Practical Wireshark

Theory ⇒

What is Wireshark ?

- Wireshark is **network traffic analyzer** tool
- Very essential tool for any security professional or system admins
- A network packet analyzer will try to **capture network packets** and tries to **display that packet data** as details as possible
- Best tool for **troubleshooting issues in your network**

Wireshark Features ⇛

- Capture live packet data from a **network interface or more**
- Open **files containing packet data** captured with tcpdump/WinDUMP, and many **other packet capture programs**
- Display Packets with **very detailed protocol information**
- **Filer** and **Search** for packets is very useful
- Colorization of Packet is most useful feature of Wireshark

🔥 Remember ⇒

Wireshark is not tool for **Intrusion Detection System [IDS]**

Also, Wireshark will not **manipulate things on the network**

Wireshark is only use to **measure** packets
Wireshark doesn't send packets on the network

Who Use Wireshark ?

- Network Administrator
  - use it to **troubleshoot network problems, monitoring**
- QA Engineers
  - use it to **verify network application**
- Network Security Engineers
  - use it to **examine security problems**
- Developers
  - use it to **debug protocol implementations**

> So, what actually Wireshark do ?
>
> Wireshark intercepts traffic and converts that binary traffic into human-readable format.
>
> This makes it easy to identify what traffic is crossing your network, how much of it, how frequently, how much latency there is between certain hops, and so forth.

> What things you know from Wireshark as important concept ?
>
> - Analyze IP Packets
> - What is Packet ?
> - Protocols

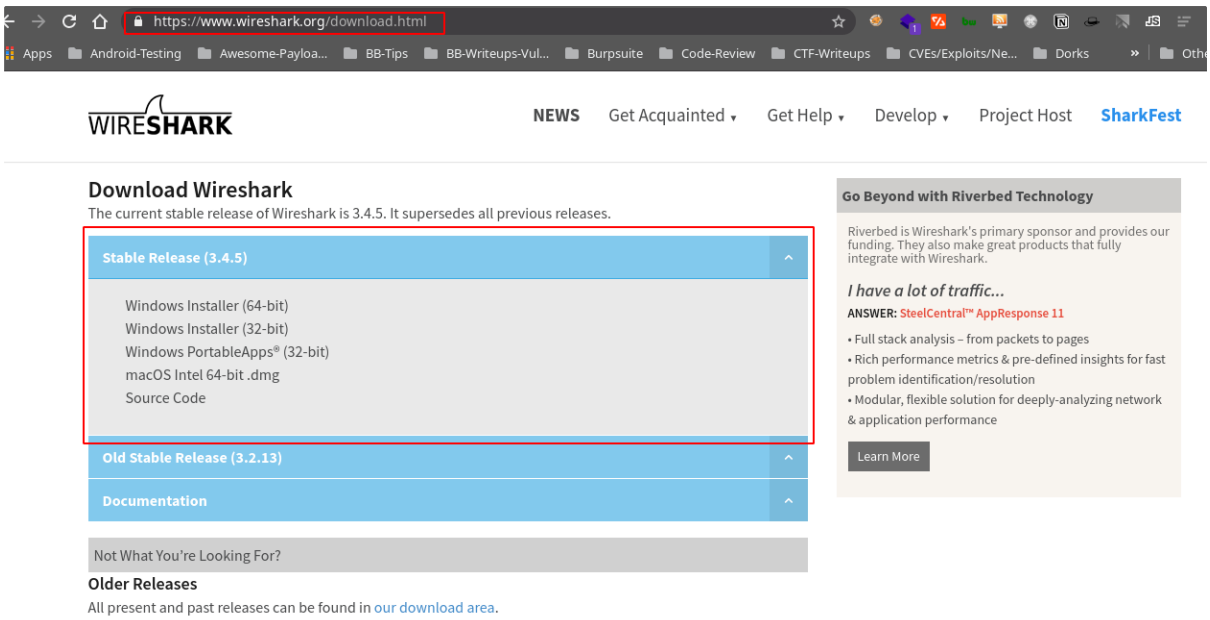
> A Packet **is a single message from any network protocol (i.e., TCP, DNS, etc.)**

How to Download wireshark ?

Downloading and installing Wireshark is easy.

Step one is to check the official Wireshark Download page for the operating system you need.

The basic version of Wireshark is free.



Download wireshark according to your operating system

You can download wireshark for windows and mac os from there

Wireshark for Linux

Installing Wireshark on Linux can be a little different depending on the Linux distribution.

Common steps in linux to download wireshark and configure

> sudo apt install wireshark
>
> sudo dpkg-reconfigure wireshark-common
>
> sudo adduser $USER wireshark

🔥 For Kali Linux ⇒

Wireshark is probably already installed! It's part of the basic package.

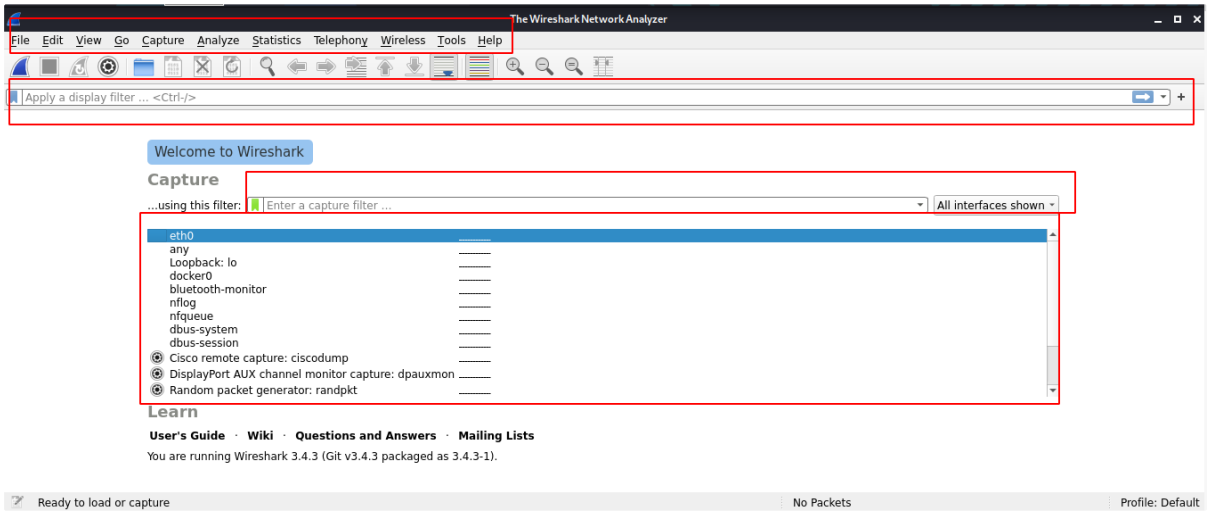
Check your menu to verify.

It's under the menu option "Sniffing & Spoofing."

You can run wireshark in terminal also ⇒ by just typing **wireshark** in terminal console

Start Wireshark from Kali Linux

- Type `sudo wireshark` in terminal of kali linux



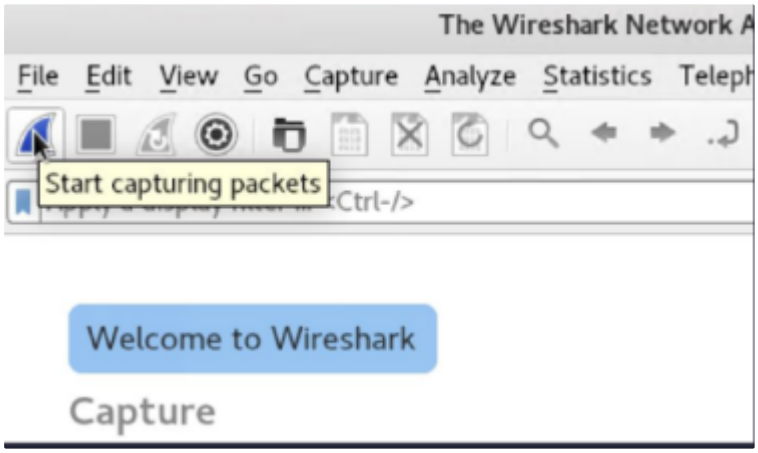
- Highlighted things is important for us to know

🔥 Remember ⇒

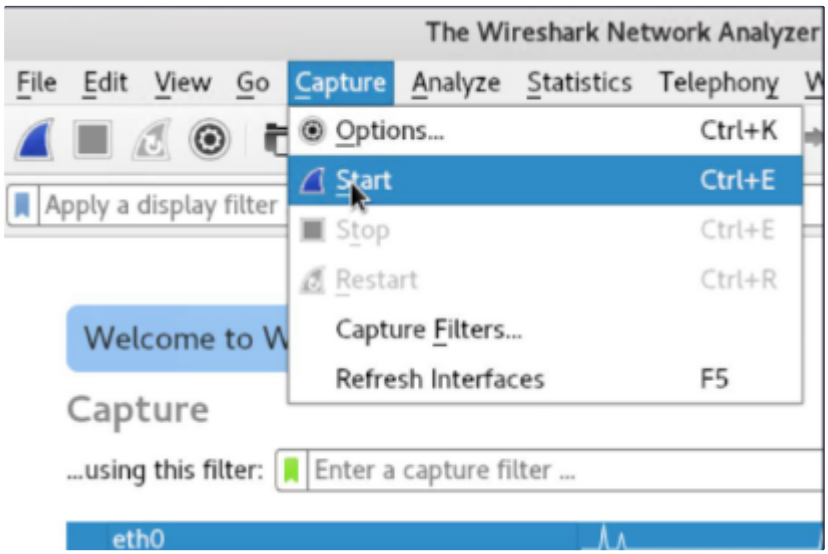
In virtual machine , you will not find **inbuilt wireless adapter (wlan0)** Because virtual machines only have **virtual (indirect) access to your hardware**

So, in virtual machine you will always find a wired network adapter ⇒ **[eth0]**

- You can select one or more of the network interfaces using “shift left-click."
- Once you have the network interface selected, you can start the capture, and there are several ways to do that
- Click the first button on the toolbar, titled “Start Capturing Packets.”

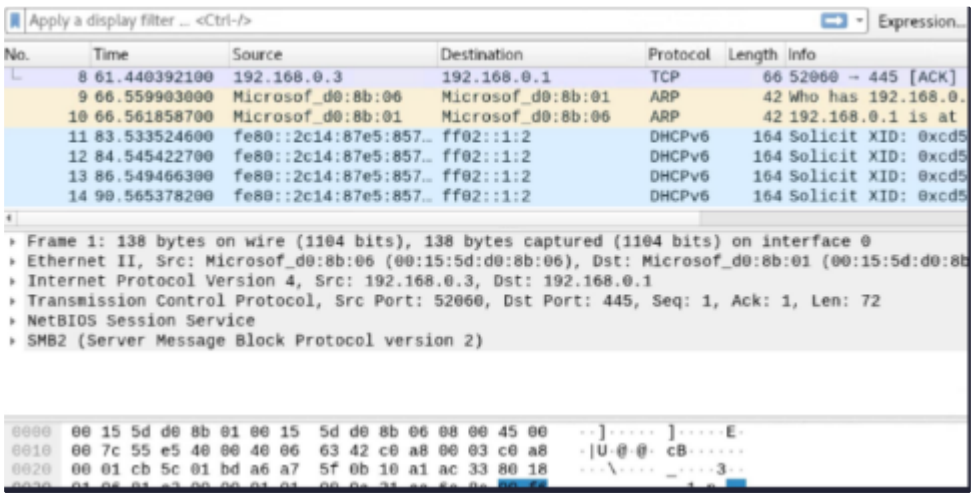


- You can select the menu item Capture -> Start.



- Or you could use the keystroke Control – E.

- During the capture, Wireshark will show you the packets that it captures in real-time.



