

## EXPERIMENT NO.8

**Aim:** Study and implement of packet sniffer tool: Wireshark

**Theory:**

Wireshark is a widely-used network protocol analyzer that allows users to capture and inspect network traffic in real-time or from stored data. It offers a user-friendly interface with powerful features such as packet filtering, protocol analysis, and detailed packet inspection. Wireshark is invaluable for network troubleshooting, protocol development, and network security analysis, providing deep insights into network behavior and aiding in the detection of anomalies and security threats. With its extensive capabilities and cross-platform support, Wireshark is an essential tool for network administrators, security analysts, developers, and anyone involved in managing or securing network infrastructure.

**Features of Wireshark:**

1. **Live Packet Capture:** Wireshark can capture live network traffic from various interfaces, allowing real-time analysis of data packets as they flow through the network.
2. **Deep Protocol Analysis:** It provides detailed insights into network protocols by allowing users to inspect packet headers, payloads, and other data fields. This depth of analysis helps in understanding protocol behavior and identifying issues.
3. **Flexible Filtering:** Wireshark offers powerful filtering capabilities to focus on specific types of network traffic based on various criteria such as IP addresses, port numbers, protocols, and packet contents. This helps in isolating and analyzing relevant packets efficiently.
4. **Customizable Display:** Users can customize the display of captured packets to suit their preferences and requirements. This includes options to configure packet views, colorization, and protocol hierarchy, enhancing readability and analysis efficiency.
5. **Comprehensive Statistics:** Wireshark provides detailed statistics on captured network traffic, including traffic volume, protocol usage, packet timing, and error rates. These statistics offer valuable insights into network performance and behavior.
6. **Cross-Platform Support:** Wireshark is available for multiple operating systems, including Windows, macOS, and Linux, ensuring broad compatibility across different environments. This allows network administrators and analysts to use the tool across various platforms seamlessly.

**Steps to download wireshark:**

1. Open ubuntu terminal

2. Install wireshark

# apt-get install wireshark

3. To know the name of your Ethernet interface: (Mostly it is "eth0")

#ifconfig

4. Start wireshark

#sudo wireshark

5. Once wireshark window opens, select the interface and click on start

### **Capturing Packets:**

After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface.

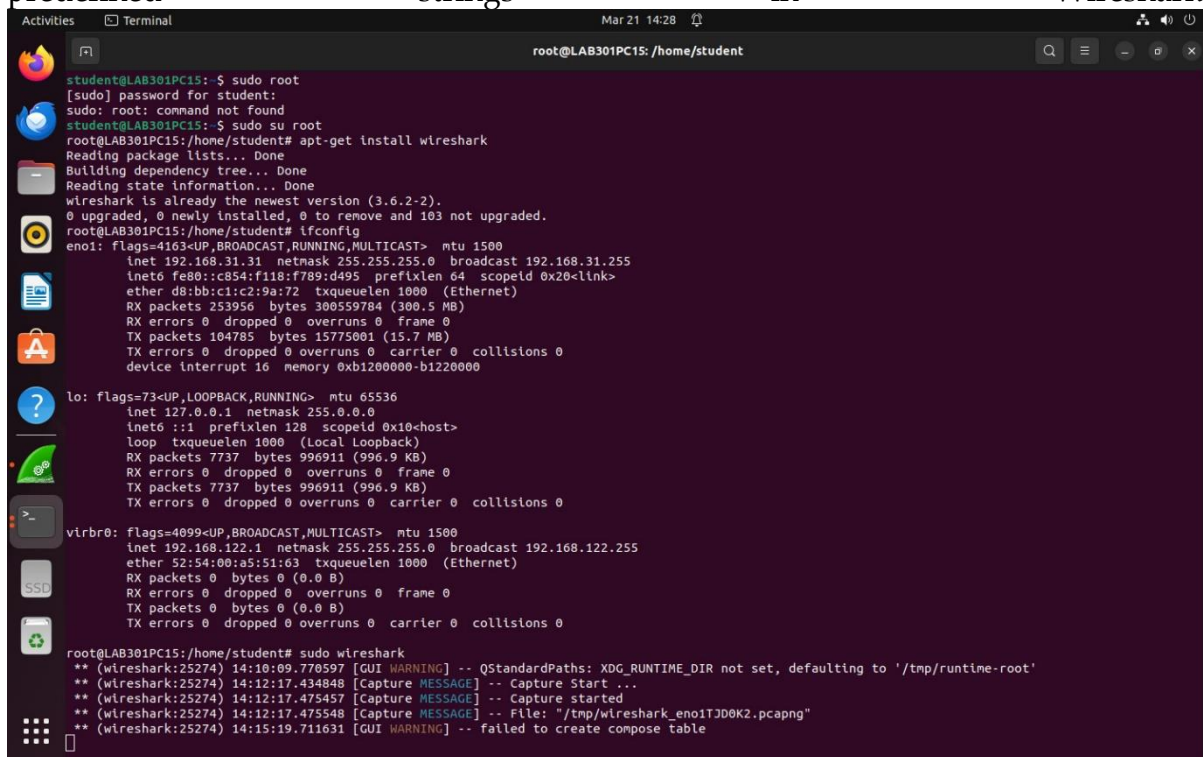
For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system.

Click the stop capture button near the top left corner of the window when you want to stop capturing traffic. Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order. Wireshark can record the capturing information in the file with extension .pcap (packet capture).

This file can be reopened for analysis in offline mode.

There is no need to remember filtering commands. Filters can be applied by putting predefined strings in Wireshark.



