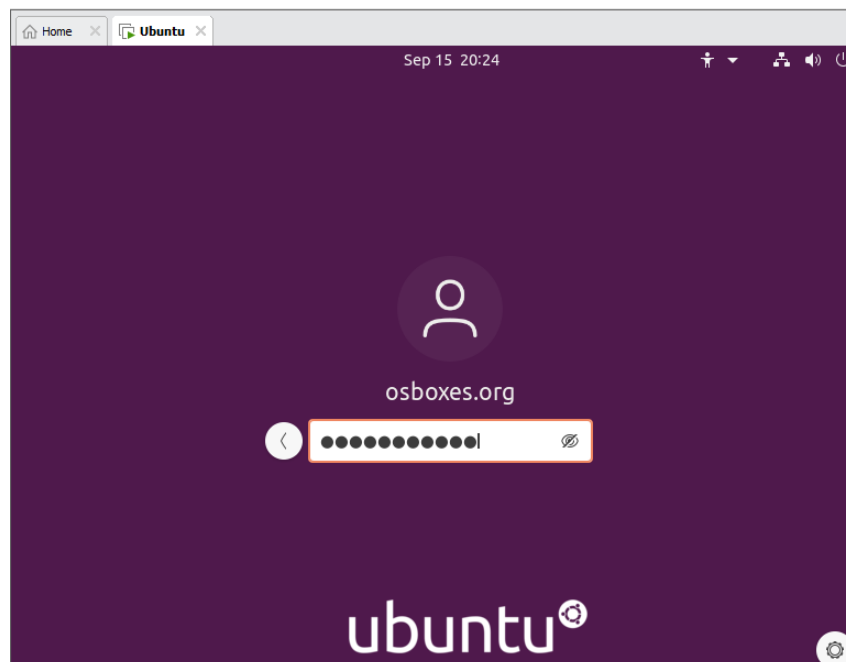


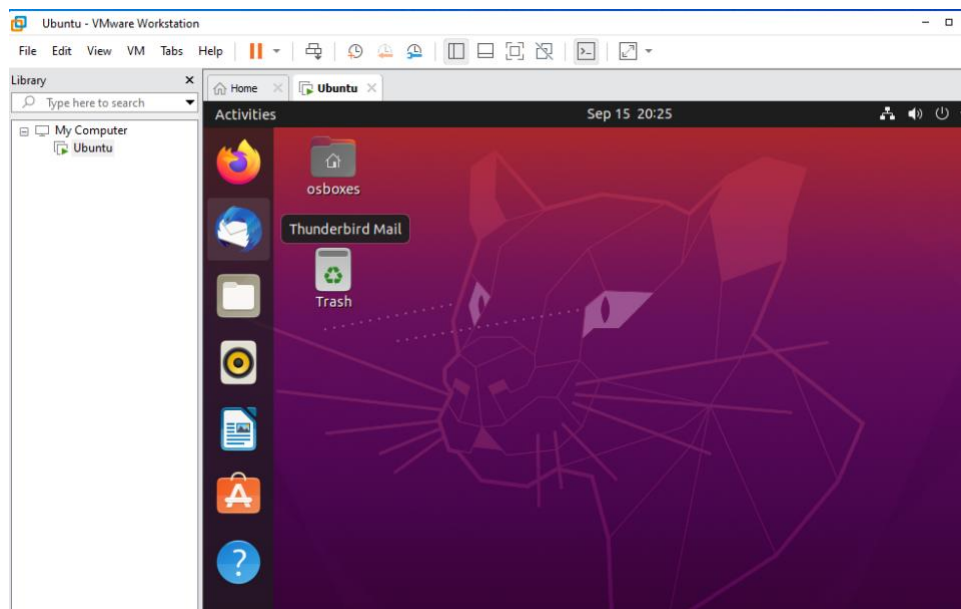
- j. Click finish on the next window and you will see your VM installed on the workstation software as shown below.



- k. Click on “Power on this virtual machine” to start the virtual machine and login using username: osboxes and password: osboxes.org.

**Note (For Sata 0:1 drive warning):** Chose “no” for the warning message where the window shows that there is no sata 0:1 drive.