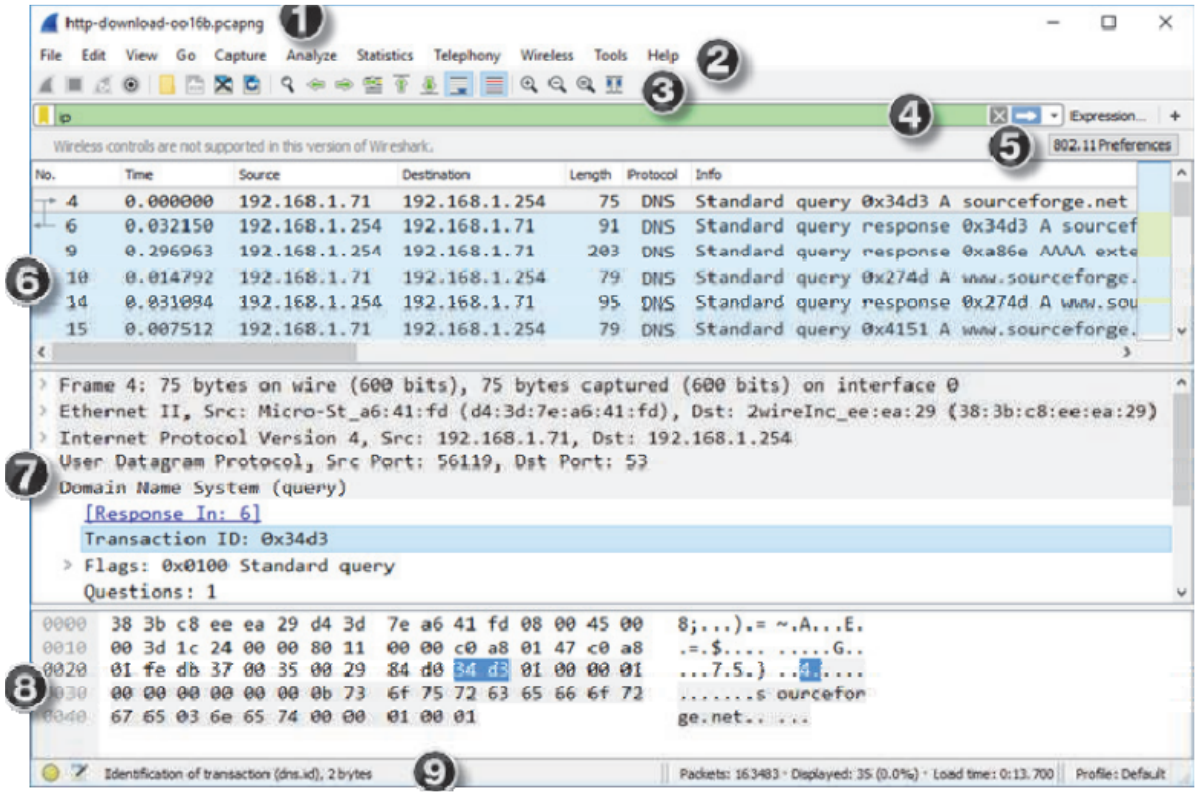
- Once you have captured all the packets you need, you use the same buttons or menu options to stop the capture.

🔥 Best practice says that you should stop Wireshark packet capture before you do analysis.

Before analyzing packet , you must know some basic things

- Wireshark shows you three different panes for inspecting packet data
- The Packet List, the top pane, is a list of all the packets in the capture
- When you click on a packet, the other two panes change to show you the details about the selected packet
- You can also tell if the packet is part of a conversation.

Graphical Interface of wireshark ⇒



(1) **Title bar** – trace file name, capture source, or "The Wireshark Network Analyzer"
(2) **Main menu** – standard menu
(3) **Main toolbar** – learn to use this set of icon buttons!
(4) **Display Filter and Filter Expressions area** – focus on specific traffic
(5) **Wireless toolbar** – define 802.11 settings
(6) **Packet List pane** – packet relationship indicator and summary of each frame
(7) **Packet Details pane** – dissected frames
(8) **Packet Bytes pane** – hex and ASCII details
(9) **Status Bar** – access to the Expert, annotations, packet counts, and profiles

You will confuse about their icons for what meaning and purpose, So ⇒

Toolbar Icon	Toolbar Item	Description
	Start	**Starts capturing packets with the same options as the last capture or the default options if none were set**
	Stop	**Stops the currently running capture**
	Restart	**Restarts the current capture session**
	Options	**Opens the "Capture Options" dialog box**
	Open	**Opens the file open dialog box, which allows you to load a capture file for viewing**
	Save As	**Save the current capture file to whatever file you would like**
	Close	**Closes the current capture. If you have not saved the capture, you will be asked to save it first**
	Reload	**Reloads the current capture file**
	Find packet	**Find a packet based on different criteria**
	Go back	**Jump back in the packet history. Hold down the Alt key (Option key on macOS) to go forward in the selection history**

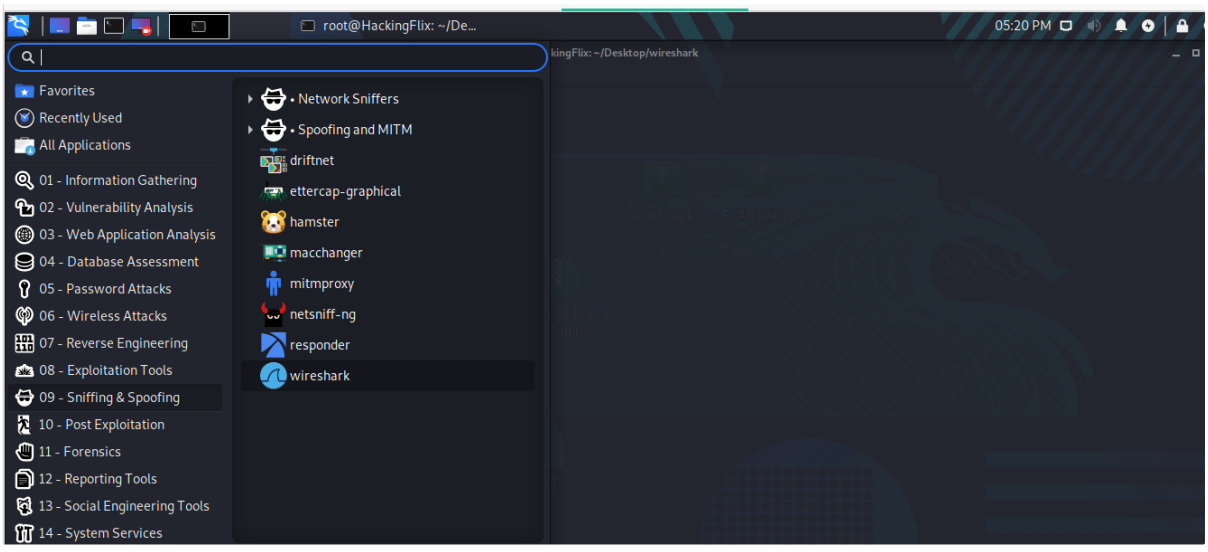
Toolbar Icon	Toolbar Item	Description
	Go Forward	**Jump forward in the packet history. Hold down the Alt key (Option on macOS) to go forward in the selection history.**
	Go To Packet	**Go to a specific packet**
	Go To First Packet	**Jump to the last packet of the capture file**
	Go To Last Packet	**Jump to the last packet of the capture file**
	Auto Scroll in Live capture	**Auto scroll packet list while doing a live capture (or not)**
	Colorize	**Colorize the packet list (or not)**
	Zoom In	**Zoom out of the packet data (increase the font size)**
	Zoom Out	**Zoom out of the packet data (decrease the font size)**
	Normal Size	**Set zoom level back to 100%**
	Resize Columns	**Resize columns, so the content fits into them.**

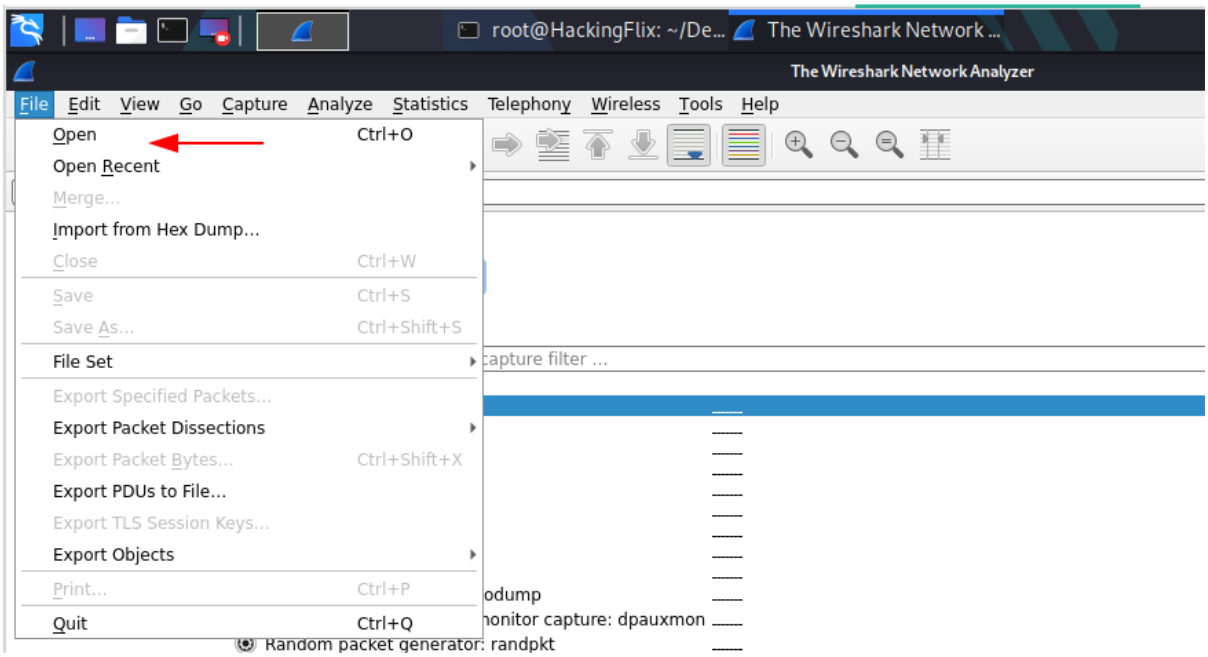
Toolbar Icon	Name	Description
	Bookmarks	Manage or select save filters.
	Filter input	**The area is provided to enter or edit a display string. A syntax check of your filter string while you are typing such as the background will turn red if you enter an incomplete or invalid string and will become green when you enter a valid string.**
	Clear	Reset the current display filter and clear the edit area.
	Apply	Apply the current value in the edit area as the new display filter.
	Recent	Select from a list of recently applied filters.
	Add Button	Add a new filter button.

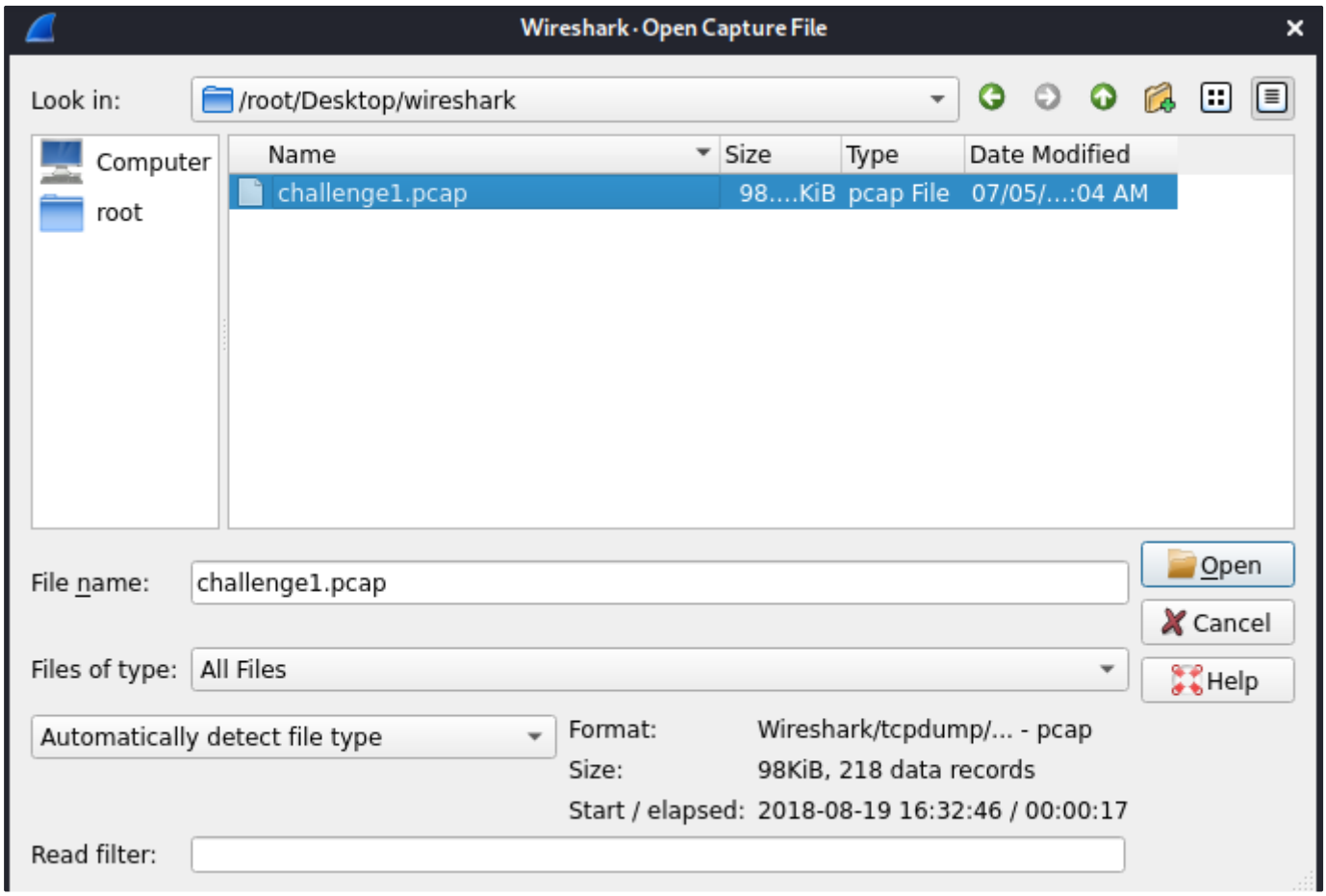
Practical Time

Theory is enough for starting, Now time to learn wireshark practically, from various **pcap files to analyze them and find useful information from there**