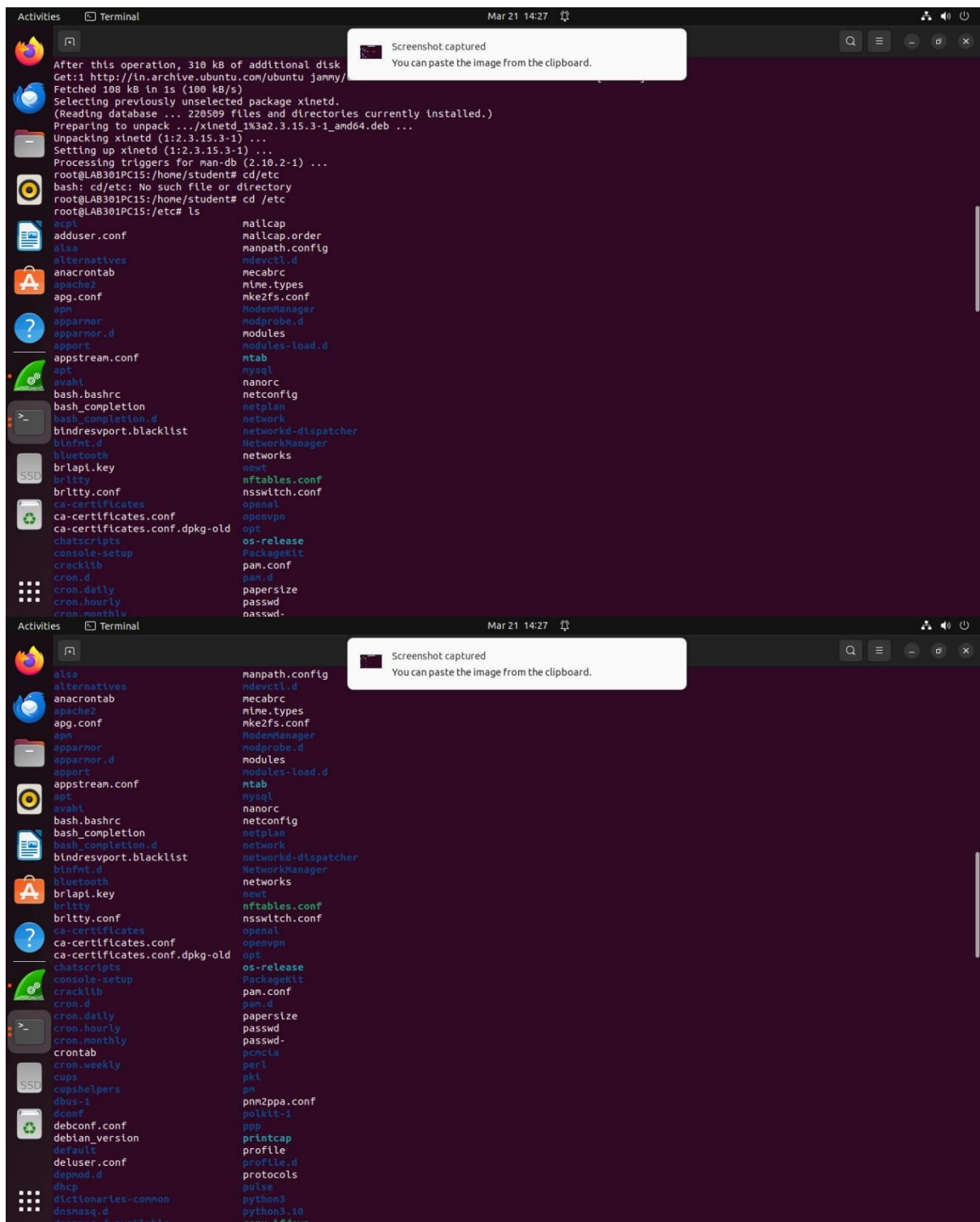
```
Activities Terminal Mar 21 14:28 root@LAB301PC15: /home/student

student@LAB301PC15: $ sudo root
[sudo] password for student:
sudo: root: command not found
student@LAB301PC15: $ sudo su root
root@LAB301PC15:/home/student# apt-get install wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wireshark is already the newest version (3.6.2-2).
0 upgraded, 0 newly installed, 0 to remove and 103 not upgraded.
root@LAB301PC15:/home/student# ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.31.31 netmask 255.255.255.0 broadcast 192.168.31.255
    inet6 fe80::c854:f118:f789:d495 prefixlen 64 scopeid 0x20<link>
    ether d8:bb:c1:c2:9a:72 txqueuelen 1000 (Ethernet)
    RX packets 253956 bytes 300559784 (300.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 104785 bytes 15775001 (15.7 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xb1200000-b1220000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 7737 bytes 996911 (996.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7737 bytes 996911 (996.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:a5:51:63 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@LAB301PC15:/home/student# sudo wireshark
** (Wireshark:25274) 14:10:09.770597 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (Wireshark:25274) 14:12:17.434848 [Capture MESSAGE] -- Capture Start ...
** (Wireshark:25274) 14:12:17.475457 [Capture MESSAGE] -- Capture started
** (Wireshark:25274) 14:12:17.475548 [Capture MESSAGE] -- File: "/tmp/wireshark_eno1TJD0K2.pcapng"
** (Wireshark:25274) 14:15:19.711631 [GUI WARNING] -- failed to create compose Table
```



```
Activities Terminal Mar 21 14:27
libblockdev UPower
libbverbs.d usb_modeswitch.config
libl3 usb_modeswitch.d
libpaper.d vdpau_wrapper.cfg
libreoffice vte
libvirt vsftpd.conf
libxrandr vtrgb
locale.alias vulkan
locale.gen wgetrc
localtime wireshark
logcheck wpa_supplicant
login.defs X11
logrotate.conf xattr.conf
logrotate.d xdg
lsb-release xinetd.conf
lvm xinetd.d
machine-id xml
magic zsh_command_not_found
magic.mime
root@LAB301PC15:/etc# gedit vsftpd.conf

(gedit:26096): dconf-WARNING **: 14:23:37.015: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)
(gedit:26096): dconf-WARNING **: 14:23:37.018: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)
(gedit:26096): dconf-WARNING **: 14:23:37.127: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)
(gedit:26096): dconf-WARNING **: 14:23:37.127: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)
(gedit:26096): dconf-WARNING **: 14:23:37.127: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)
(gedit:26096): dconf-WARNING **: 14:23:37.127: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)
** (gedit:26096): WARNING **: 14:23:57.004: Set document metadata failed: Setting attribute metadata::gedit-spell-language not supported
** (gedit:26096): WARNING **: 14:23:57.004: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported
** (gedit:26096): WARNING **: 14:24:12.141: Set document metadata failed: Setting attribute metadata::gedit-position not supported
(gedit:26096): dconf-WARNING **: 14:24:12.169: failed to commit changes to dconf: Failed to execute child process "dbus-launch" (No such file or directory)
root@LAB301PC15:/etc# service xinetd restart
root@LAB301PC15:/etc# service vsftpd restart

Activities Terminal Mar 21 14:27
student@LAB301PC15:~$ sudo su root
[sudo] password for student:
root@LAB301PC15:/home/student# sudo apt-get install vsftpd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 103 not upgraded.
Need to get 123 kB of archives.
After this operation, 326 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 vsftpd amd64 3.0.5-0ubuntu1 [123 kB]
Fetched 123 kB in 1s (136 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 220452 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.5-0ubuntu1_amd64.deb ...
Unpacking vsftpd (3.0.5-0ubuntu1) ...
Setting up vsftpd (3.0.5-0ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /lib/systemd/system/vsftpd.service.
Detected unsafe path transition / (owned by admin) → /run (owned by root) during canonicalization of /run/vsftpd.
Processing triggers for man-db (2.10.2-1) ...
root@LAB301PC15:/home/student# sudo apt-get install xinetd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  xinetd
0 upgraded, 1 newly installed, 0 to remove and 103 not upgraded.
Need to get 108 kB of archives.
After this operation, 310 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 xinetd amd64 1:2.3.15.3-1 [108 kB]
Fetched 108 kB in 1s (100 kB/s)
Selecting previously unselected package xinetd.
(Reading database ... 220509 files and directories currently installed.)
Preparing to unpack .../xinetd_1%3a2.3.15.3-1_amd64.deb ...
Unpacking xinetd (1:2.3.15.3-1) ...
Setting up xinetd (1:2.3.15.3-1) ...
Processing triggers for man-db (2.10.2-1) ...
root@LAB301PC15:/home/student# cd /etc
bash: cd /etc: No such file or directory
root@LAB301PC15:/home/student# cd /etc
root@LAB301PC15:/etc# ls
acpi mailcap
adduser.conf mailcap.order
alsa nannath.config
```