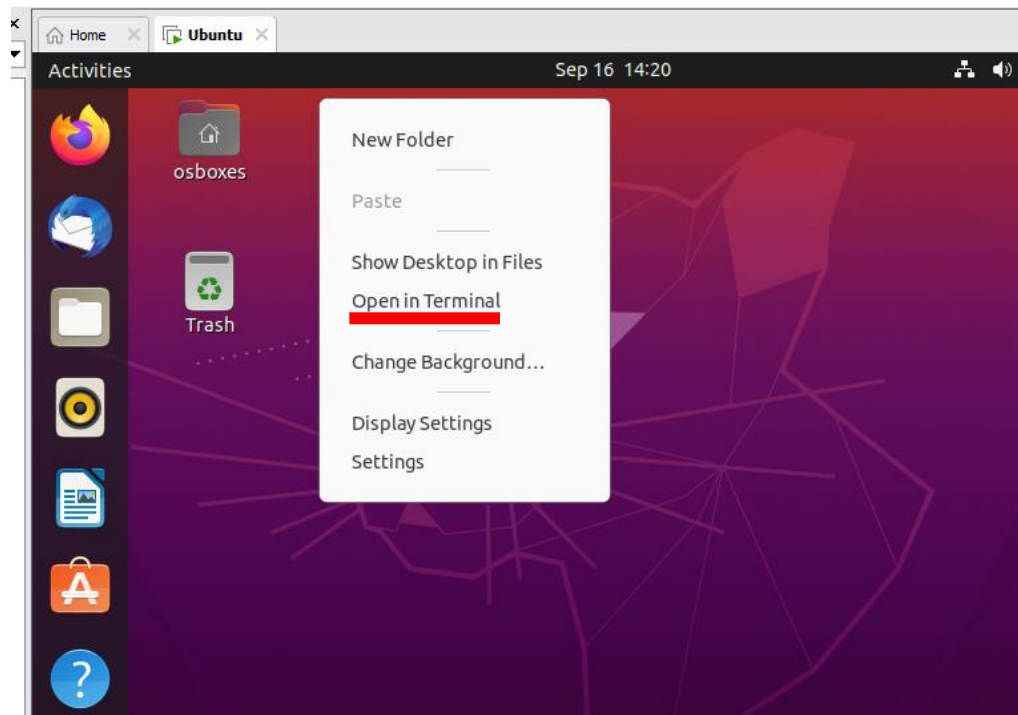




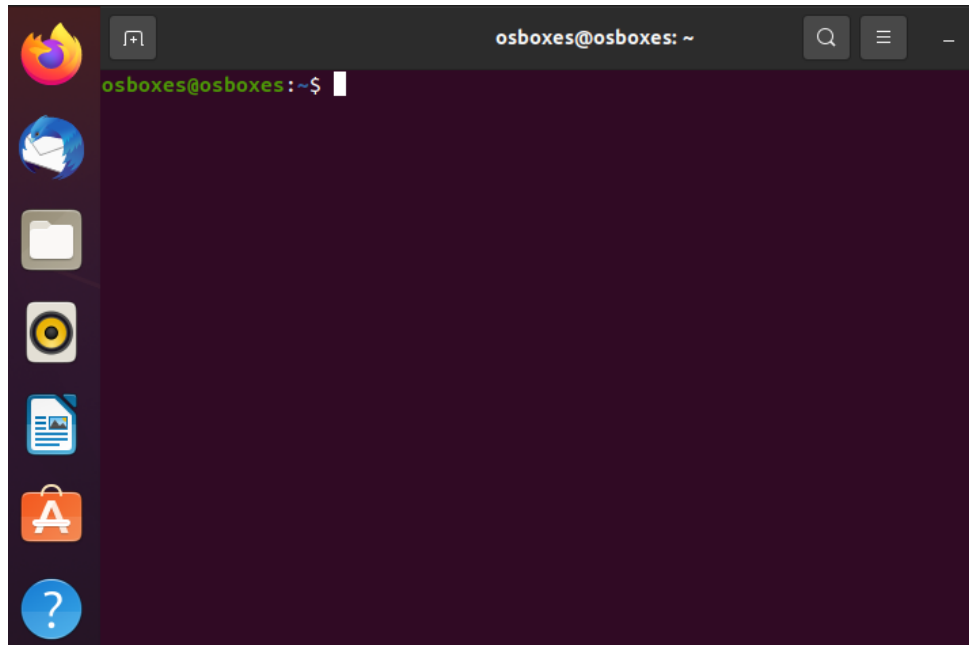
1. Congratulations! Your virtual machine is ready.