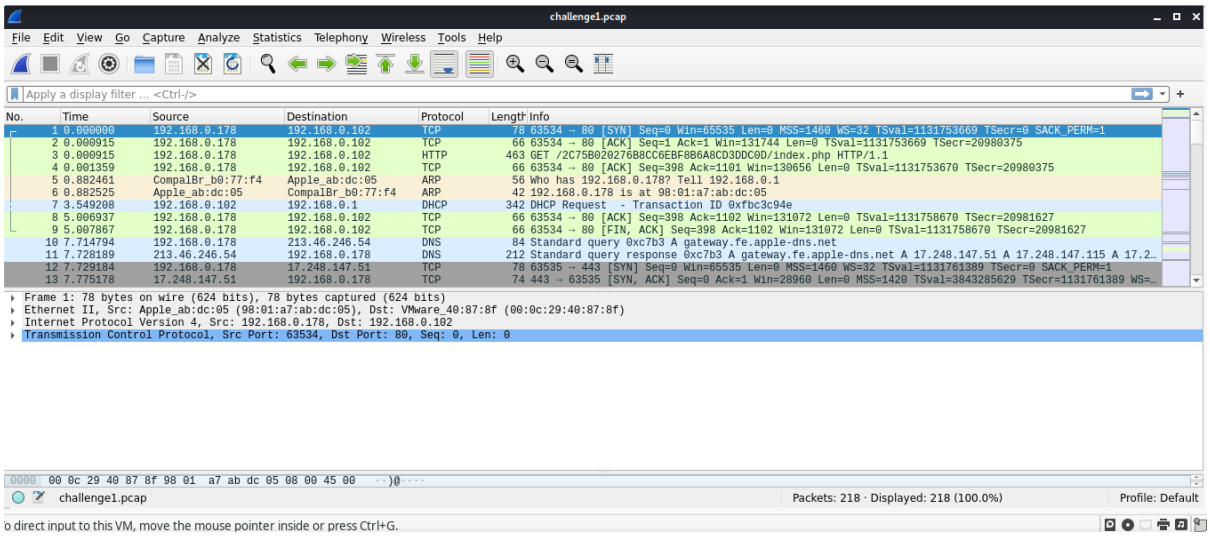
challenge1.pcap







Open **challenge1.pcap** file



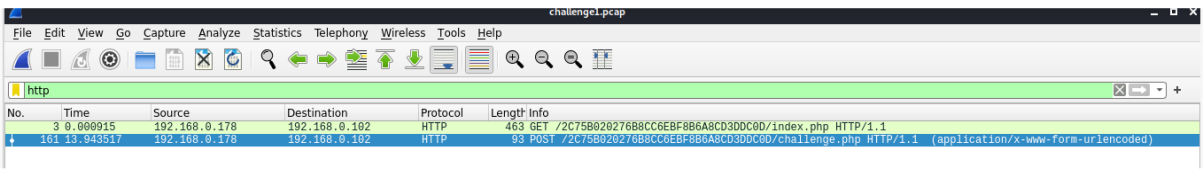
Now, we will do packet analyzes task by task

Task 1

Find out what HOST is using in pcap file ?

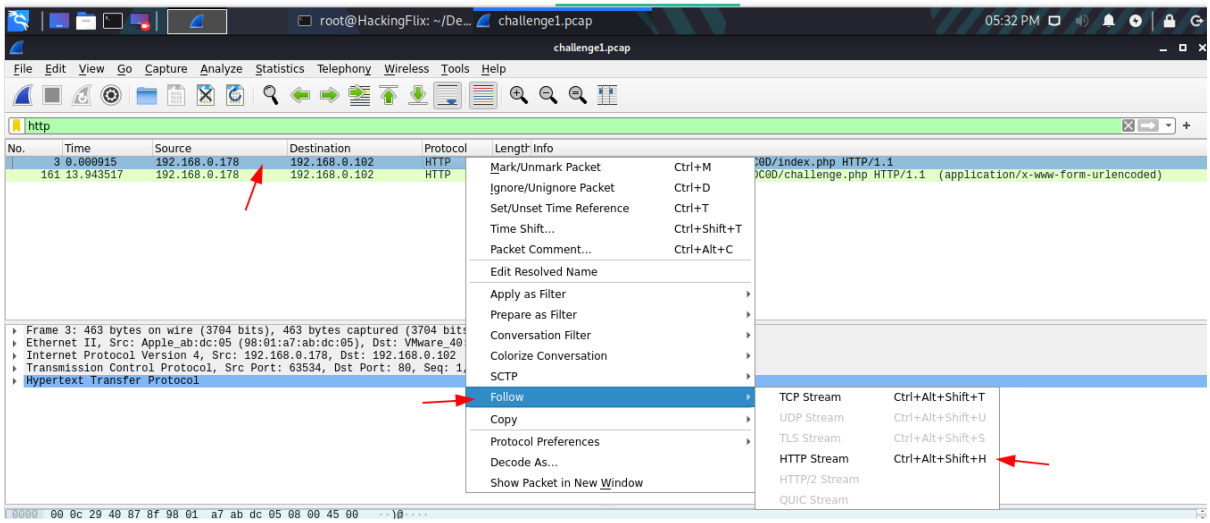
ANS ⇒

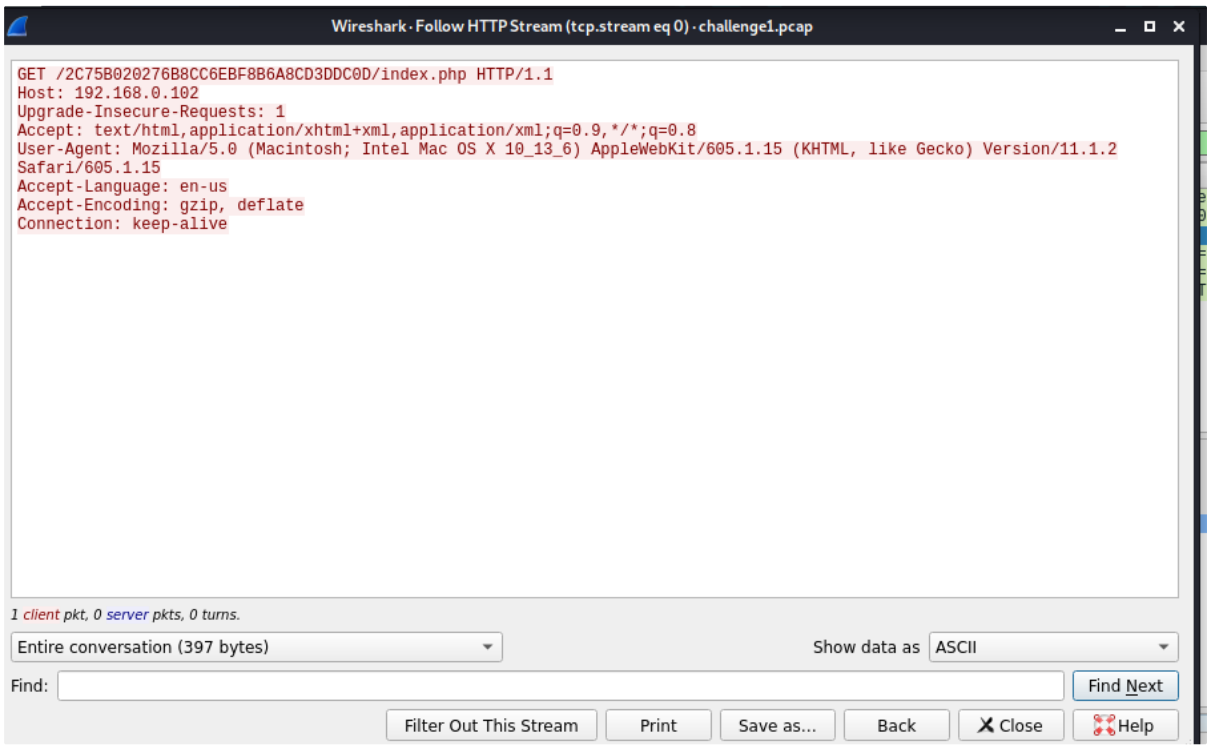
We know **host** is related to **http** part. So in filter type **http**



You will see 2 http results.