The screenshot shows a Linux desktop environment with a terminal window and a text editor window. The terminal window is titled 'root@LAB301PC15: /etc' and shows the following commands and output:

```
student@LAB301PC15: $ ping google.com
PING google.com (142.250.77.78) 56(84) bytes of data:
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=1 ttl=59 time=6.25 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=2 ttl=59 time=3.83 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=3 ttl=59 time=3.45 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=4 ttl=59 time=3.36 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=5 ttl=59 time=3.49 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=6 ttl=59 time=3.30 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=7 ttl=59 time=2.95 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=8 ttl=59 time=6.37 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=9 ttl=59 time=3.30 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=10 ttl=59 time=3.31 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=11 ttl=59 time=2.68 ms
64 bytes from bon07s27-in-f14.1e100.net (142.250.77.78): icmp_seq=12 ttl=59 time=3.32 ms
^C
--- google.com ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11017ms
rtt min/avg/max/mdev = 2.678/3.800/6.373/1.153 ms
student@LAB301PC15: $ ip.addr = 192.168.42.3
Command 'ip.addr' not found, did you mean:
  command 'ipmaddr' from deb net-tools (1.60+git20181103.0eebece-1ubuntu5)
Try: sudo apt install <deb name>
student@LAB301PC15: $ ip.addr = 192.168.42.3
Command 'ip.addr' not found, did you mean:
  command 'ipmaddr' from deb net-tools (1.60+git20181103.0eebece-1ubuntu5)
Try: sudo apt install <deb name>
student@LAB301PC15: $ ipmaddr = 192.168.42.3
Usage: ipmaddr [ add | del ] MULTIADDR dev STRING
       ipmaddr show [ dev STRING ] [ ipv4 | ipv6 | link | all ]
       ipmaddr -V | -version
student@LAB301PC15: $ sudo su root
[sudo] password for student:
root@LAB301PC15:/home/student# sudo apt-get install vsftpd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 103 not upgraded.
Need to get 123 kB of archives.
After this operation, 326 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 vsftpd amd64 3.0.5-0ubuntu1 [123 kB]
Fetched 123 kB in 1s (136 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 220452 files and directories currently installed.)
```