## Task 2: Install Wireshark

- a. Open terminal in ubuntu machine as shown below.







- b. Type command “`sudo add-apt-repository ppa:wireshark-dev/stable`” and press enter twice to add wireshark repository.

```
osboxes@osboxes: ~$ sudo add-apt-repository ppa:wireshark-dev/stable
[sudo] password for osboxes:
Latest stable Wireshark releases back-ported from Debian package versions.

Back-porting script is available at https://github.com/rbalint/pkg-wireshark-ubuntu-ppa

From Ubuntu 16.04 you also need to enable "universe" repository, see:
http://askubuntu.com/questions/148638/how-do-i-enable-the-universe-repository

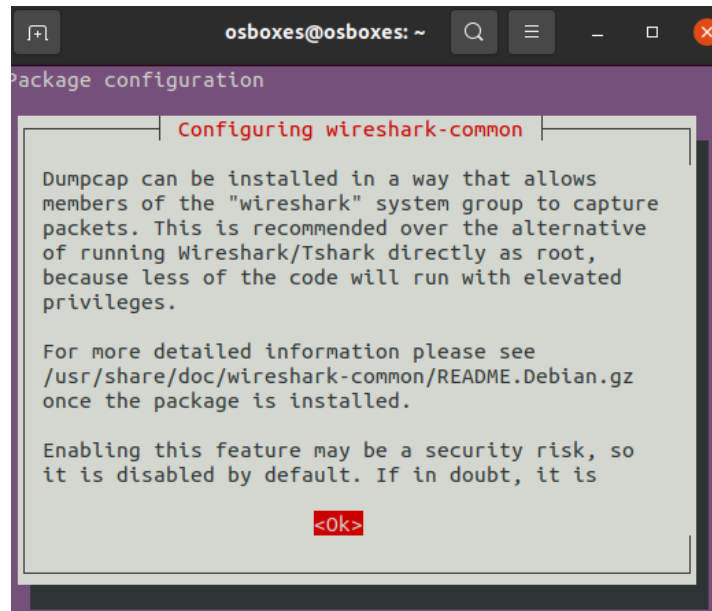
The packaging repository for Debian and Ubuntu is at: https://salsa.debian.org/debian/wireshark
More info: https://launchpad.net/~wireshark-dev/+archive/ubuntu/stable
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease [24.4 kB]
```

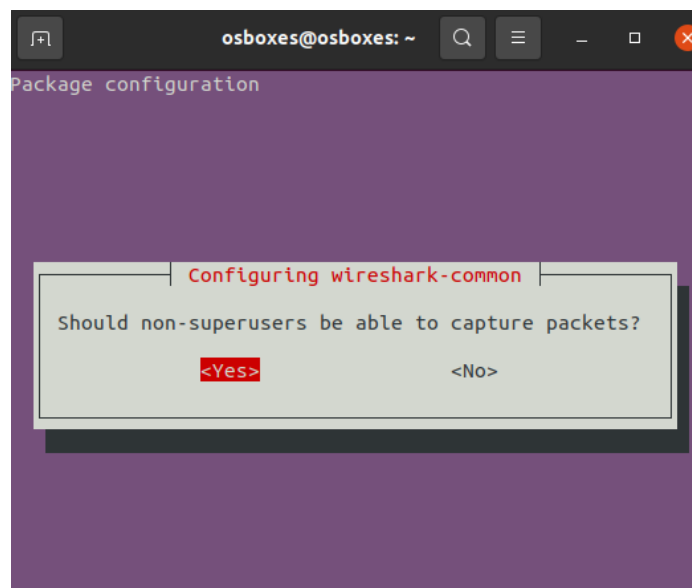
- c. Next, type command “`sudo apt install wireshark`” on the terminal and press enter.

```
osboxes@osboxes:~$ sudo apt install wireshark
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libdouble-conversion3 liblua5.2-0
  libminizip1 libpcr2-16-0 libqt5core5a libqt5dbus5
  libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins
  libqt5multimediastools5 libqt5multimediawidgets5
```

- d. Press “Ok” button on the next screen.