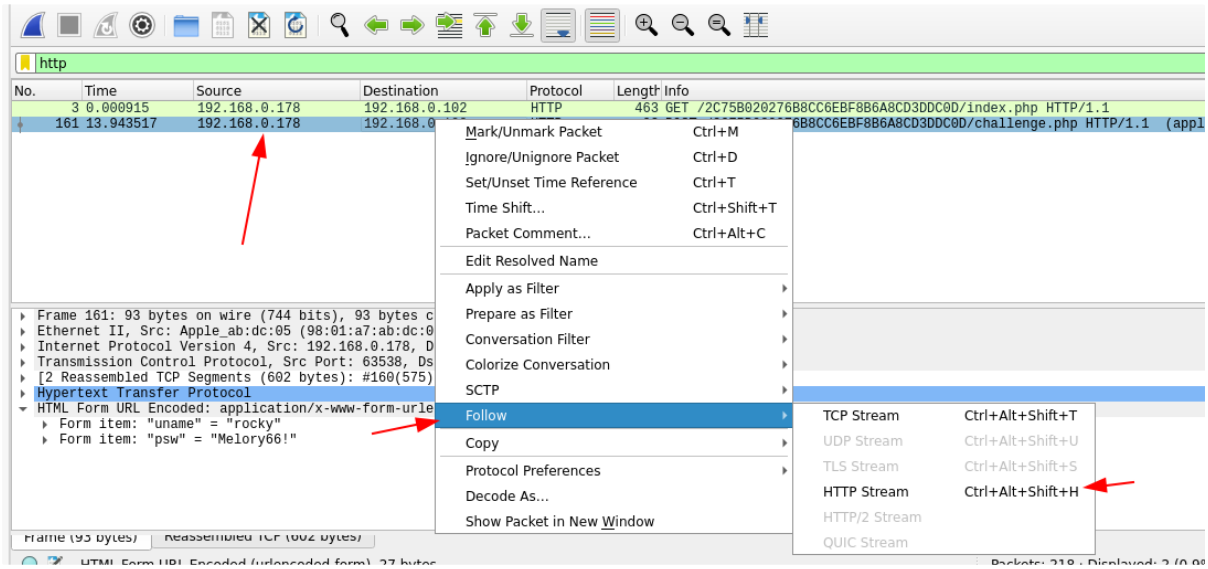
Check 1st one

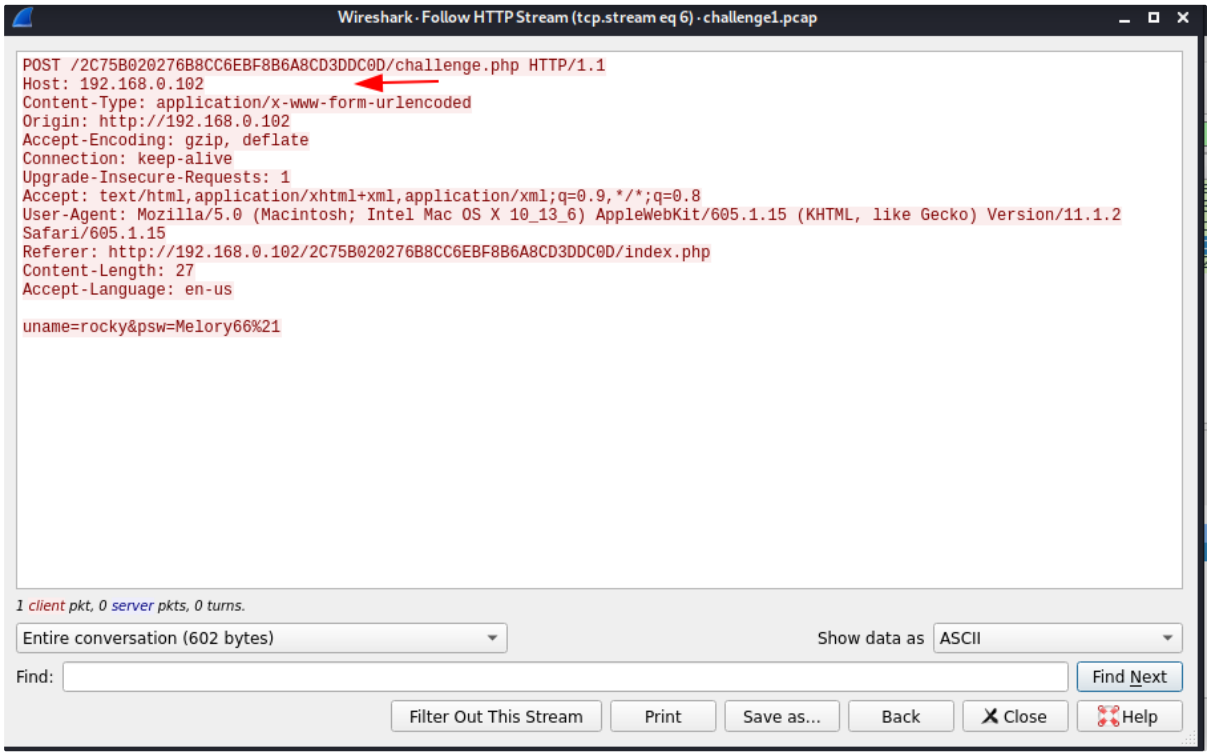




We got **Host: 192.168.0.102**

Now, check 2nd Http result



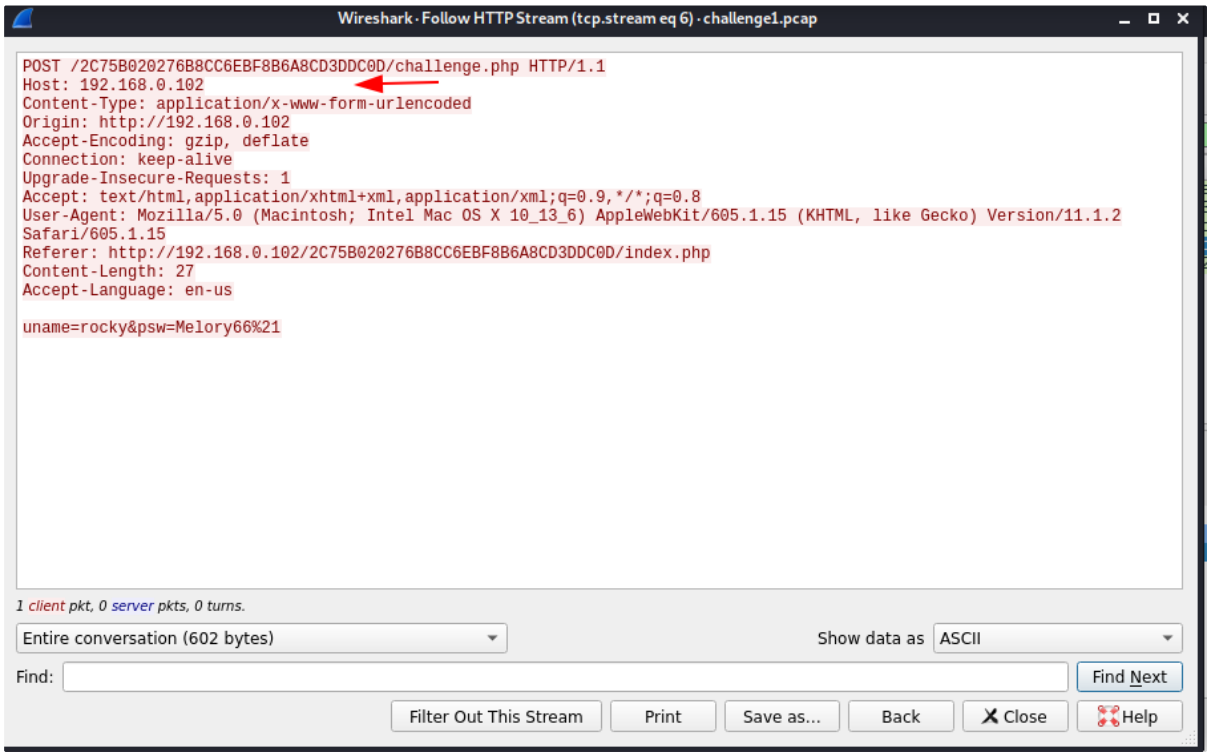


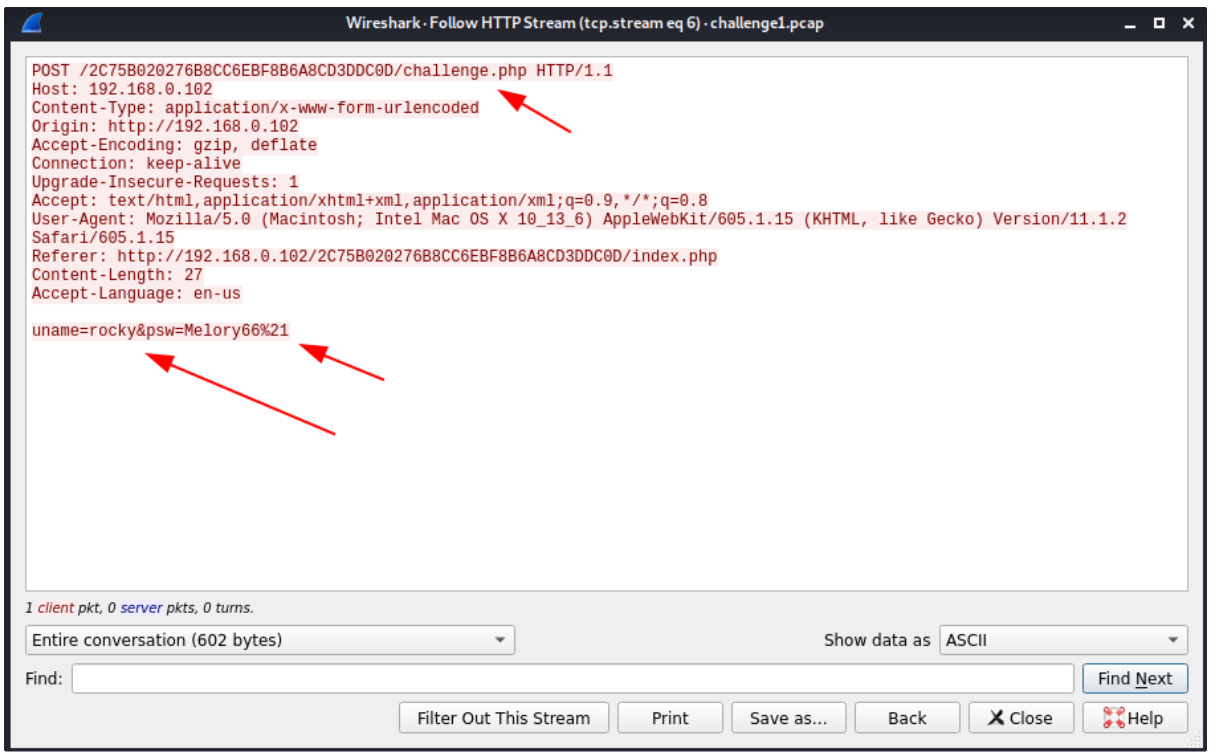
Ok, Now we got our 2nd Host ⇒ **Host: 192.168.0.102**

We, can see that we have got traffic from same HOST

Task 2

What other information we can get from same challenge pcap file about Host 192.168.0.102

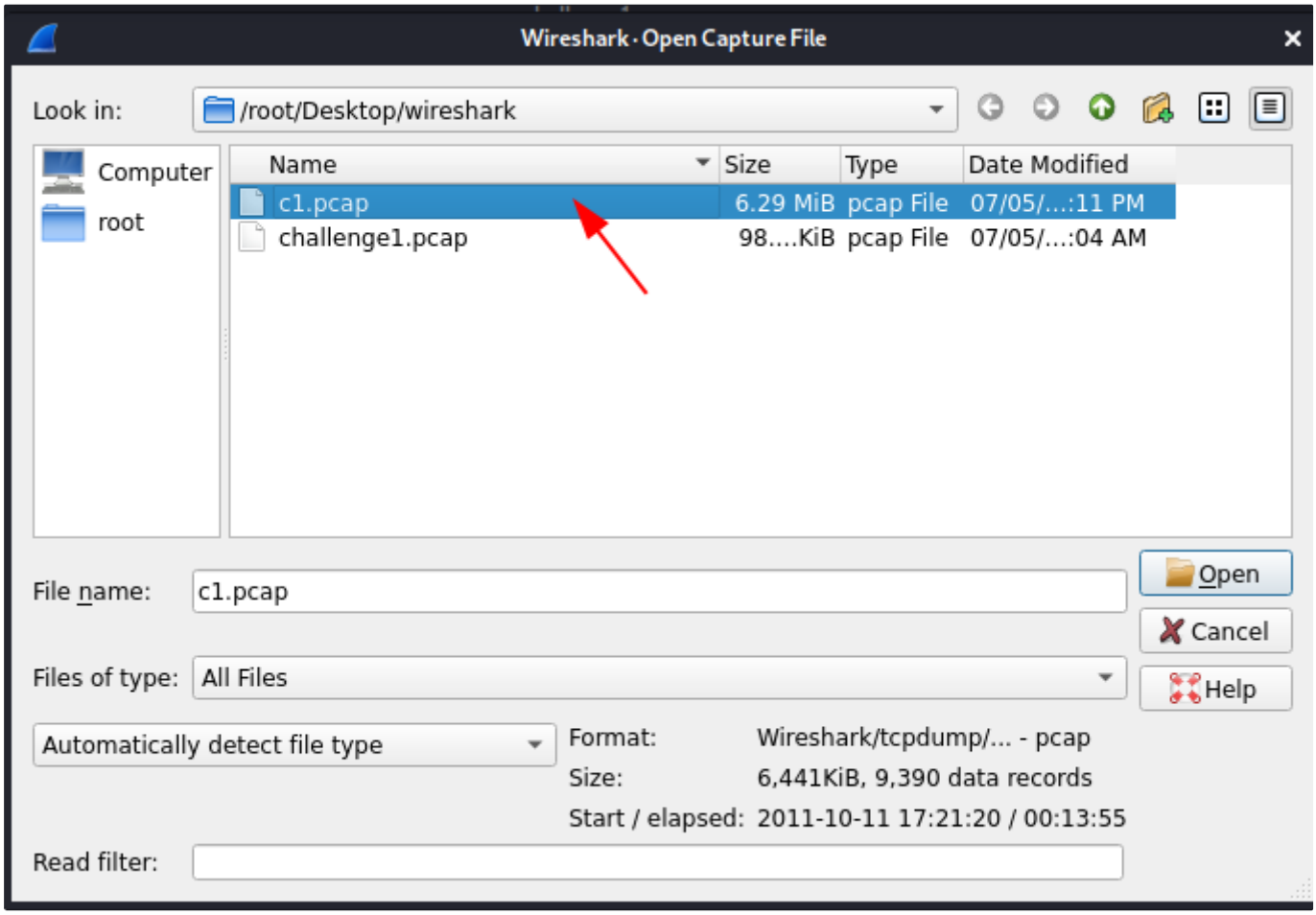


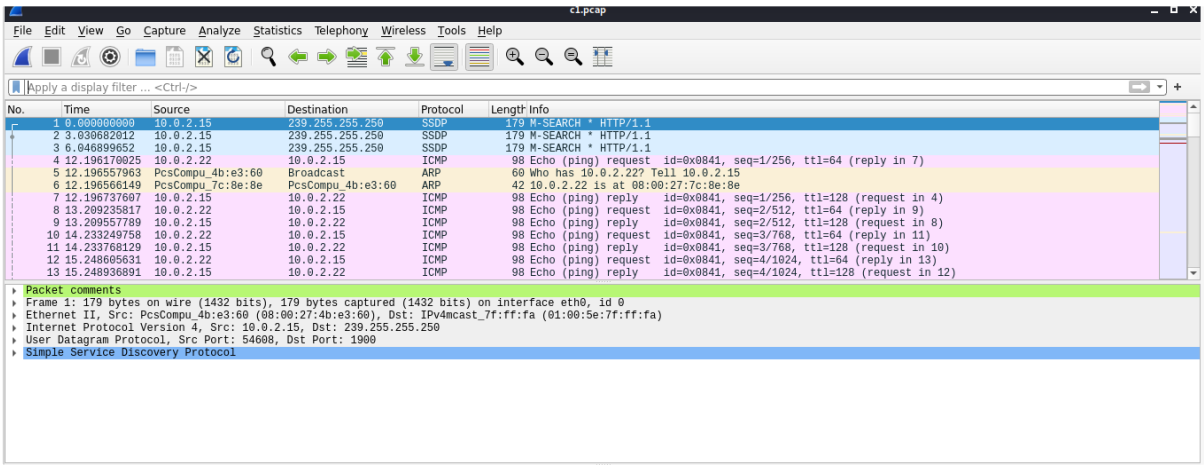


We, can see there is POST Request and username and password is in clear text mean no encoding on challenge.php file