The text editor window is titled 'vsftpd.conf /etc' and shows the following content:

```
1 # Example config file /etc/vsftpd.conf
2 #
3 # The default compiled in settings are fairly paranoid. This sample file
4 # loosens things up a bit, to make the ftp daemon more usable.
5 # Please see vsftpd.conf.5 for all compiled in defaults.
6 #
7 # READ THIS: This example file is NOT an exhaustive list of vsftpd options.
8 # Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
9 # capabilities.
10 #
11 #
12 # Run standalone? vsftpd can run either from an inetd or as a standalone
13 # daemon started from an initscript.
14 listen=NO
15 #
16 # This directive enables listening on IPv6 sockets. By default, listening
17 # on the IPv6 "any" address (::) will accept connections from both IPv6
18 # and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
19 # sockets. If you want that (perhaps because you want to listen on specific
20 # addresses) then you must run two copies of vsftpd with two configuration
21 # files.
22 listen_ipv6=YES
23 #
24 # Allow anonymous FTP? (Disabled by default).
25 anonymous_enable=YES
26 #
27 # Uncomment this to allow local users to log in.
28 local_enable=YES
29 #
30 # Uncomment this to enable any form of FTP write command.
31 write_enable=YES
32 #
33 # Default umask for local users is 077. You may wish to change this to 022,
34 # if your users expect that (022 is used by most other ftpd's)
35 local_umask=022
36 #
37 # Uncomment this to allow the anonymous FTP user to upload files. This only
```

## Commands:-

### 1. Capturing packets of a particular host

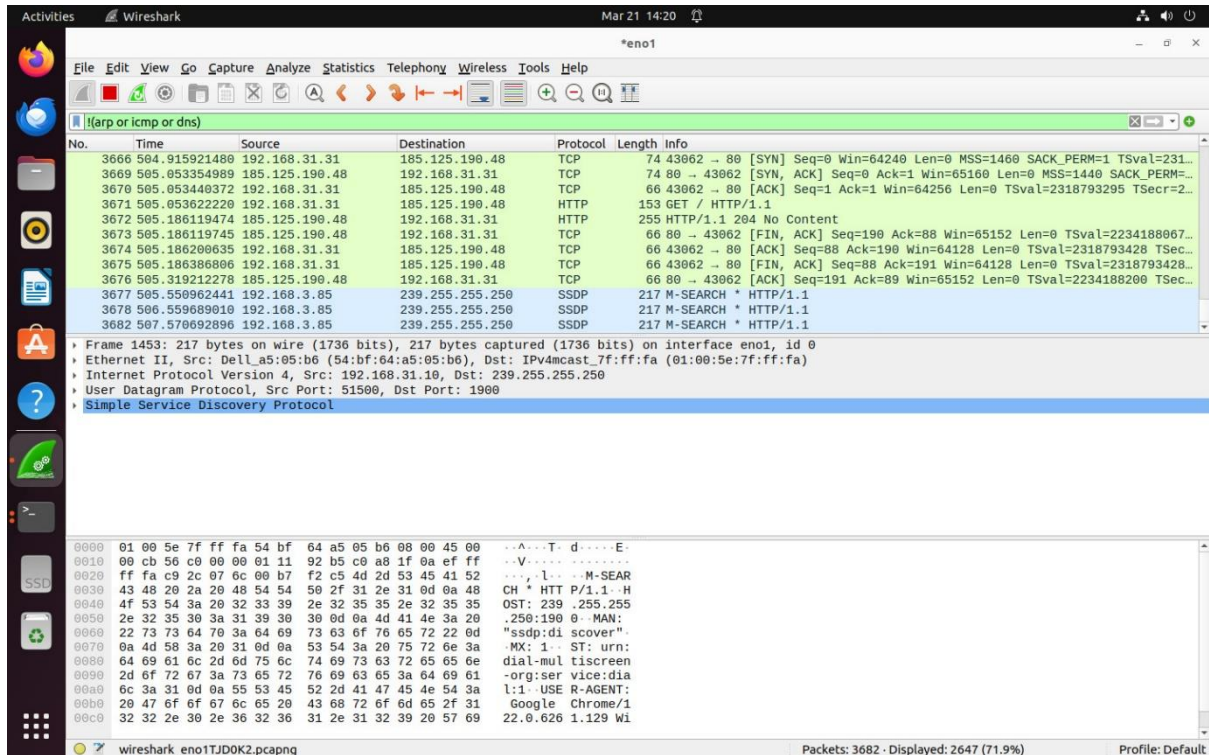
`ip.addr == 192.163.31.45`

Sets a filter for any packet with 192.163.31.45, as either the source or destination.

### 2. To capture a conversation between specified hosts

ip.addr == 192.163.31.45 && ip.addr == 104.26.2.23

Sets a conversation filter between the two defined IP addresses



The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type `—dns` and you'll see only DNS packets. When you start typing, Wireshark will help you auto complete your filter.

## Commands:-

1. To filter packets for a specific protocol : `http`

Activities Wireshark Mar 21 14:20

\*eno1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request

No.	Time	Source	Destination	Protocol	Length	Info
3324	468.026479191	192.168.3.85	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3329	468.925605676	192.44.44.202	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3330	468.925606036	192.44.44.202	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3336	469.033273730	192.168.3.85	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3339	469.941369469	192.44.44.202	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3340	469.941369865	192.44.44.202	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3342	470.035293232	192.168.3.85	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3359	470.956869286	192.44.44.202	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3360	470.956869631	192.44.44.202	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
3384	471.627602550	192.168.31.17	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
3466	474.627255088	192.168.31.17	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
3499	477.641736101	192.168.31.17	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1

Frame 1453: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface eno1, id 0

Ethernet II, Src: Dell\_a5:05:b6 (54:b6:a5:05:b6), Dst: IPv4mcast\_7f:ff:fa (01:00:5e:7f:ff:fa)