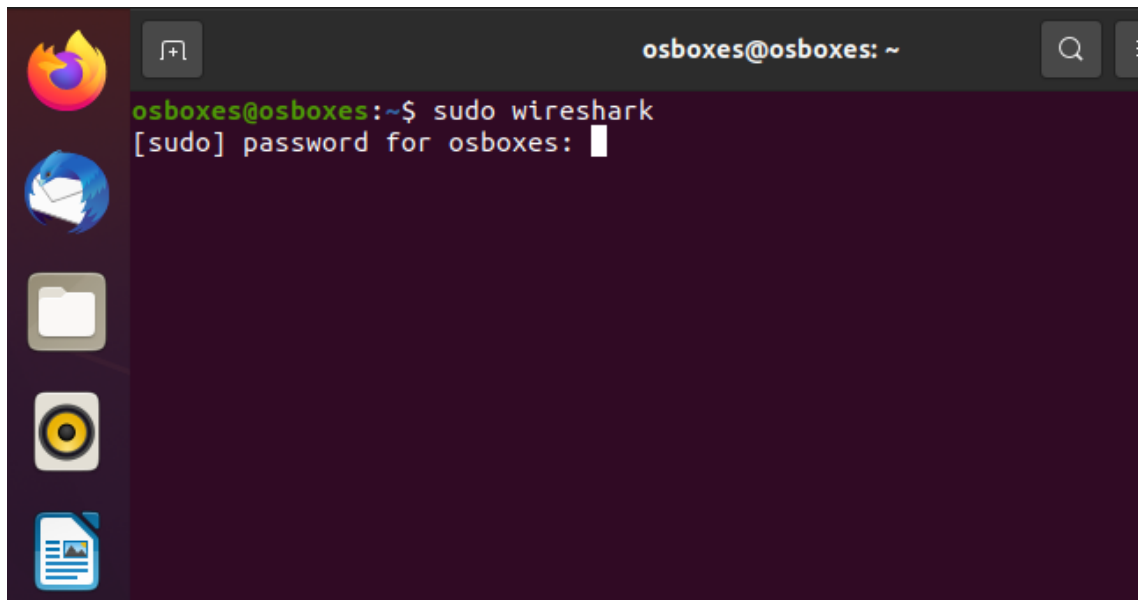
e. Press “Yes” button in the next screen.



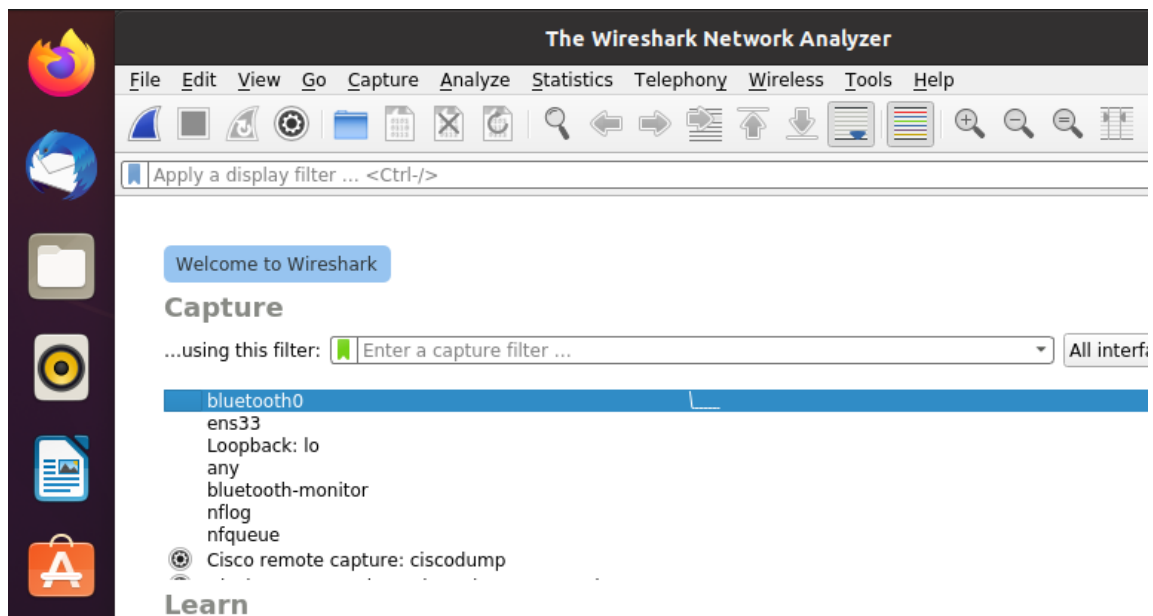
f. Congratulations! You have installed wireshark in Ubuntu

