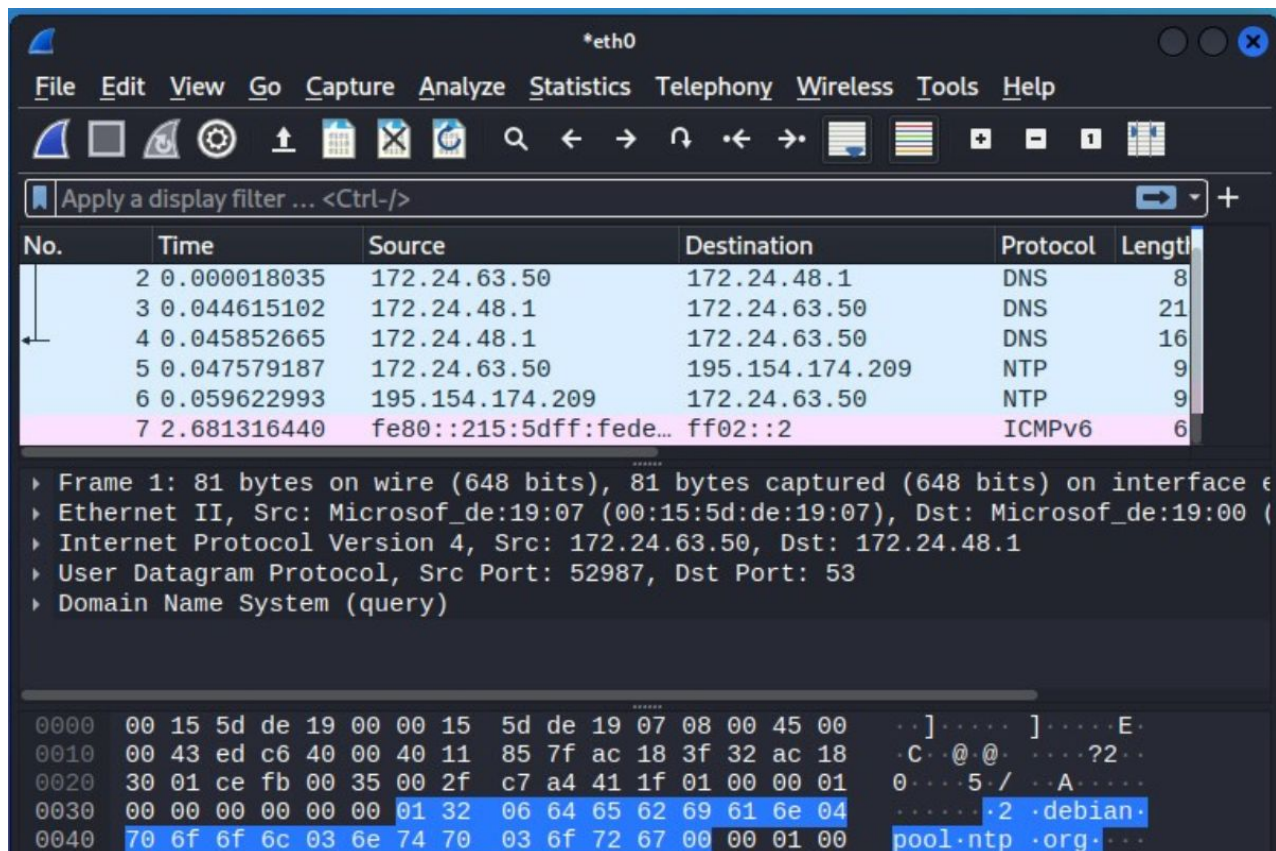


# How To Install & Use Wireshark On Kali Linux

 infosecscout.com/wireshark-on-kali-linux

Patrick Fromaget



Wireshark is a free and open-source tool to capture and analyze network traffic. Basically, it will intercept network packets and display their content in a nice interface, so you can analyze them. It's available on most operating systems. I will show you how to use it on Kali Linux, and share interesting features for you to use.