Now, we will use another **pcap file and do more tasks**

c1.pcap





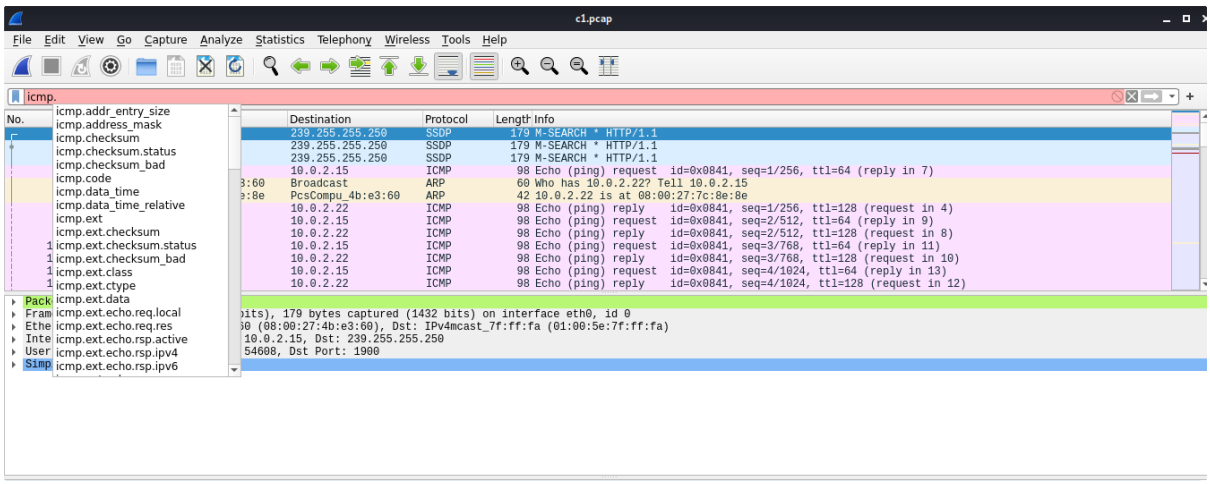
Task 1

How many ping requests were sent in the c1.pcap capture?

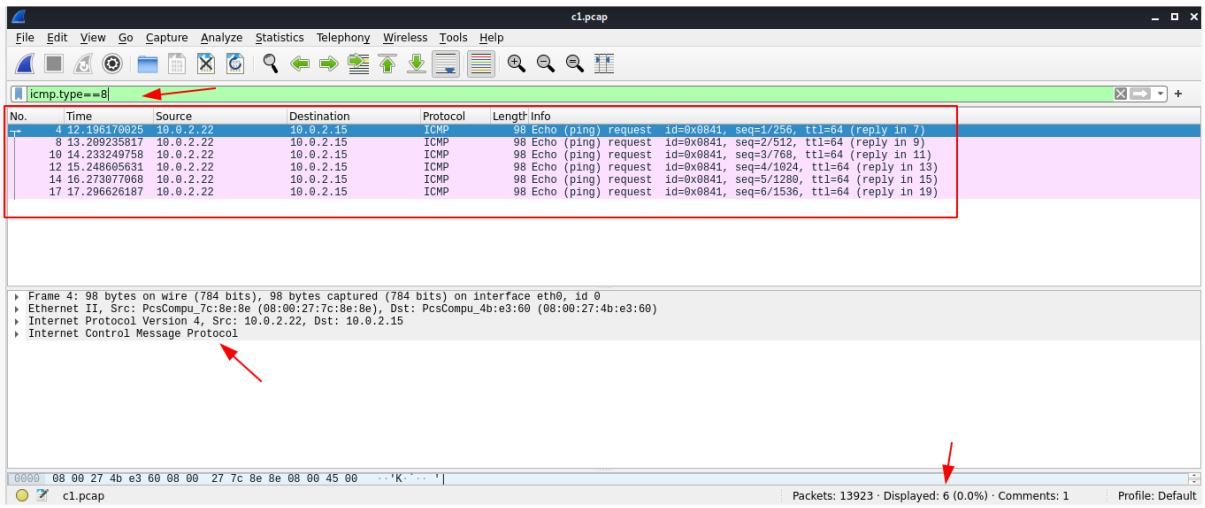
Ans :

Open the packet capture and apply the following filter: **icmp.type==8**

This filter will show all ICMP (aka ping) packets that are Type 8, which is an echo ping request



Above pic, in this you will see there are many options to use, so, this will let you know what you need to filter for specific result. For now let's use **icmp.type==8**



So, you see there are total 6 icmp packets so we got our ans

So, answer is there are 6 **ping requests**

Task 2

What is the IP address of the device associated with **08:00:27:4b:e3:60?**

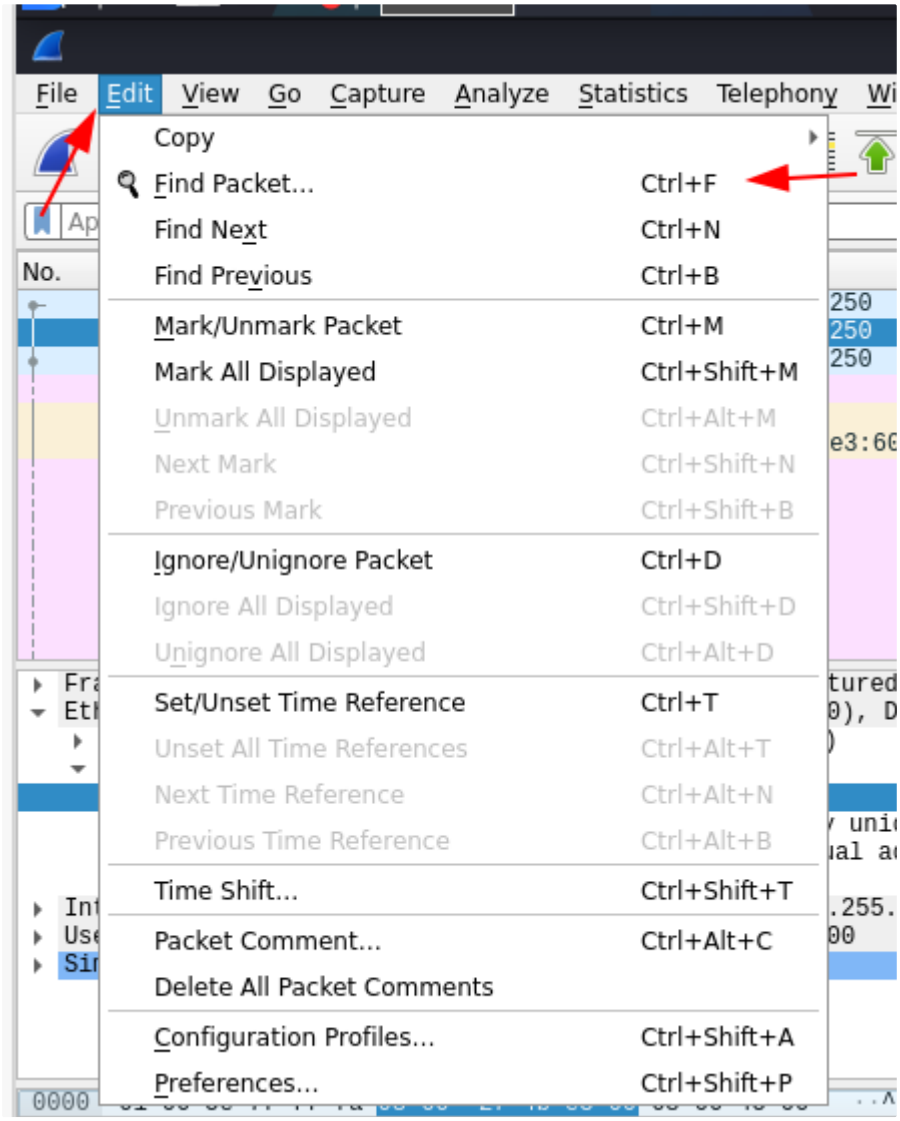
Ok, now we need to find IP for mac address **08:00:27:4b:e3:60**

So, to find out IP , we need to find hexadecimal value **for MAC Address**

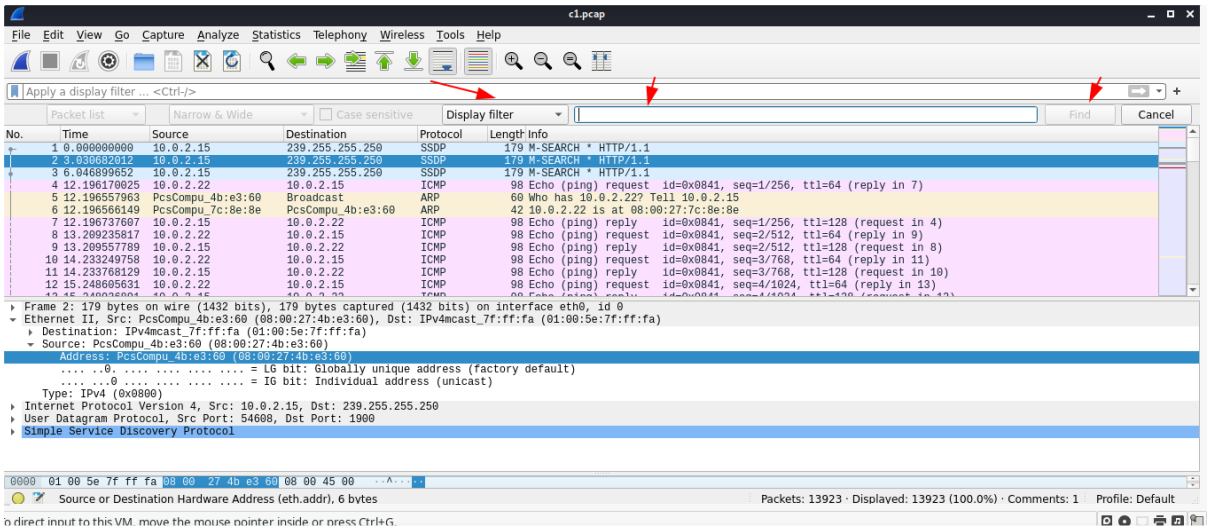
This ⇒ **08:00:27:4b:e3:60** ⇒ is a **six-byte hexadecimal value separated by colon (:)**

How to do this ?