**Task 3: Test Wireshark tool**

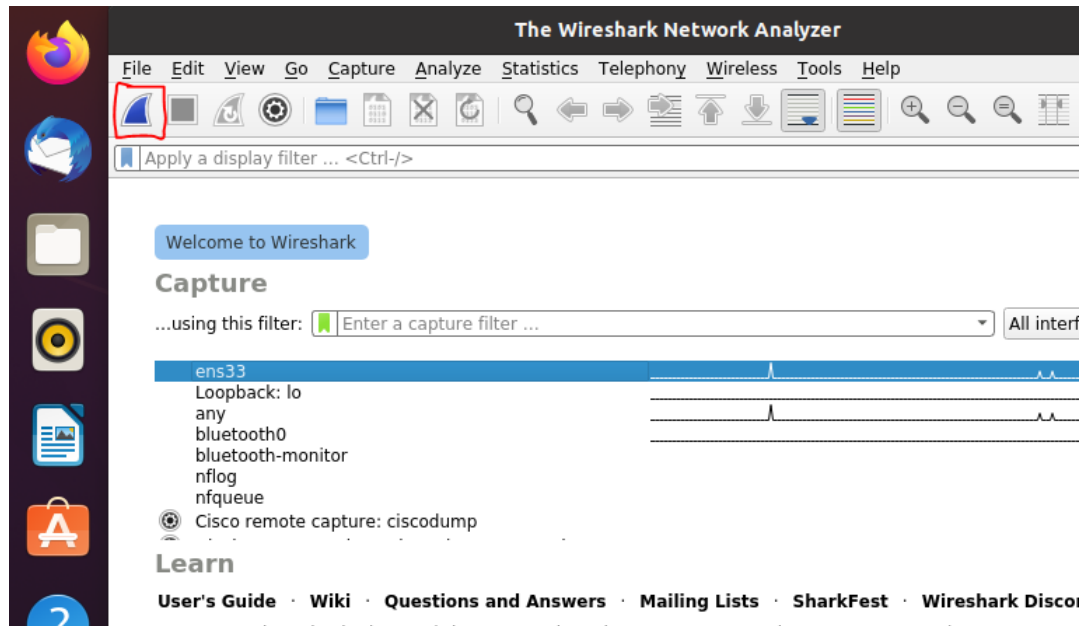
- a. Open terminal and type “sudo wireshark” and press enter, type in the password for osboxes user.