**Wireshark is installed by default on Kali Linux, and can be used directly after installation. It's one of the most important tool included in this distribution, used by hackers and pen testers to analyze network traffic.**

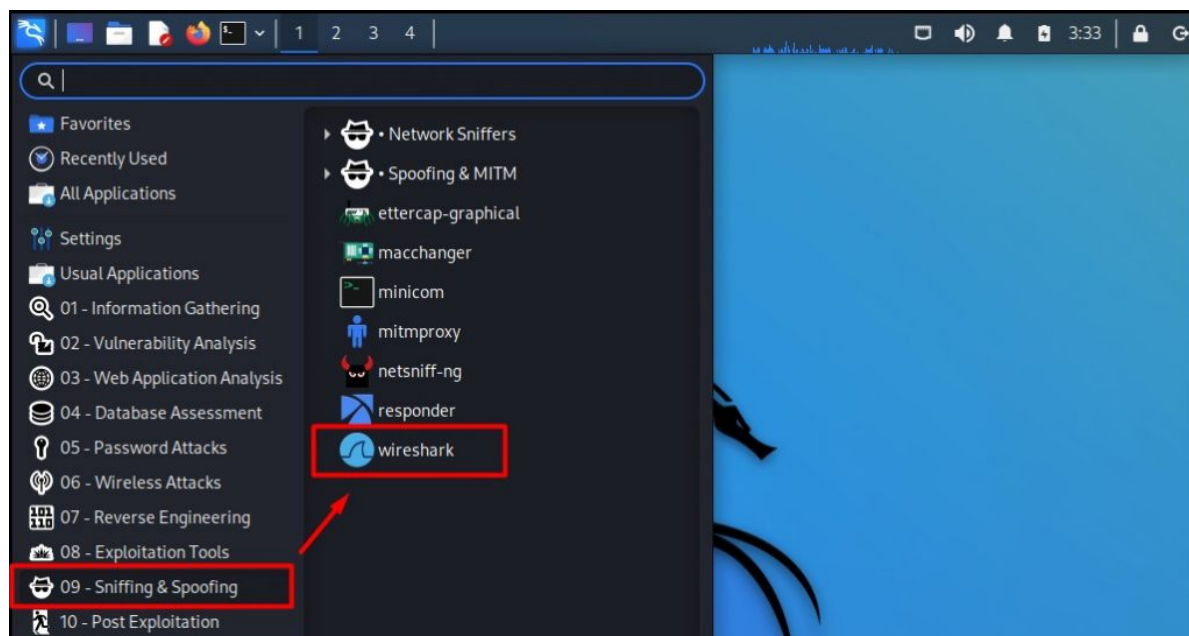
But once installed, the first steps might be a bit confusing if you never used it before. So keep reading for the full installation procedure, and an introduction to some of the most powerful features.

## How To Install Wireshark on Kali Linux

**Wireshark is included by default on Kali Linux, whatever version you are using, it's even pre-installed on the Live system, so there is no need to install anything to use Wireshark.**

As Kali Linux comes with a lot of tools included, you may have a hard time to find Wireshark in the main menu, here is how to start the application:

- Open the main menu.
- Go to Sniffing & Spoofing (number 9 in the current release).



- Find Wireshark at the end of the list and click on it.

You can also use the search engine in the main menu to find it faster, or create a shortcut somewhere to have it at hand all the time.

### Hide your IP address and location with a free VPN:

[Try it for free now](#), with advanced security features.

2900+ servers in 65 countries. It's free. Forever.

If, for any reason, Wireshark is not available in the main menu, you can always install it manually. Wireshark is available in the default repositories, so you can open a terminal and type:

```
sudo apt update
```

```
sudo apt install wireshark
```