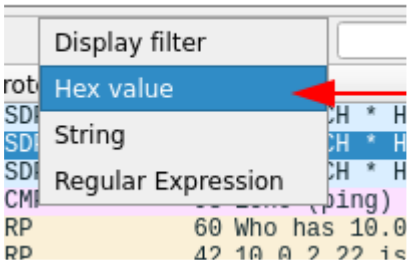
First ⇒



Second ⇒

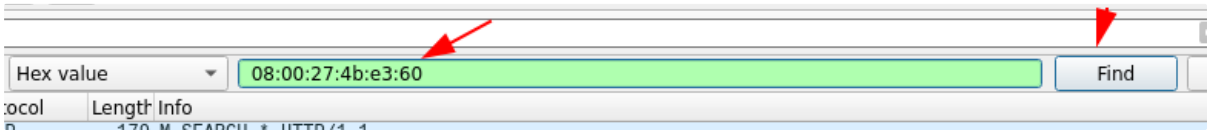


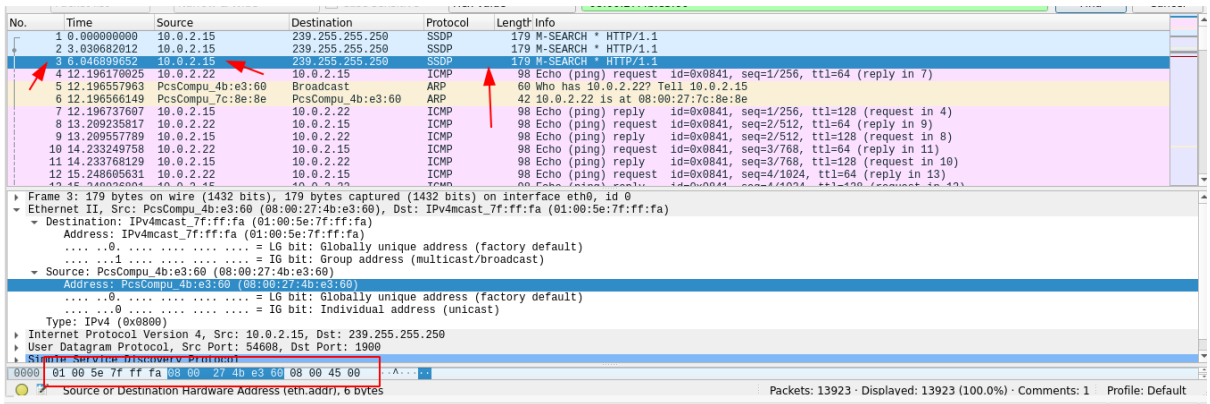
Change Display Filter to **Hex value**



Third ⇒

Now enter **mac address and click on find**





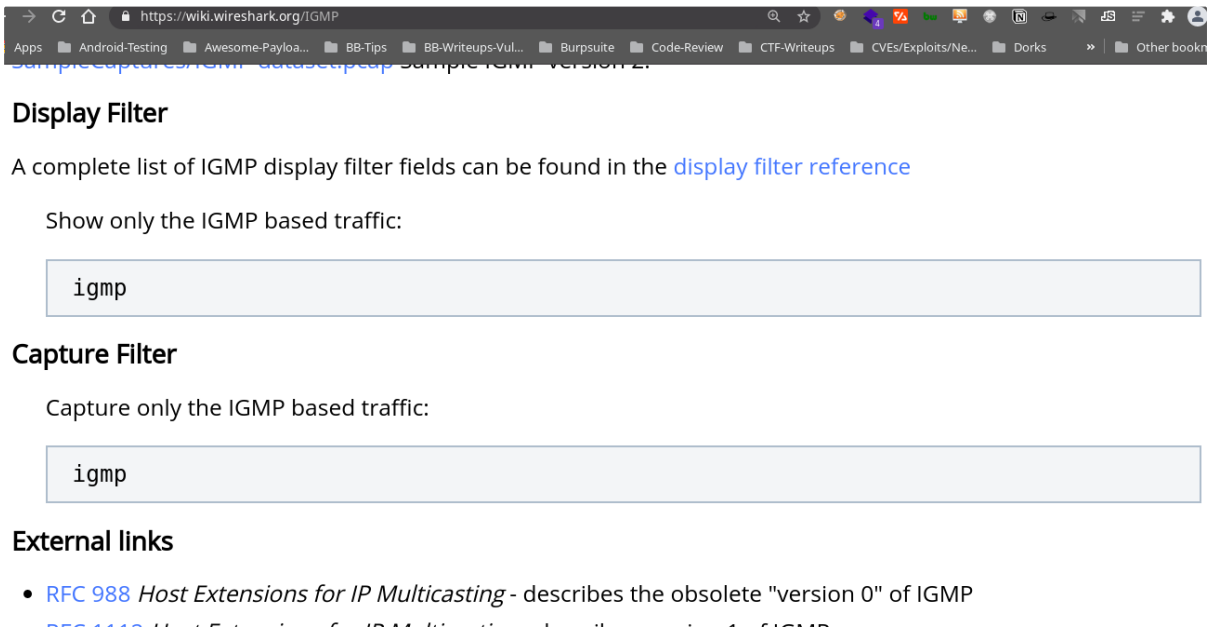
So, we got our IP Address ⇒ **10.0.2.15**

Task 3

What version of **Internet Group Management Protocol** is in use?

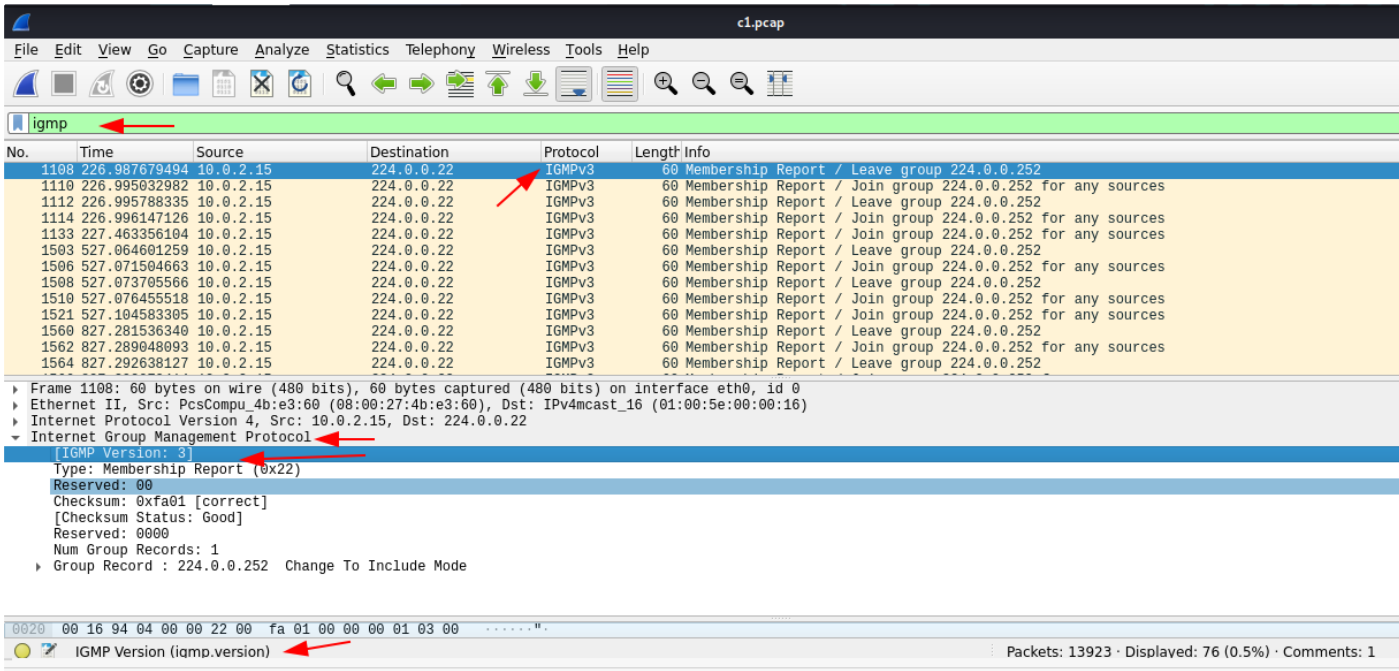
ANS :

Google IGMP



Whenever you in doubt, just google like above

Now ⇒ go to filter and type **igmp** ⇒ and in packet detail pane + packet list pane check for IGMP Version



So, IGMP Version is ⇒ **3**

Task 4

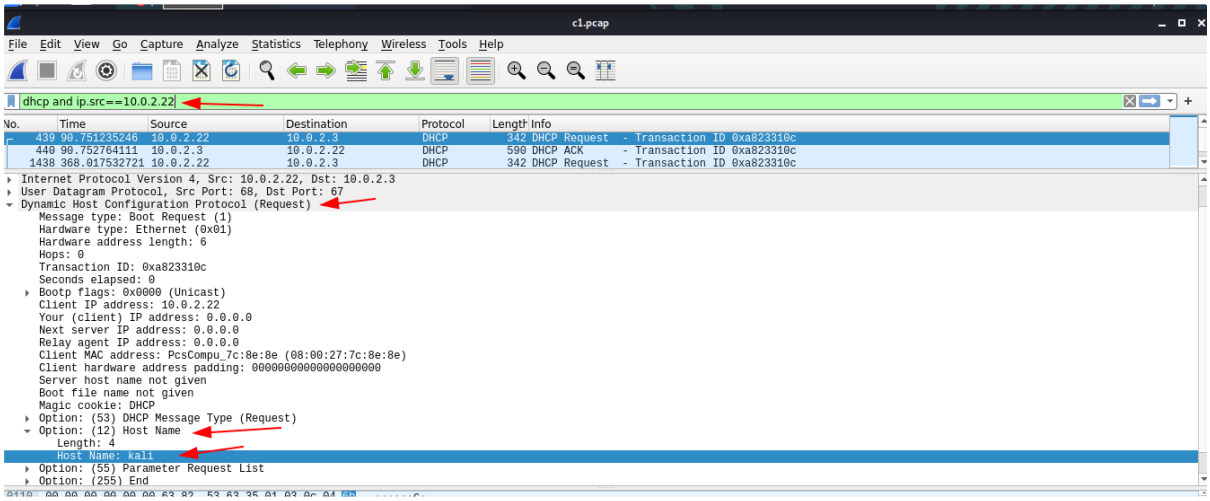
What is the name of the host located at 10.0.2.22?

ANS :

Host names can be identified in dynamic host configuration protocol (DHCP) traffic

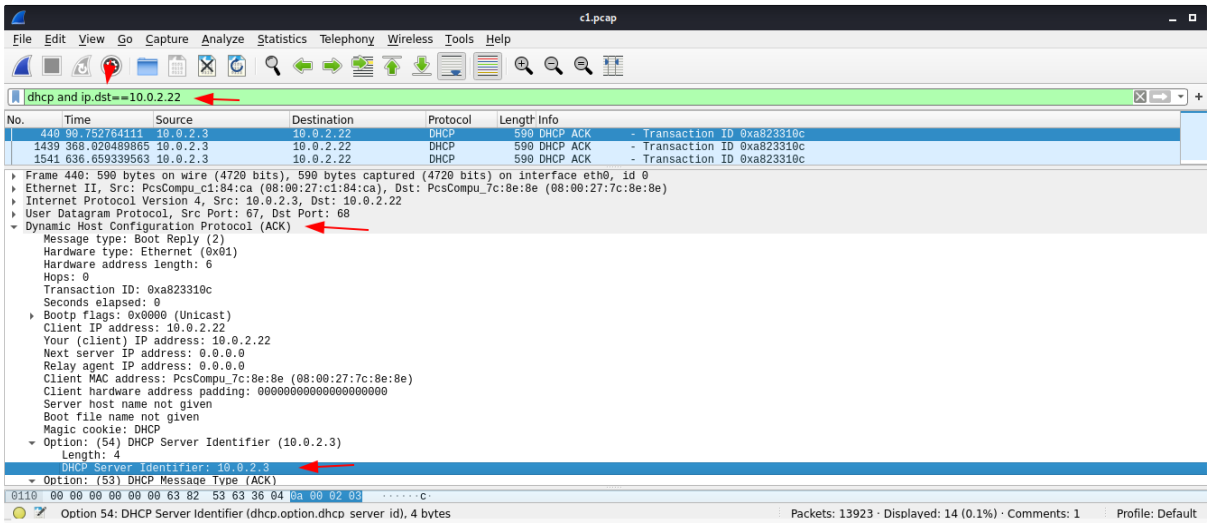
Using a filter ⇒ **dhcp and ip.src == 10.0.2.22**

where **ip.src is source of IP**



So, we got our hostname ⇒ **Kali**

Suppose now, we want to know **IP Address of DHCP Server , then now change ip.src to ip.dst**



and we will get out IP ⇒ **10.0.2.3**

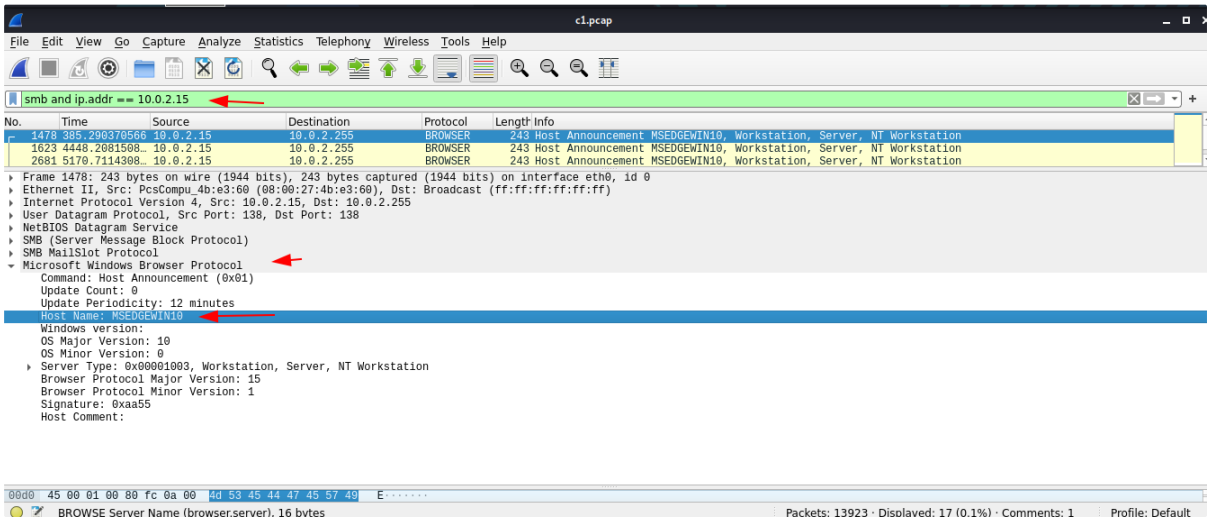
Task 5

What is the name of the host located at 10.0.2.15?