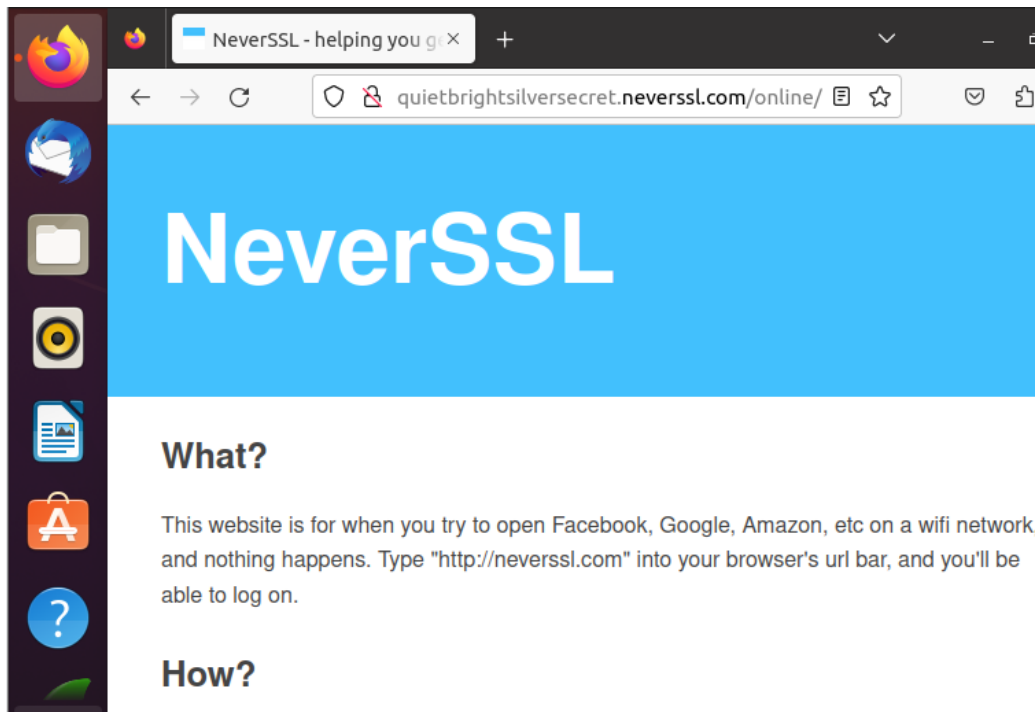
- b. You will see a window similar to below one.



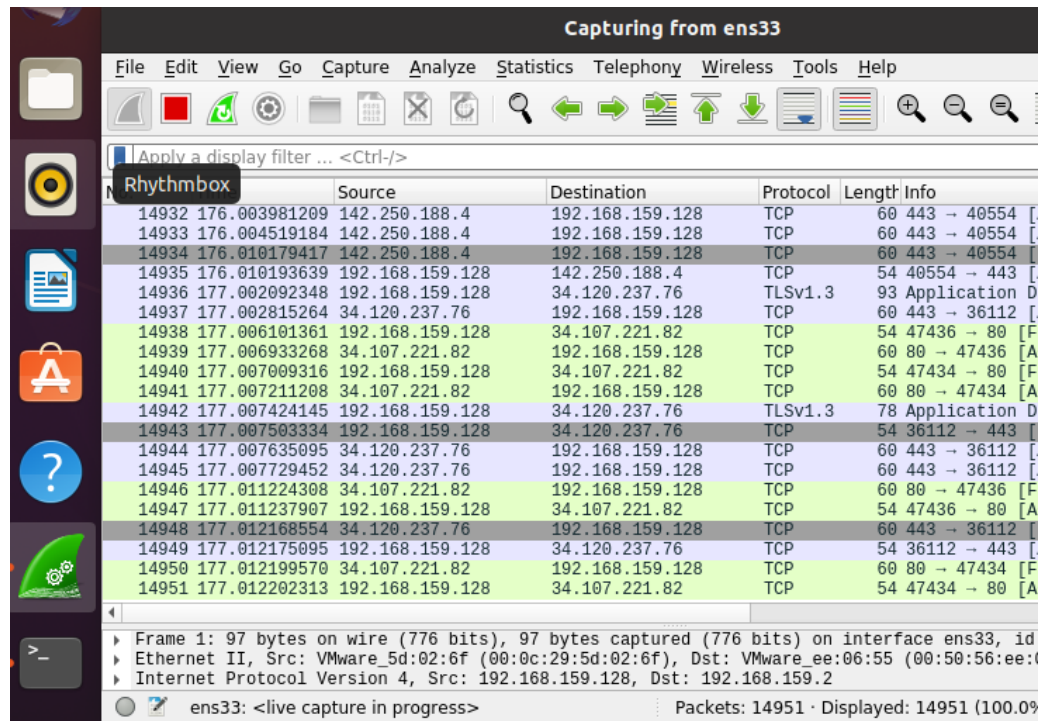
- c. Select the interface that looks similar to ens33 and click on the blue shark icon



- d. Now open firefox inside the virtual machine/ Ubuntu OS and type [www.neverssl.com](http://www.neverssl.com) in the search bar.