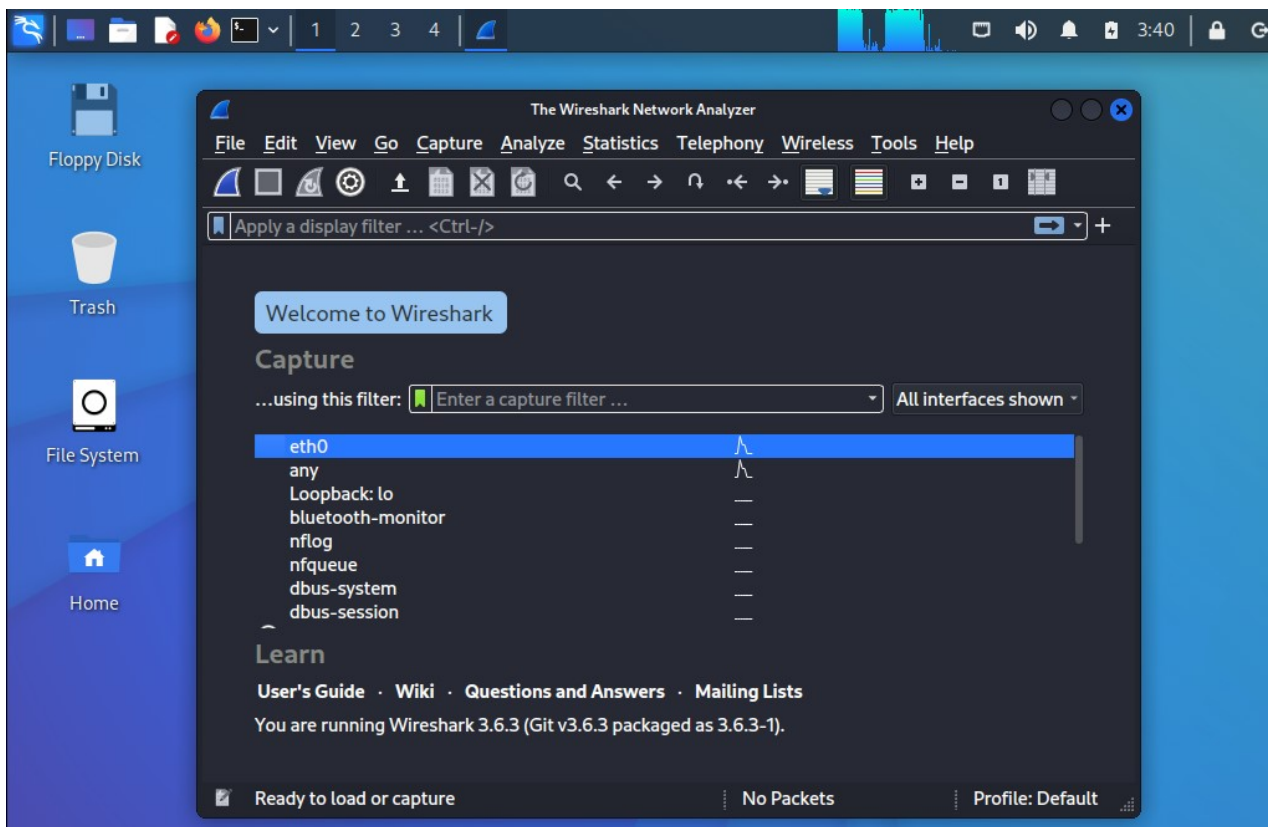
To get it back on your system

## How To Use Wireshark on Kali Linux

Installing Wireshark is pretty straightforward. If you haven't used this tool, the difficulty is probably starting now to understand exactly what it does and how to use it. Let's browse the most important features together.

### First steps with Wireshark

When you start Wireshark for the first time, the interface looks like this:



You'll find the main actions in the shortcut bar at the top of the screen. You also need to pick a network interface to listen to (in general, it will be eth0). And you have the full menu for all the ninja features included in this tool.

I won't explain everything in this article, but I absolutely want to explain how to capture the network traffic and analyzer the results, so let's get right to it.

## Capture network traffic

The main feature that you'll use frequently with Wireshark is the capture. Basically, the idea is to listen what's happening on one of your network interfaces. If your computer is just one element of your network, it will mostly be your own network usage, and a few talks between your device and the other ones.

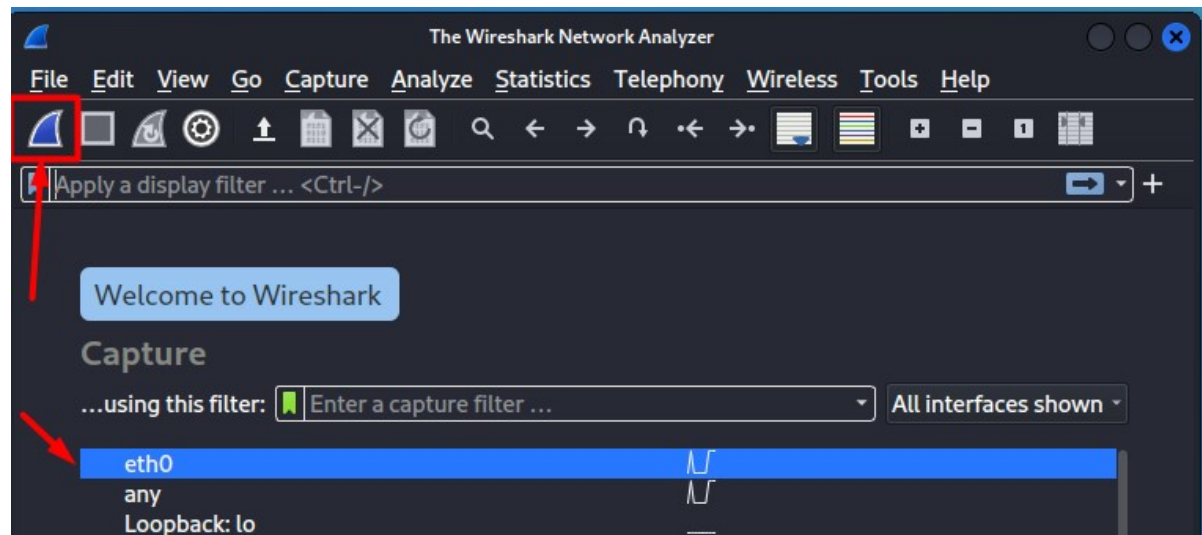
**But when your device is an important node of this network (DNS server, gateway, etc.), it will record almost anything happening on the network. This will be pretty useful for the analysis part I'll introduce later (and it's also used by hackers and pen-testers).**

