Using tcpdump to analyze traffic

Tcpdump is a network utility used to capture incoming and outgoing traffic. Here's everything you need to know about using tcpdump on Linux.

Linux comes with a wide variety of network utilities to choose from. Tcpdump is a powerful network tool that can capture and analyze network traffic if you need to troubleshoot network problems on Linux.

Let's practice with the tcpdump command and explore how to use it to capture network traffic.

Install tcpdump in Linux

Tcpdump comes pre-installed with all mainstream Linux distributions and security-based alternatives, so you should be able to use it right away by typing tcpdump with the sudo prefix.

In case you are unable to run the tcpdump command and stuck at the error *"tcpdump: command not found"* , let's learn how to install tcpdump on Linux machine.

To install tcpdump, fire up a terminal and run the command corresponding to the Linux distribution you are currently using:

On Debian/Ubuntu derivatives, run:

```
sudo pacman -S tcpdump
```

On Arch-based systems, run:

```
sudo pacman -S tcpdump
```

To install the tcpdump utility on Fedora, CentOS, and RHEL, issue the following command:

```
sudo dnf install tcpdump
```

Note that if you are asked to install libcap, enter **Yes** or **Y** as this is a core dependency, otherwise tcpdump will refuse to start. This will install the tcpdump utility and resolve the *"command not found"* error .

Now that tcpdump is installed on your system, let's explore the different options and functionalities it offers.

Capture network traffic with tcpdump

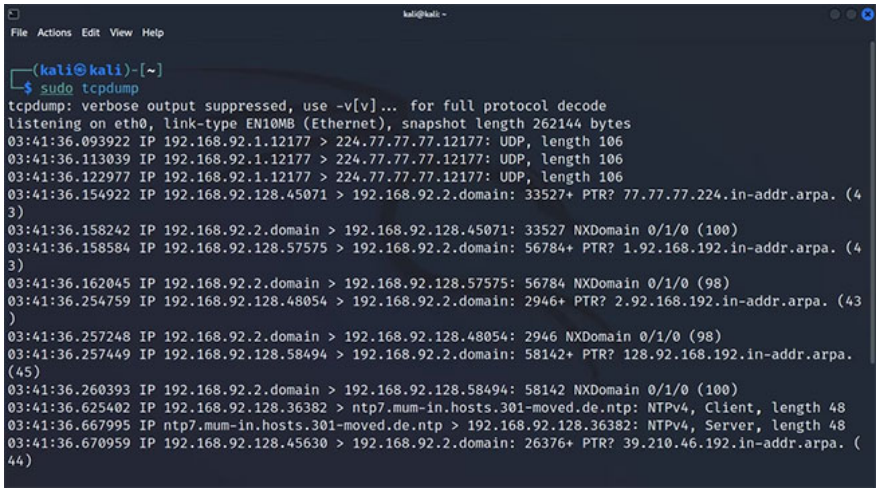
Tcpdump provides a lot of flags to modify its execution but it can also be run as a standalone command. However, running tcpdump without any flags or arguments will miss its full potential. It is better to use some flags to tweak the execution and output as needed.

Enter this command to monitor network transmissions using tcpdump:

```
sudo tcpdump
```

Now, tcpdump will start automatically capturing network packets until a interrupt signal is sent with the **Ctrl + Z** shortcut , interrupting the process manually. To limit the total number of packets captured, use the -c flag and enter the desired packet limit next to it:

```
sudo tcpdump -c 5
```



If you can't understand the output right now, you first need to get familiar with the tcpdump output format.

Check available network interfaces with tcpdump

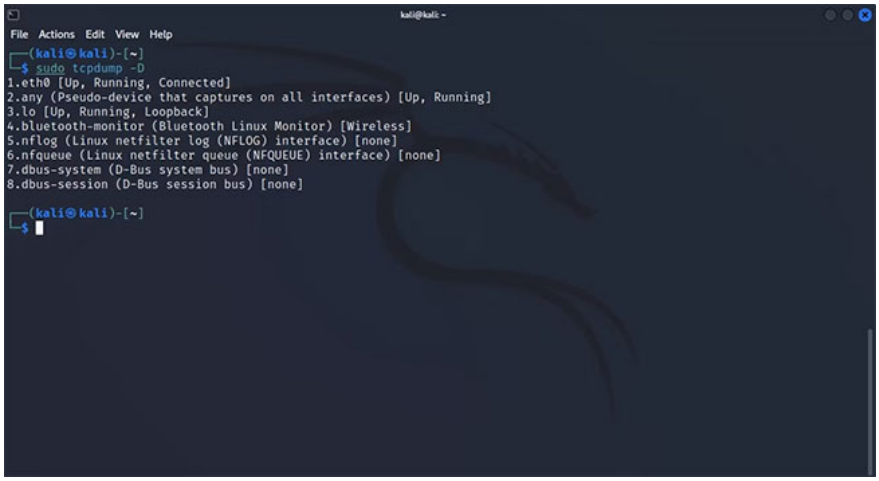
By default, tcpdump captures traffic from any available network interface. If there are multiple network interfaces active, you may want to specify the network interface from which tcpdump will capture packets. To start tcpdump on a specific interface, you will first have to find out the interface name.