- e. Next, in wireshark tool, click on the red stop button to stop packet capture



**Note:** The Wireshark interface has five major components as shown in figure 1.2:

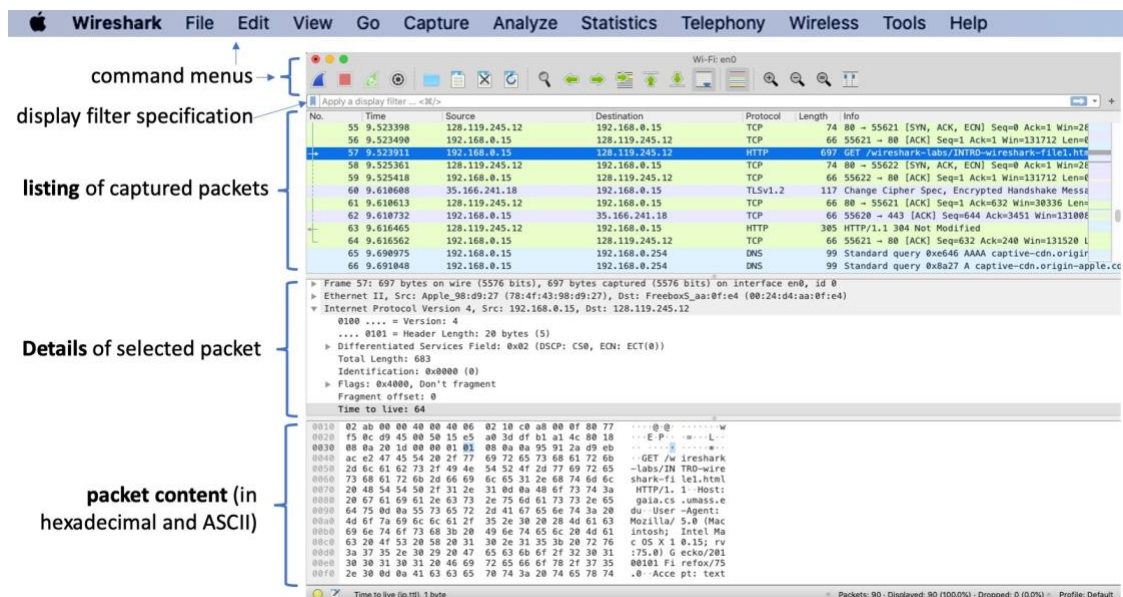


Figure 1.2: Wireshark packet capture sections

- The **command menus** are standard pulldown menus located at the top of the Wireshark window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data and exit the Wireshark application. The Capture menu allows you to begin packet capture.

- The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; note that this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.
- The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the plus/minus boxes or right/downward-pointing triangles to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.
- The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
- Towards the top of the Wireshark graphical user interface, is the **packet display filter** field, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.
- Additionally, look at the columns to identify time, source host, destination host, type of protocol, length of packet and information on the packet.

#### Task 4: Record your observation and answer the following questions.

1. List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window? Attach screen shots of your observation. What are these protocols used for?
2. On the display filter specification bar, type http and press enter. Attach screenshot of your result?
3. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display

*Format*, then select *Time-of-day*.) You may see several get request and Ok messages, you can pick any and record your finding. Attach a screenshot.