Anyway, here is how to start a capture with Wireshark:

- **Select the interface you want to capture in the list.**

In general, it will be "eth0" if your computer is plugged via Ethernet, or "wlan0" if you are using a Wi-Fi connection.

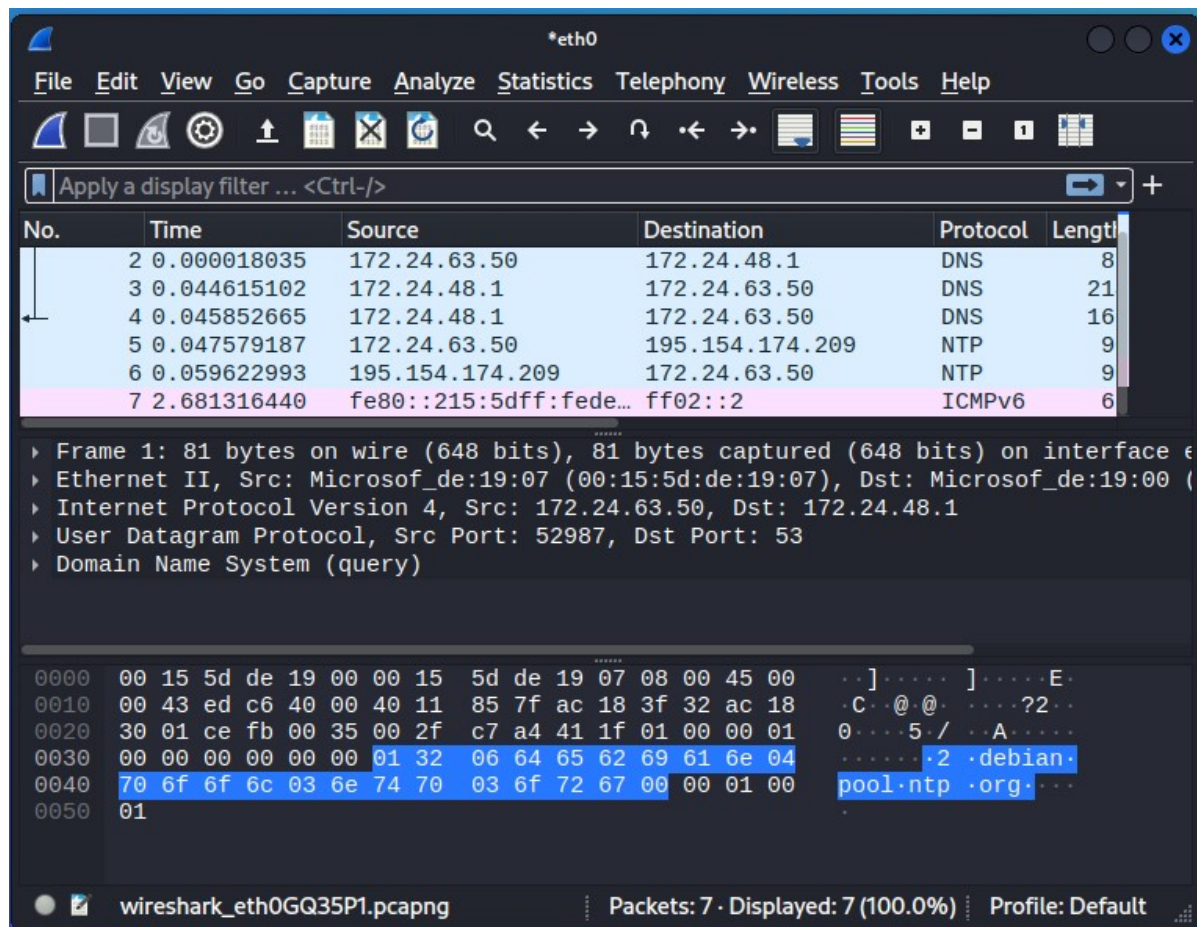
- Click on the first icon in the top bar.