ANS:

In this case we are going to look for server message block (SMB) protocol activity, which features host announcements

Filter for this ⇒ **smb and ip.addr == 10.0.2.15**



So, we got host ⇒ **MSEDGEWIN10**

Task 6

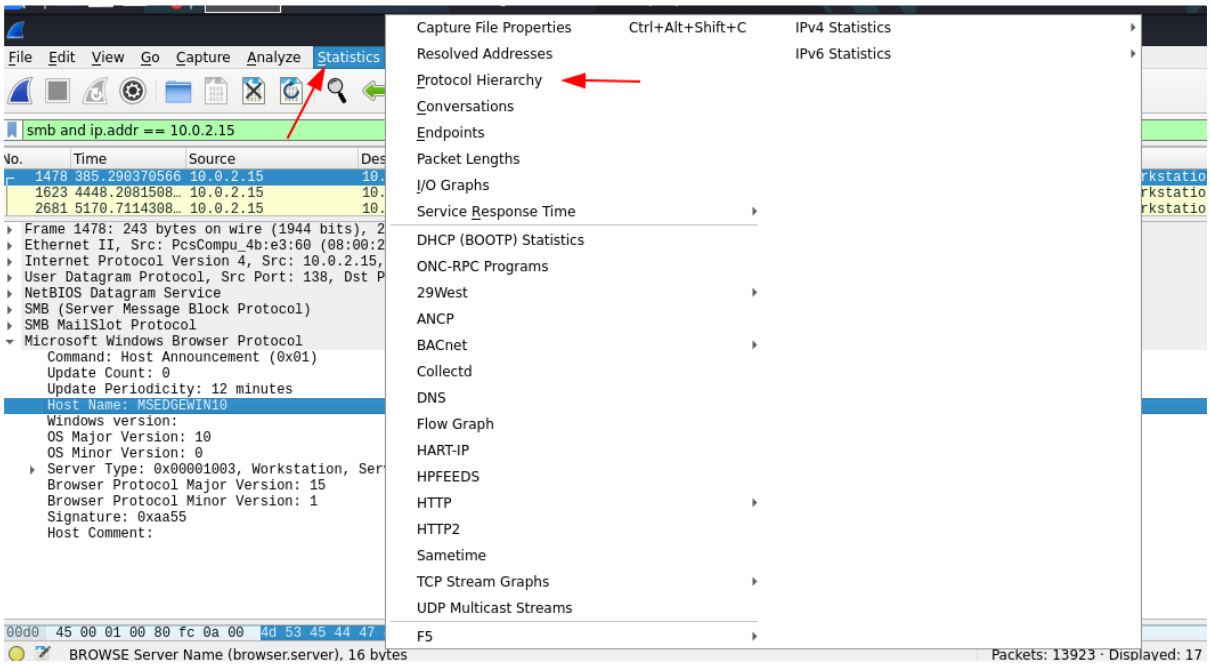
What is the IP address of the attacker?

Now, suppose you want to know the IP of attacker, and for this we need to check those IP Address which give lots of request to our server

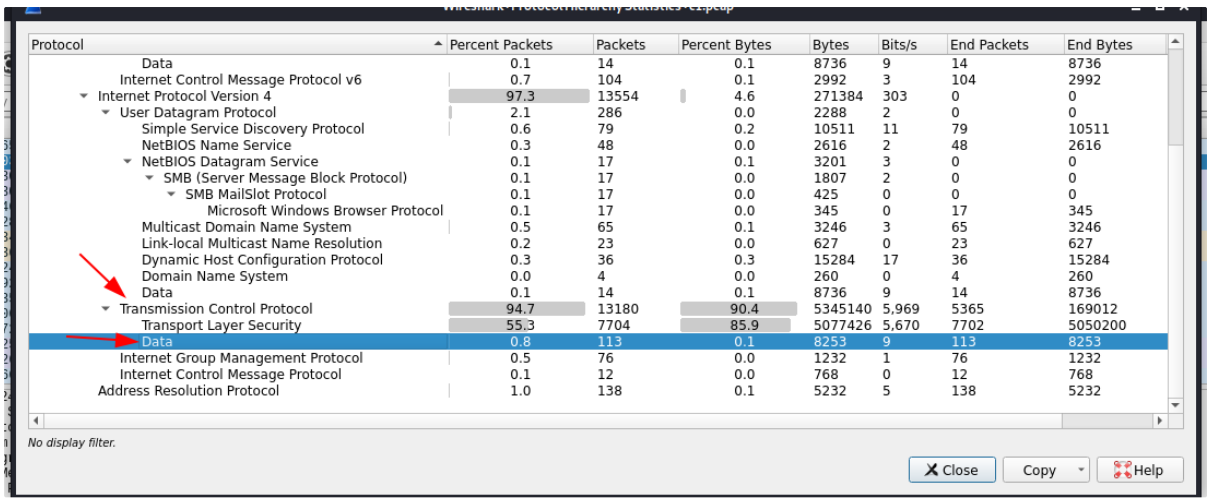
So, we need to identify the attack activity in the capture.