Answer: some milliseconds

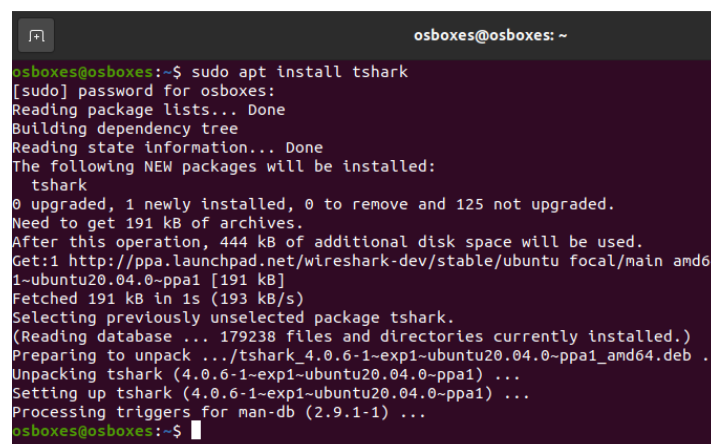
4. What is the Internet address of the [www.neverssl.com](http://www.neverssl.com)? What is the Internet address of your computer? Attach a screenshot for each of the answers.
5. What HTTP status codes do you see in the “info” column? What is the purpose of status codes?
6. Print the two HTTP messages (GET and OK) referred to in question 3 above. To do so, select *Print* from the Wireshark *File* command menu, and select the “*Selected Packet Only*” and “*Print as displayed*” radial buttons, and then click OK. Attach screenshots of the printed packets.

## Tshark

TShark is a network protocol analyzer in CLI mode. It lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file. TShark’s native capture file format is pcap format, which is also the format used by tcpdump and various other tools.

## Task 5: Reading captured files using Tshark

- a. Install Tshark from command line utility or terminal using command “`sudo apt install tshark`”



```
osboxes@osboxes: ~  
osboxes@osboxes:~$ sudo apt install tshark  
[sudo] password for osboxes:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  tshark  
0 upgraded, 1 newly installed, 0 to remove and 125 not upgraded.  
Need to get 191 kB of archives.  
After this operation, 444 kB of additional disk space will be used.  
Get:1 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal/main amd64  
tshark 4.0.6-1-exp1-ubuntu20.04.0-ppa1 [191 kB]  
Fetched 191 kB in 1s (193 kB/s)  
Selecting previously unselected package tshark.  
(Reading database ... 179238 files and directories currently installed.)  
Preparing to unpack .../tshark_4.0.6-1-exp1-ubuntu20.04.0-ppa1_amd64.deb ...  
Unpacking tshark (4.0.6-1-exp1-ubuntu20.04.0-ppa1) ...  
Setting up tshark (4.0.6-1-exp1-ubuntu20.04.0-ppa1) ...  
Processing triggers for man-db (2.9.1-1) ...  
osboxes@osboxes:~$
```

- b. Download the packet capture file to your virtual machine from the link <http://ethereal-trace-4>.

- c. Now read the captured file using command “tshark -r http-ethereal-trace-4 | head” as shown below

```
osboxes@osboxes:~/Desktop$ tshark -r http-ethereal-trace-4 | head
 1  0.000000 192.168.1.102 → 192.168.1.104 SNMP 92 get-request 1.3.6.1.4.1.1
 2  0.017529 192.168.1.104 → 192.168.1.102 SNMP 93 get-response 1.3.6.1.4.1.1
 3  3.017792 192.168.1.102 → 192.168.1.104 SNMP 92 get-request 1.3.6.1.4.1.1
```

- d. Use command “tshark -r http-ethereal-trace-4 ip.src==192.168.1.102” to list packets sourced from the host 192.168.1.102.
- e. Use command “tshark -r http-ethereal-trace-4 ip.dst==192.168.1.104” to list packets destined to the host 192.168.1.104.
- f. Use command “tshark -n -r http-ethereal-trace-4 -q -z conv,tcp” to print conversation of TCP protocol

### Task 6: Answer questions on Tshark Task 5

1. How many packets can you see when you run the command mentioned in option ‘c’. What protocols did you observe in the displayed window? Attach a screenshot.
2. How many packets are sourced from host 192.168.1.102? (You can get your answer when you run command shown in point ‘d’)?
3. How many packets are destined to the host 134.241.6.82?
4. Attach screenshot of output from task 5.f.

**Submission Guideline:** Submit a PDF version of your answers on canvas.