You can also double-click on the interface name on the home page, use the capture menu, or just press CTRL+E.



- If everything is working properly, the window will start to be filled with a table refreshing constantly:

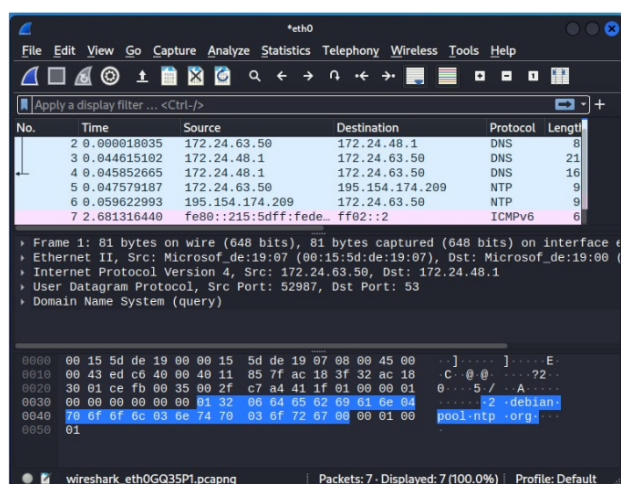


Each line is a packet detected by Wireshark.

Let this run as long as needed. It will keep capturing the network traffic until you press the stop button (the red one in the top bar).

## Packets analysis

After doing a capture of the network traffic, you can then analyze its content. The screen is split in three main parts:



Packets list, with filters



Packet details



Packet bytes

- **Packets list:** the first part. Where you can see all captured packets, and use the display filters to only show those that interest you. I'll get back to this later.
- **Packet details:** when you select one packet, you can see its content, in a more or less readable text format.
- **Packet bytes:** the exact packet content, with bytes and hexadecimal format (less useful for us ^^).

On the first part, you'll see the macro information, like source, destination and protocol. It will help you to select the ones you are interested in. For example, if you are looking for suspect HTTP activity from a specific IP address, you can skip everything unrelated (like DNS requests and other IP addresses). I'll show you how to filter this list in the next section.

Packet analysis with Wireshark could be a dedicated article, or even a full book on its own. So, I won't give you more details here, but you can [check the official documentation](#) to learn more about it.

## Filters

---

But the main issue when you are looking for something specific on Wireshark, is to filter the packets list (the first table). Devices talk quite a lot on our networks, and it might be overwhelming to see all of these packets.

That's why **Wireshark includes a field near the top of the screen, where you can enter a formula to only show the packets that are potentially interesting for you (or exclude them).**

Here is a first example:

```
tcp.port == 80
```

It's exactly what you think, it will display only the packets using the port 80 (HTTP traffic in general).

Reading these filters is quite intuitive, but instead of trying random formulas, here are some of the most useful ones:

- **Filter the IP address (to analyze only one device on your network):**

```
ip.addr==192.168.222.8
```

- **You can also filter the source or destination IP addresses with:**

```
ip.src==192.168.222.8
```

```
ip.dst==192.168.222.25
```

- **As seen in the previous example, you can filter the ports with:**

```
tcp.port==80
```

```
udp.port==5060
```

Many other filters options are available, but those few should already be pretty useful to filter your list.

Also, you can use different operators and boolean statement to create more complex

filters.

Here are a few examples:

Filter	Description
ip.src!=192.168.222.25	Source IP address is not 192.168.222.25
vnc or http	Only display VNC or HTTP protocols
ip.src==192.168.222.8 and ip.dst==192.168.222.1	Filter traffic between my computer and the gateway

When you start typing something in the filter field, it will autofill with available options and your filter history. So, even if it seems complicated when you start from scratch, it will become easier and easier overtime. And as for the packet analysis, you can easily find help online for more complex filters.