Internet Protocol Version 4, Src: 192.168.31.10, Dst: 239.255.255.250

User Datagram Protocol, Src Port: 51500, Dst Port: 1900

Simple Service Discovery Protocol

0000 01 00 5e 7f ff fa 54 b6 64 a5 05 b6 08 00 45 00 ...A...T...d...E...  
0010 00 cb 56 c0 00 00 01 11 92 b5 c0 a8 1f 0a ef ff ...V... ..  
0020 ff fa c9 2c 07 6c 00 b7 f2 c5 4d 2d 53 45 41 52 ...L... ..M-SEAR  
0030 43 48 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48 CH \* HTTP/1.1..H  
0040 4f 53 54 3a 20 32 33 39 2e 32 35 35 2e 32 35 35 OST: 239.255.255.250  
0050 2e 32 35 30 3a 31 39 30 30 0d 0a 4d 41 4e 3a 20 .250:1900..MAN:  
0060 22 73 73 64 79 3a 64 69 73 63 6f 76 65 72 22 0d "ssdp:discover".  
0070 0a 4d 50 3a 20 31 0d 0a 53 54 3a 20 75 72 6e 3a MX: 1..ST: urn:  
0080 64 69 61 6c 2d 6d 75 6c 74 69 73 63 72 65 65 6e dial-multiscreen  
0090 2d 6f 72 6f 73 73 65 72 76 69 63 65 3a 64 69 61 -org:service:dia  
00a0 6c 3a 31 0d 0a 55 53 45 52 d4 47 45 4e 54 3a l:1-USE R-AGENT:  
00b0 20 47 6f 6f 6f 6c 65 20 43 68 72 6f 6d 65 2f 31 Google Chrome/1  
00c0 32 32 2e 30 2e 36 32 36 31 2e 31 32 39 20 5f 69 22.0.626 1.129 Wi

Request: Boolean Packets: 3500 - Displayed: 668 (19.1%) Profile: Default

Activities Wireshark Mar 21 14:18

\*eno1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.reset=0

No.	Time	Source	Destination	Protocol	Length	Info
1040	106.851198981	192.168.31.31	162.213.33.48	TCP	74	39088 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=86...
1051	107.855891618	192.168.31.31	162.213.33.48	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 39088 → 443 [SYN] Seq=...
1061	108.879792734	192.168.31.31	162.213.33.48	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 39088 → 443 [SYN] Seq=...
1070	109.903778441	192.168.31.31	162.213.33.48	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 39088 → 443 [SYN] Seq=...
1077	110.927889319	192.168.31.31	162.213.33.48	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 39088 → 443 [SYN] Seq=...
1081	111.951781236	192.168.31.31	162.213.33.48	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 39088 → 443 [SYN] Seq=...
1090	113.907792044	192.168.31.31	162.213.33.48	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 39088 → 443 [SYN] Seq=...
1456	204.910478566	192.168.31.31	91.189.91.98	TCP	74	40162 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=130...
1458	205.118600276	91.189.91.98	192.168.31.31	TCP	74	80 → 40162 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1440 SACK_PERM=...
1459	205.118681087	192.168.31.31	91.189.91.98	TCP	66	40162 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1394942343 TSecr=1...
1460	205.118854833	192.168.31.31	91.189.91.98	HTTP	153	GET / HTTP/1.1
1461	205.327323103	91.189.91.98	192.168.31.31	HTTP	251	HTTP/1.1 204 No Content

Frame 1456: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eno1, id 0

Ethernet II, Src: Micro-St\_c2:9a:72 (d8:bb:c1:c2:9a:72), Dst: 9c:53:22:05:6a:19 (9c:53:22:05:6a:19)

Internet Protocol Version 4, Src: 192.168.31.31, Dst: 91.189.91.98

Transmission Control Protocol, Src Port: 40162, Dst Port: 80, Seq: 0, Len: 0

0000 9c 53 22 05 6a 19 d8 bb c1 c2 9a 72 08 00 45 00 ...S...j... ..E...  
0010 00 3c 4d 71 40 00 40 06 56 64 c0 a8 1f 1f 5b bd ...<Mq@.@...Vd...[...  
0020 5b 62 9c e2 00 50 44 12 0a 8f 00 00 00 0a 02 [b...PD... ..  
0030 fa f0 97 15 00 00 02 04 05 b4 04 02 08 0a 53 25 .....%\$  
0040 20 b7 00 00 00 00 01 03 03 07

wireshark\_eno1TJ0K2.pcapng Packets: 2646 - Displayed: 20 (0.8%) Profile: Default



Activities Wireshark Mar 21 14:17

\*eno1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port==80

No.	Time	Source	Destination	Protocol	Length	Info
1456	204.919478506	192.168.31.31	91.189.91.98	TCP	74	40162 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1394...
1458	205.118600276	91.189.91.98	192.168.31.31	TCP	74	80 → 40162 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1440 SACK_PERM=1...
1459	205.118601087	192.168.31.31	91.189.91.98	TCP	66	40162 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1394942343 TSecr=18...
1460	205.118654833	192.168.31.31	91.189.91.98	HTTP	153	GET / HTTP/1.1
1461	205.327323103	91.189.91.98	192.168.31.31	HTTP	251	HTTP/1.1 204 No Content
1462	205.327323362	91.189.91.98	192.168.31.31	TCP	66	80 → 40162 [FIN, ACK] Seq=186 Ack=88 Win=65152 Len=0 TSval=1801378536 ...
1463	205.327398933	192.168.31.31	91.189.91.98	TCP	66	40162 → 80 [ACK] Seq=88 Ack=186 Win=64128 Len=0 TSval=1394942552 TSecr...
1464	205.327579338	192.168.31.31	91.189.91.98	TCP	66	40162 → 80 [FIN, ACK] Seq=88 Ack=187 Win=64128 Len=0 TSval=1394942552 ...
1465	205.536095508	91.189.91.98	192.168.31.31	TCP	66	80 → 40162 [ACK] Seq=187 Ack=89 Win=65152 Len=0 TSval=1801378744 TSecr...