Let's do this

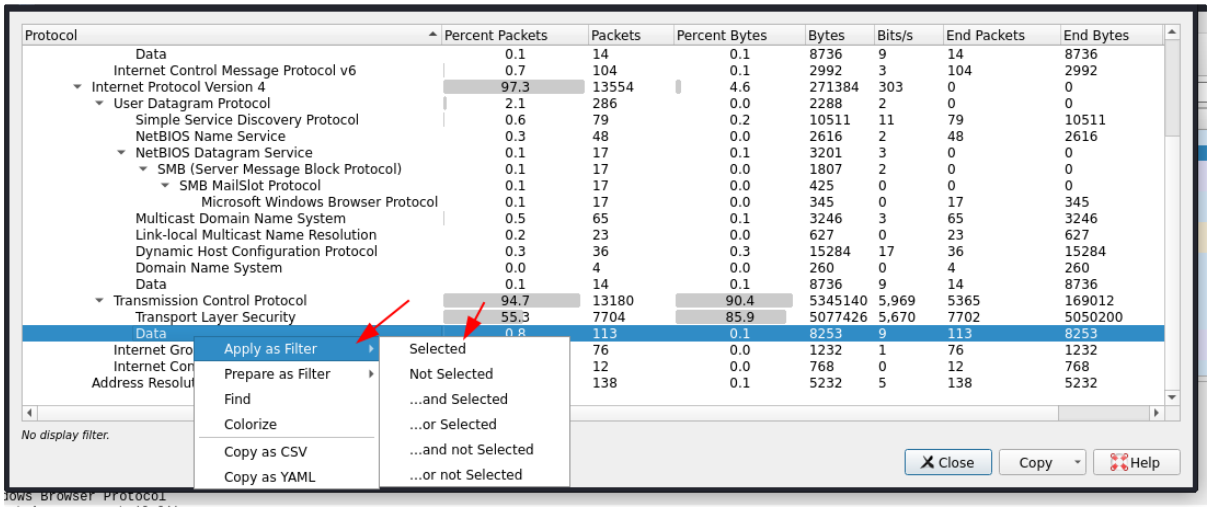
Step 1 ⇒



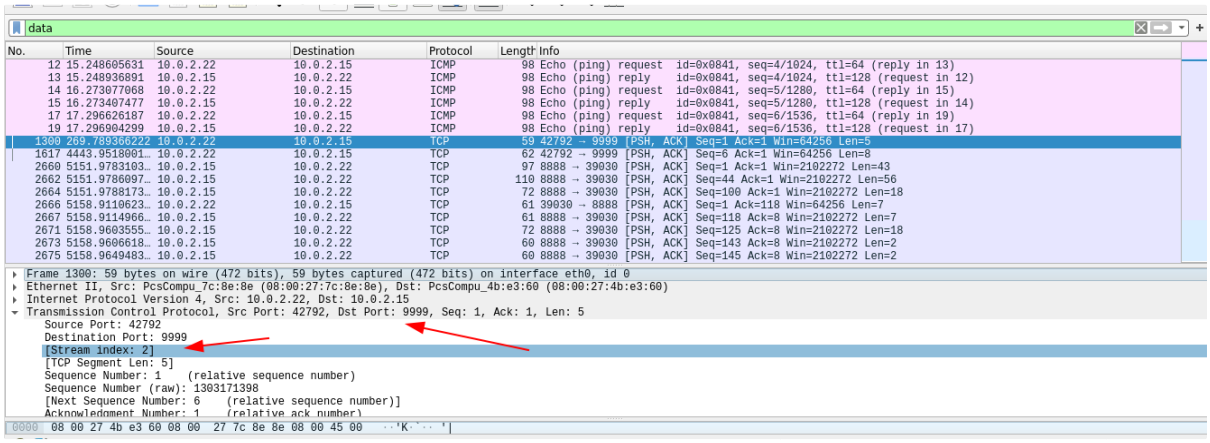
Step 2 ⇒

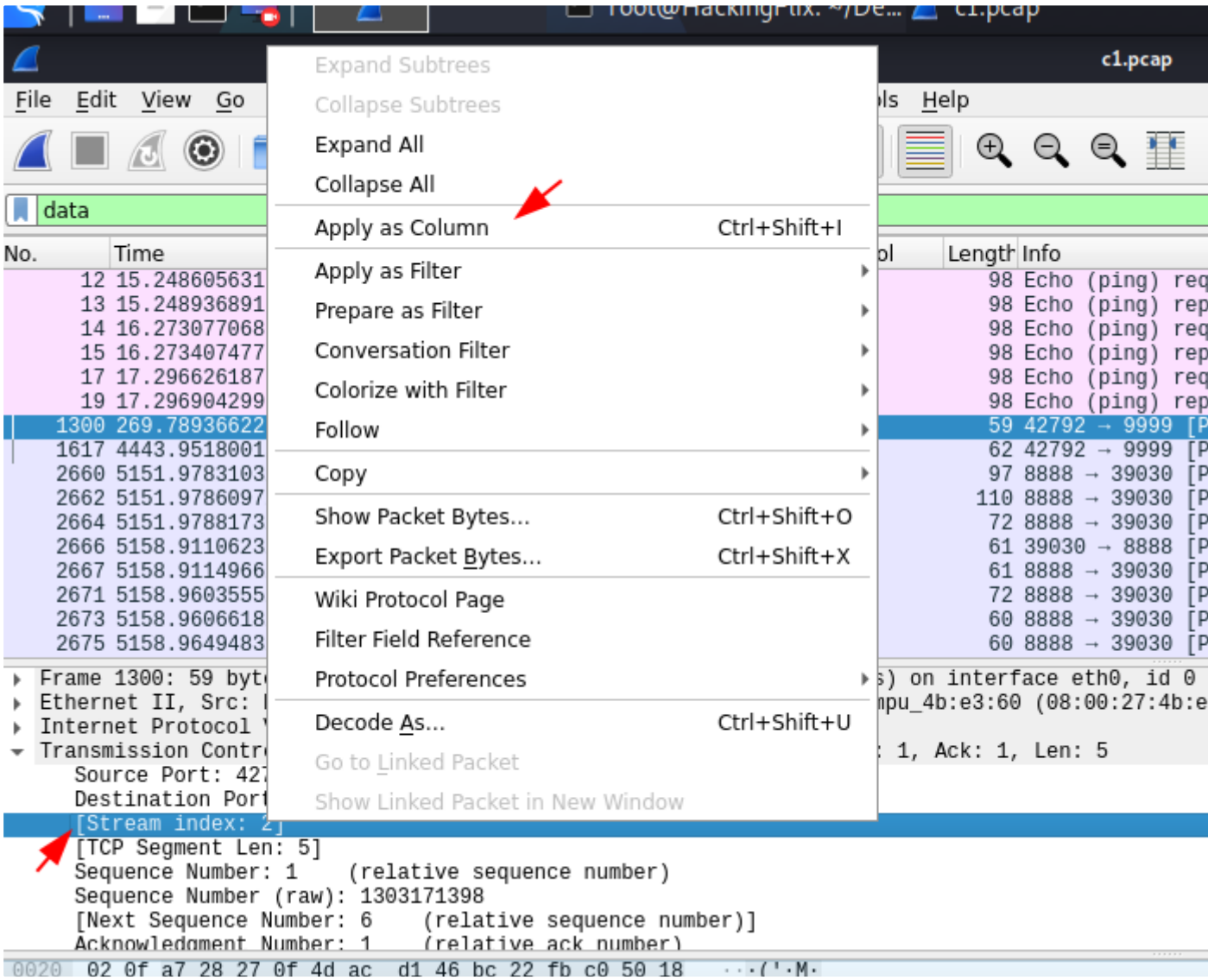


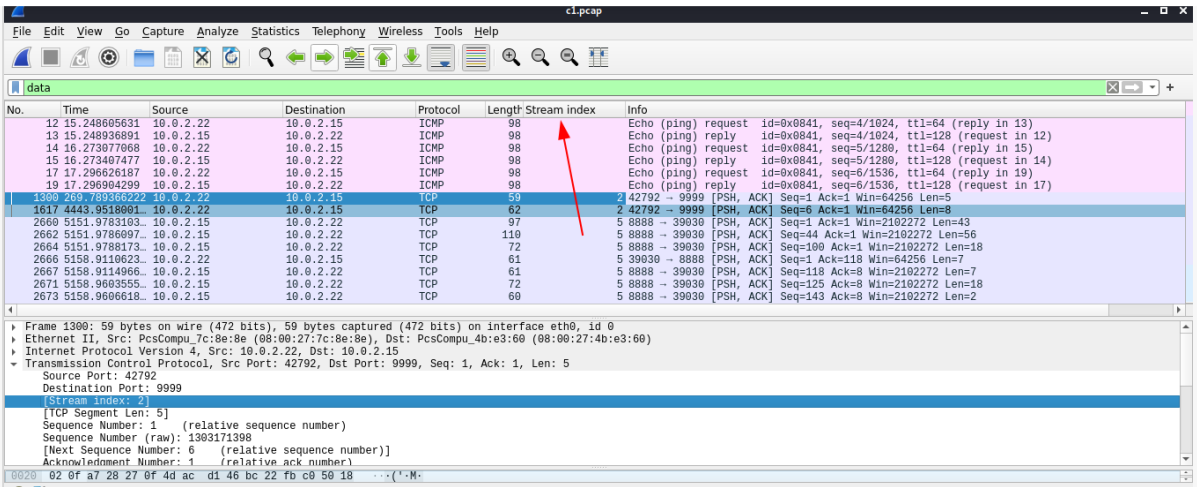
Step 3 ⇒



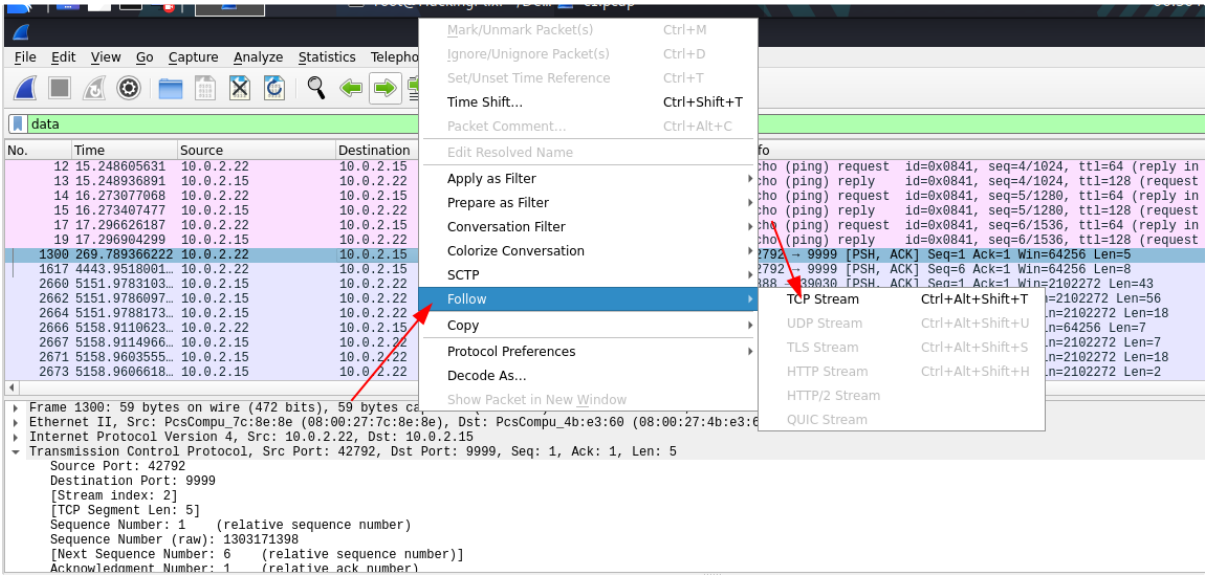
Step 4 ⇒

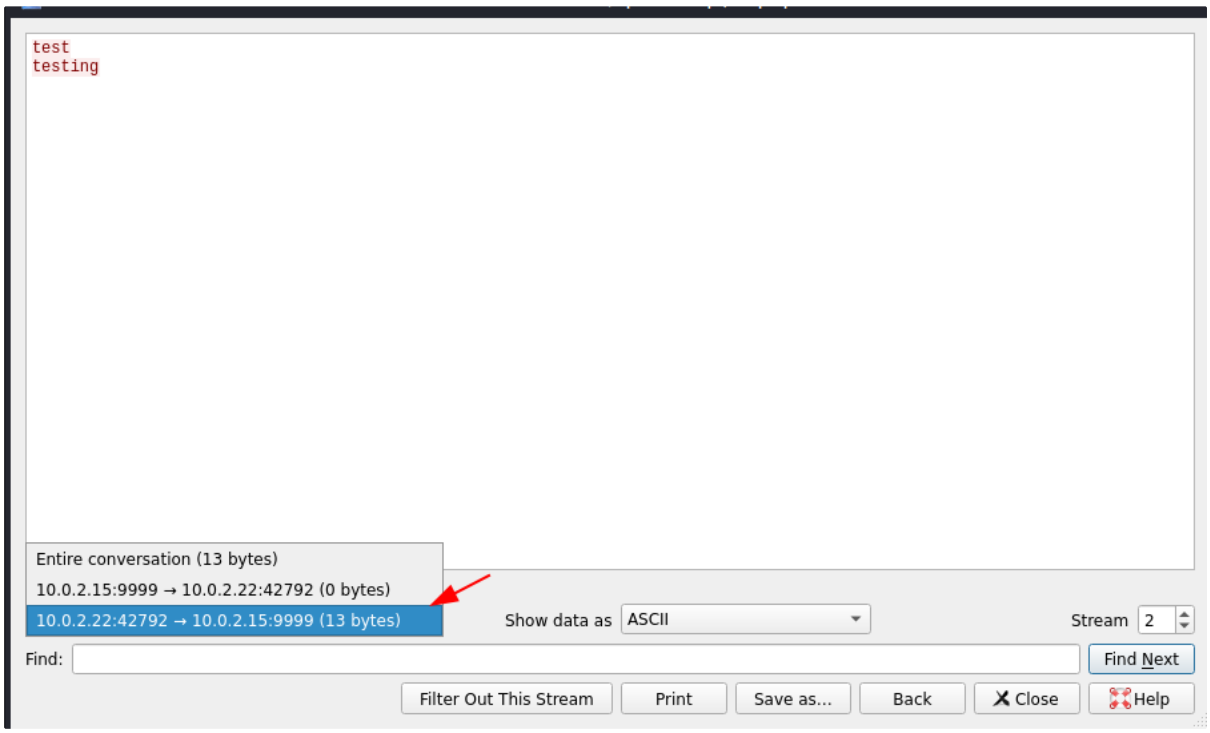






Step 5 ⇒

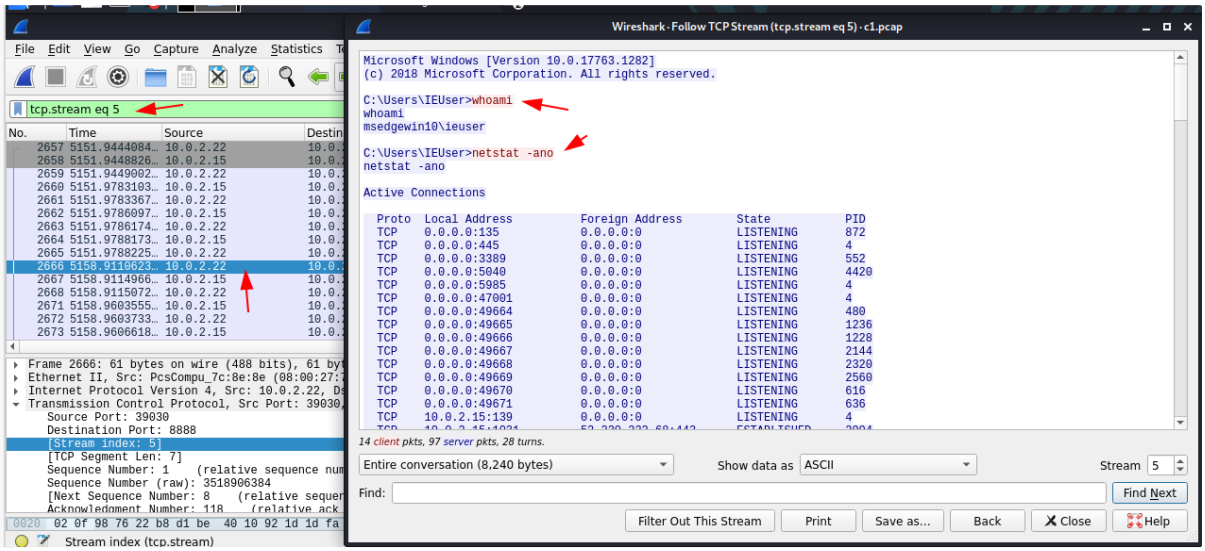




So, we got our IP ⇒ **10.0.2.22** ⇒ this is the likely attacker IP and the threat actor appears to have remote shell access to the victim box.

Now, if you want to know what actually attacker did ? or what command he used ? then ⇒

Just follow any tcp stream index 5 and right click ⇒ follow ⇒ tcp stream and we see ⇒ **whoami** command is run by attacker



So, we have learn many new things in this challenge through tasks. like how to use filter for specific task

---

BONUS

Let's us see lists of important Wireshark Filters

English	C-like	Description and example
eq	==	Equal. ip.src==10.0.0.5
ne	!=	Not equal. ip.src!=10.0.0.5
gt	>	Greater than. frame.len > 10
lt	<	Less than. frame.len < 128
ge	>=	Greater than or equal to. frame.len ge 0x100
le	<=	Less than or equal to. frame.len ⇐ 0x20
contains		Protocol, field or slice contains a value. sip.To contains "a1762"
matches	~	Protocol or text field match Perl regualar expression. http.host matches "acme\.(org|com|net)"
bitwise_and	&	Compare bit field value. tcp.flags & 0x02
and	&&	Logical AND. ip.src==10.0.0.5 and tcp.flags.fin
or	||	Logical OR. ip.scr==10.0.0.5 or ip.src==192.1.1.1
xor	^^	Logical XOR. tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29
not	!	Logical NOT. not llc
[...]		Slice Operator. eth.addr[0:3]==00:06:5B
in		Membership Operator. tcp.port in {80 443 8080}

Examples ⇒

- tcp.port eq 80 ⇒
  - filter only that tcp protocol whose port is **80**
- tcp.srcport===443 ⇒
  - filter only that tcp whose source port is **443**
- List

Filter for HTTP and HTTPS traffic:

```
tcp.port==443 or tcp.port==80
ssl or http
tcp.port in {80 443 8080}
tcp.port == 80 || tcp.port == 443 || tcp.port == 8080
```

- List

Filter for a protocol:
tcp
udp
dns

- Filter for IP Addresses ⇒
  - ip.addr == 10.43.54.65
  - ! (ip.addr == 10.43.54.65

You will have doubt what actually is **src and dst ?**

> **src** ⇒ Source - where the packet came from
>
> **dst ⇒** Destination - where the packet is going