## Wireshark Alternatives In Command Line

If you want to record network activity on another device, or want to use SSH to connect to your Kali Linux system, it's possible to use other tools, as Wireshark doesn't offer a command line interface.

Here are two alternatives you can try in this case.

### Tcpdump

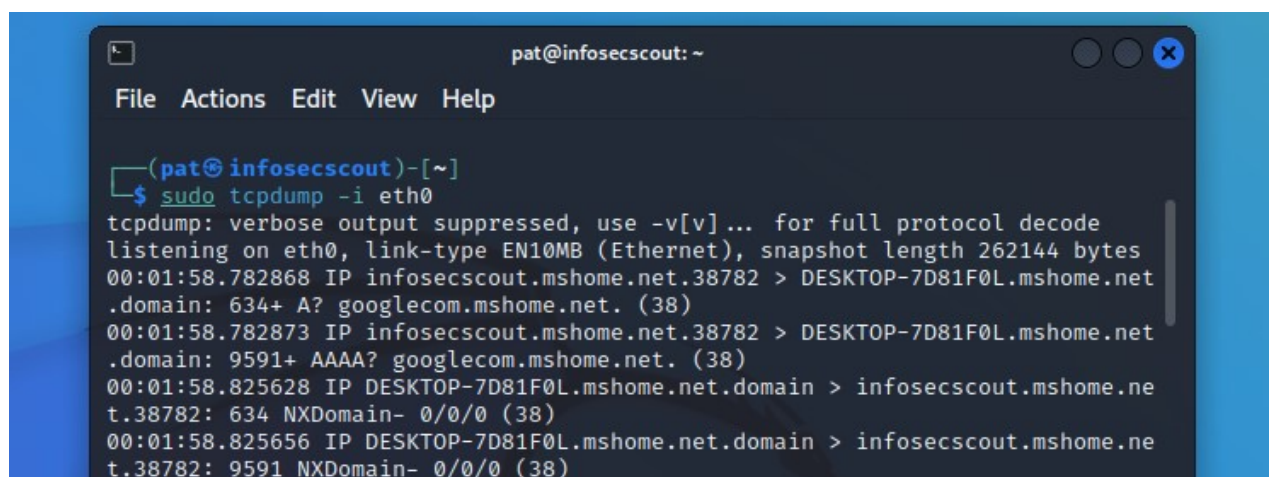
**Tcpdump is a command-line tool you can use to capture network traffic.**

Tcpdump is preinstalled on Kali Linux, but if needed you can easily install it on any device via ATP:

```
sudo apt install tcpdump
```

Using the main command will just show all the packets on your screen:

```
sudo tcpdump -i <interface>
```



**Hide your IP address and location with a free VPN:**

[Try it for free now](#), with advanced security features.

2900+ servers in 65 countries. It's free. Forever.

This is not really useful.

But **you can add several options to your command, to only show what you want, and store the result in a capture file**, for example:

```
sudo tcpdump -i eth0 -w tcpdump.cap
```

You'll then record only the traffic on the Ethernet network card, and save the results in a file (tcpdump.cap). Use CTRL+C to stop the capture.

What's great is that you can then open this file with Wireshark (File > Open), and use all the nice features we have seen previously.

I'll generally have Wireshark on my computer, do captures on my servers with tcpdump and then open the file on the computer to analyze it.

**To see all the options for tcpdump, either use:**

```
sudo tcpdump --help
```

or

```
man tcpdump
```

Or maybe you are a pen tester and got access to a Linux device, that is important on the network you are auditing. You may be able to run tcpdump on it, and transfer the capture file to your computer to analyze it comfortably with Wireshark.

## Tshark

---

**Tshark is an alternative to Wireshark, to be used in the terminal directly.** It's created by the same developers as Wireshark, so you'll find many similarities.