Frame 1456: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eno1, id 0

Ethernet II, Src: Micro-St\_c2:9a:72 (d8:bb:c1:c2:9a:72), Dst: 9c:53:22:05:6a:19 (9c:53:22:05:6a:19)

Internet Protocol Version 4, Src: 192.168.31.31, Dst: 91.189.91.98

Transmission Control Protocol, Src Port: 40162, Dst Port: 80, Seq: 0, Len: 0

0000 9c 53 22 05 6a 19 d8 bb c1 c2 9a 72 08 00 45 00 .S".j...-...E-  
0010 00 3c 4d 71 40 00 00 06 56 64 c0 a8 1f 1f 5b bd .<Mq@ @: Vd....[  
0020 5b 62 9c e2 00 50 44 12 0a 8f 00 00 00 00 a0 02 [b...PD:.....  
0030 fa f0 97 15 00 00 02 04 05 b4 04 02 08 0a 53 25 .....S%  
0040 20 b7 00 00 00 01 03 03 07 .....

wireshark\_eno1TJ00K2.pcapng Packets: 2308 - Displayed: 9 (0.4%) Profile: Default

Activities Wireshark Mar 21 14:17

\*eno1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

Draw packets using your coloring rules

No.	Time	Source	Destination	Protocol	Length	Info
947	98.171550826	203.212.24.46	192.168.31.31	DNS	97	Standard query response 0xa97b A google.com A 142.250.77.78 OPT
948	98.171551078	203.212.24.46	192.168.31.31	DNS	109	Standard query response 0xfba3 AAAA google.com AAAA 2404:6800:4009:81...
951	98.179630217	192.168.31.31	203.212.24.46	DNS	97	Standard query 0x22f5 PTR 78.77.250.142.in-addr.arpa OPT
952	98.183509869	203.212.24.46	192.168.31.31	DNS	136	Standard query response 0x22f5 PTR 78.77.250.142.in-addr.arpa PTR bom...
1036	106.848316305	192.168.31.31	203.212.24.46	DNS	89	Standard query 0xed8c A metrics.ubuntu.com OPT
1037	106.848523037	192.168.31.31	203.212.24.46	DNS	89	Standard query 0x7aac AAAA metrics.ubuntu.com OPT
1038	106.850656096	203.212.24.46	192.168.31.31	DNS	169	Standard query response 0xed8c A metrics.ubuntu.com A 162.213.33.48 N...
1039	106.850744275	203.212.24.46	192.168.31.31	DNS	150	Standard query response 0x7aac AAAA metrics.ubuntu.com SOA ns1.canonl...
1088	114.906532016	192.168.31.31	203.212.24.46	DNS	100	Standard query 0x3ccd AAAA connectivity-check.ubuntu.com OPT
1089	114.909093196	203.212.24.46	192.168.31.31	DNS	500	Standard query response 0x3ccd AAAA connectivity-check.ubuntu.com AAA...
1454	204.906051754	192.168.31.31	203.212.24.46	DNS	100	Standard query 0x6720 A connectivity-check.ubuntu.com OPT
1455	204.908835347	203.212.24.46	192.168.31.31	DNS	350	Standard query response 0x6720 A connectivity-check.ubuntu.com A 91.1...

Frame 1455: 356 bytes on wire (2848 bits), 356 bytes captured (2848 bits) on interface eno1, id 0

Ethernet II, Src: 9c:53:22:05:6a:19 (9c:53:22:05:6a:19), Dst: Micro-St\_c2:9a:72 (d8:bb:c1:c2:9a:72)