**It's also pre-installed on Kali Linux, and available in the default repository on most distributions, so, if needed, you can install it with:**

```
sudo apt install tshark
```

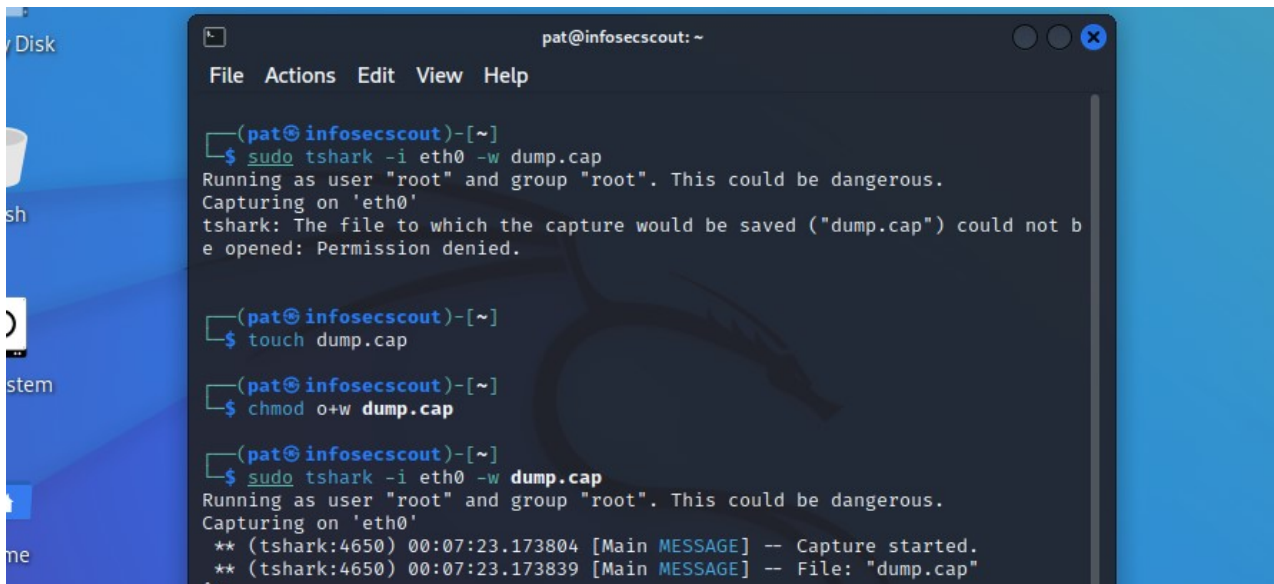
And then use a similar command to create a capture. But you need to create the destination file first, and add some permissions (I don't know exactly why you need this with sudo, but it doesn't work without it).

```
touch tshark.cap
```

```
chmod o+w tshark.cap
```

```
sudo tshark -i eth0 -w tshark.cap
```



A terminal window titled 'pat@infosecscout: ~' with a menu bar (File, Actions, Edit, View, Help). The terminal shows a sequence of commands and their outputs. First, the user runs 'sudo tshark -i eth0 -w dump.cap', which results in a 'Permission denied' error. Then, the user runs 'touch dump.cap' and 'chmod o+w dump.cap'. Finally, the user runs 'sudo tshark -i eth0 -w dump.cap' again, which successfully starts the capture on 'eth0' and saves it to 'dump.cap'. The terminal output includes timestamps and messages from tshark.

```
(pat@infosecscout)-[~]
$ sudo tshark -i eth0 -w dump.cap
Running as user "root" and group "root". This could be dangerous.
Capturing on 'eth0'
tshark: The file to which the capture would be saved ("dump.cap") could not be opened: Permission denied.

(pat@infosecscout)-[~]
$ touch dump.cap

(pat@infosecscout)-[~]
$ chmod o+w dump.cap

(pat@infosecscout)-[~]
$ sudo tshark -i eth0 -w dump.cap
Running as user "root" and group "root". This could be dangerous.
Capturing on 'eth0'
** (tshark:4650) 00:07:23.173804 [Main MESSAGE] -- Capture started.
** (tshark:4650) 00:07:23.173839 [Main MESSAGE] -- File: "dump.cap"
```

Like with tcpdump, you can press **CTRL+C** to stop the capture, and import the file in Wireshark to analyze it. But tshark also has a ton of options you can use, to do the same things as in Wireshark with the command line (for example, `-f` allow you to use capture filters, and `-Y` to use display filters).

**Whenever you're ready for more security, here are things you should think about:**

- **Break free from Gmail**: You should be able to choose what happens to your data. With Proton, only you can read your emails. [Get private email](#).
- **Protect yourself online**: Use a high-speed Swiss VPN that safeguards your privacy. Open-source, no activity logs. [Get Proton VPN risk-free](#).
- **Master Linux commands**: A sure method to learn (and remember) Linux commands. Useful ones only, one at a time, with clear explanations. [Download the e-book](#).