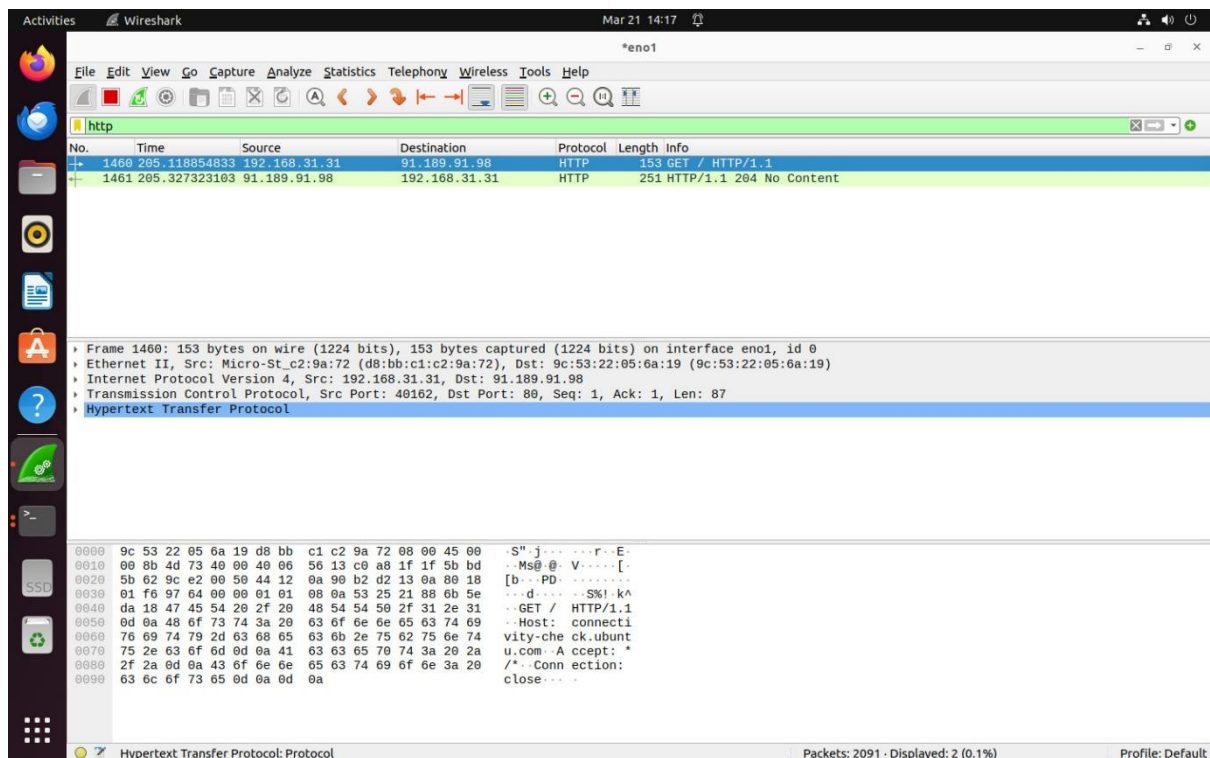
Internet Protocol Version 4, Src: 203.212.24.46, Dst: 192.168.31.31

User Datagram Protocol, Src Port: 53, Dst Port: 57962

Domain Name System (response)

0000 d8 bb c1 c2 9a 72 9c 53 22 05 6a 19 08 00 45 00 .....S".j...E-  
0010 01 56 5f cd 00 00 3d 11 59 00 cb d4 18 2e c0 a8 -V\_...= Y.....  
0020 1f 1f 00 35 e2 0a 01 42 d4 bf 67 20 81 80 00 01 .5.j.B..g.....  
0030 00 0c 00 03 00 01 12 63 6f 6e 6e 65 63 74 69 76 ....c onnectiv  
0040 09 74 79 2d 63 08 05 63 0b 06 75 62 75 6e 74 75 ity-chec k.ubunt  
0050 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00 01 00 .com.....  
0060 00 00 1b 00 04 5b bd 5b 62 c0 0c 00 01 00 01 00 ...[ [ b.....  
0070 00 00 1b 00 04 b9 7d be 30 c0 0c 00 01 00 01 00 .....} b.....  
0080 00 00 1b 00 04 b9 7d be 62 c0 0c 00 01 00 01 00 .....} 1.....  
0090 00 00 1b 00 04 b9 7d be 31 c0 0c 00 01 00 01 00 .....} a.....  
00a0 00 00 1b 00 04 b9 7d be 61 c0 0c 00 01 00 01 00 .....[ [ 1.....  
00b0 00 00 1b 00 04 5b bd 5b 31 c0 0c 00 01 00 01 00 .....}.....  
00c0 00 00 1b 00 04 b9 7d be 12 c0 0c 00 01 00 01 00 .....}.....

Domain Name System: Protocol Packets: 2116 - Displayed: 14 (0.7%) Profile: Default



## To capture packets from the FTP server. (Login ID and Password)

What is FTP?

FTP stands for File Transfer Protocol. As the name suggest this network protocol allows you to transfer files or directories from one host to another over the network whether it is your LAN or Internet. The package required to install FTP is known as VSFTPD (Very Secure File Transfer Protocol Daemon)/

### **Steps:-**

1. Get root access: `$ sudo su root`
2. Find your ip address: `# ifconfig`

### Installation of FTP server in Ubuntu

Name of Packages required: VSFTPD, XINETD

1. `# sudo apt-get install vsftpd`
2. `# sudo apt-get install xinetd`

The above command will install and start the xinetd superserver on your system. The chances are that you already have xinetd installed on your system. In that case you can omit the above installation command. In the next step we need to edit the FTP server's configuration file which is present in `/etc/vsftpd.conf`.

3. `# cd /etc`
4. `# ls`
5. `# gedit vsftpd.conf`

Change the following line:

Anonymous\_enable=NO To Anonymous\_enable=YES

This will instruct the FTP server to allow connecting with an anonymous client.

6. Save and close the gedit file

Now, that we are ready we can start the FTP server in the normal mode with:

7. # service xinetd restart

8. # service vsftpd restart OR # init.d/vsftpd restart

Connecting to a client present in other machine

\$ ftp ip address of the FTP server

Name: anonymous

Please specify the password.

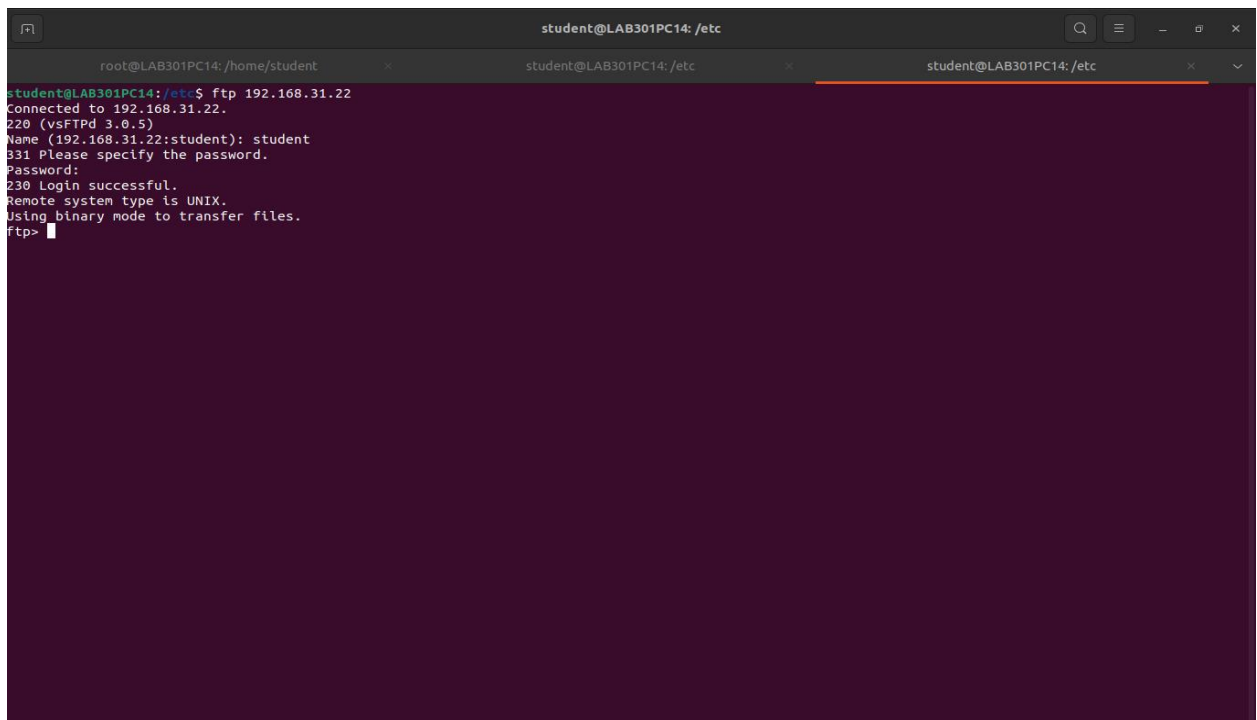
Password:

Login successful. (even if the login is not successful then also wireshark will capture the id and password)

ftp>

ftp> quit

Goodbye.



```
student@LAB301PC14: /etc
root@LAB301PC14: /home/student
student@LAB301PC14: /etc
student@LAB301PC14: /etc
student@LAB301PC14:/etc$ ftp 192.168.31.22
Connected to 192.168.31.22.
220 (vsFTPd 3.0.5)
Name (192.168.31.22:student): student
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Start WIRESHARK. In the FILTER field put FTP. This will filter all FTP packets

Wireshark interface showing a packet capture on interface eno1. The packet list displays FTP traffic between 192.168.31.22 and 192.168.31.45. The packet details pane shows the structure of the captured packet, including Ethernet II, Internet Protocol Version 4, and File Transfer Protocol (FTP) fields. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
56	17.359918009	192.168.31.22	192.168.31.45	FTP	86	Response: 220 (vsFTPD 3.0.5)
90	27.155026611	192.168.31.45	192.168.31.22	FTP	80	Request: USER student
92	27.155711847	192.168.31.22	192.168.31.45	FTP	100	Response: 331 Please specify the password.
123	32.170990473	192.168.31.45	192.168.31.22	FTP	83	Request: PASS complab301
126	32.239057740	192.168.31.22	192.168.31.45	FTP	89	Response: 230 Login successful.
128	32.239245595	192.168.31.45	192.168.31.22	FTP	72	Request: SYST
130	32.239748459	192.168.31.22	192.168.31.45	FTP	85	Response: 215 UNIX Type: L8
131	32.239998442	192.168.31.45	192.168.31.22	FTP	72	Request: FEAT
132	32.240397163	192.168.31.22	192.168.31.45	FTP	81	Response: 211-Features:
133	32.240397364	192.168.31.22	192.168.31.45	FTP	87	Response: EPRT
135	32.240713540	192.168.31.22	192.168.31.45	FTP	110	Response: PASV

Frame 56: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface eno1, id 0  
 Ethernet II, Src: Micro-St\_c2:9e:09 (d8:bb:c1:c2:9e:09), Dst: Micro-St\_c2:9d:17 (d8:bb:c1:c2:9d:17)  
 Internet Protocol Version 4, Src: 192.168.31.22, Dst: 192.168.31.45  
 Transmission Control Protocol, Src Port: 21, Dst Port: 43842, Seq: 1, Ack: 1, Len: 20  
 File Transfer Protocol (FTP)  
 [Current working directory: ]

```

0000  d8 bb c1 c2 9d 17 d8 bb c1 c2 9e 09 08 00 45 00  .....E:
0010  00 48 dc cf 40 00 40 06 9e 4c c0 a8 1f 16 c0 a8  H..@..L....
0020  1f 2d 00 15 ab 42 02 b4 87 20 60 c6 72 20 80 18  ....B....r..
0030  91 fe f9 fb 00 00 01 01 08 0a d9 3f ca 52 cf f5  -.....? R..
0040  e3 68 32 32 30 20 28 76 73 46 54 50 64 20 33 2e  .h220 (v sFTPD 3.
0050  30 2e 35 29 0d 0a  .....0.5)
  
```

While the client is establishing a connection with the FTP server, the Wireshark running in the background of the FTP server is able to capture all FTP packets. So, the Name and Password entered by the client is visible in plain text in Wireshark. Apart from that the source and the destination address is also visible. If many clients are trying to connect with the server then source address, name and password are visible for all of them.