Here's how to list all available network interfaces with tcpdump:

```
sudo tcpdump -D
```

Alternatively, you can add the **--list-interface** flag to the command:

```
sudo tcpdump --list-interfaces
```



The returned output contains a list of all active network interfaces that tcpdump can listen on. To configure tcpdump to capture transmissions from a specific network interface, enter the following command:

```
sudo tcpdump -i interface_id
```

Alternatively, you can add the **--interface** flag to the command:

```
sudo tcpdump --interface interface_id
```

Now that you have captured a few packets, let's take a closer look at them and see how you can tweak the output to make it more readable.

Explore tcpdump filters

Tcpdump has the potential to capture a large amount of traffic in a single run. Such an information overload can throw you off track when investigating or troubleshooting a problem with a particular host or network protocol.

This is where tcpdump filters come in handy. You can chain the tcpdump command with certain flags to filter out network traffic and capture specific packets. You can then store these packets and analyze them later to find the root of any network-related issues. Let's take a look at how to use filters in tcpdump.

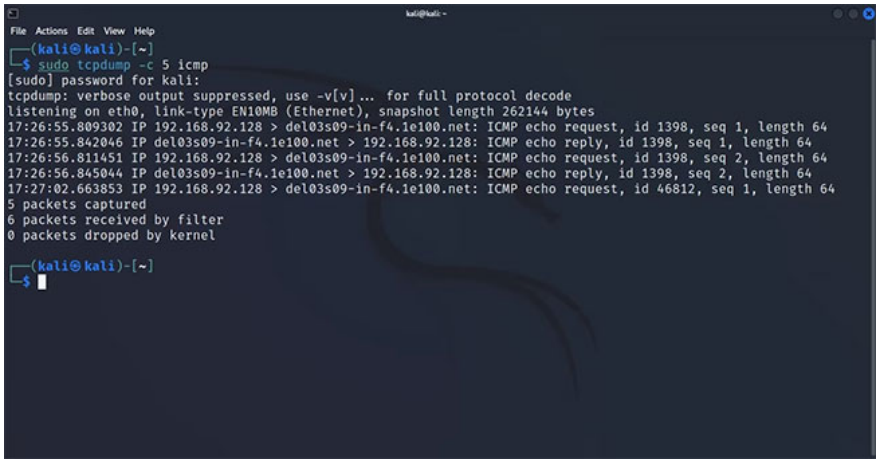
Packet filtering based on the network protocol being used

To filter packets transmitted over a specific protocol, enter the protocol name using the tcpdump command and it will capture only packets transmitted over the specified network protocol.

For example, to capture ICMP-based packets, you simply append icmp to the end of the tcpdump command. The process is the same if you only want to capture UDP or TCP packets.

```
sudo tcpdump -c 5 icmp
```

This command will only return output if there is data exchange via the ICMP protocol.



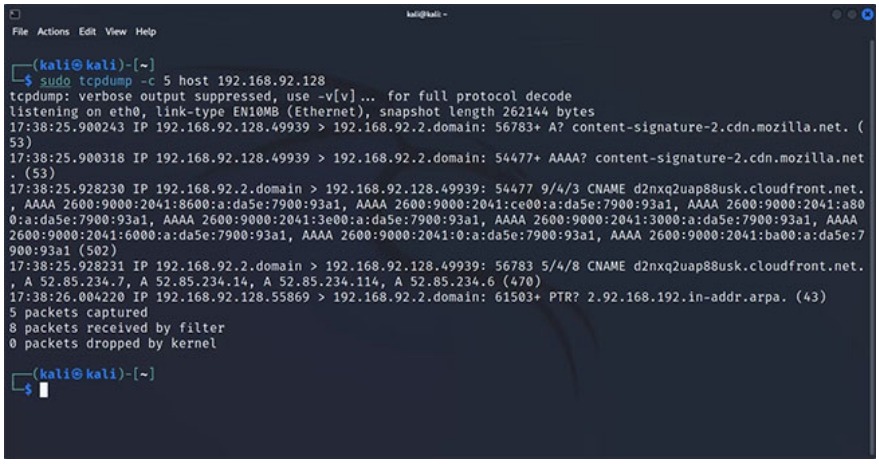
Host-based packet filtering

You can configure tcpdump to capture packets related to a single host with the host parameter. This is especially useful when all systems on your network are up and running except for one. This filter allows you to perform targeted investigations and speeds up the overall troubleshooting process because you are not distracted by unnecessary data.

To capture packets related to a specific host, specify the host's network address with the host parameter:

```
sudo tcpdump -c 5 host 192.168.2.1
```

Similar to the network protocol filter, this command will only return output if any ongoing transmissions are related to the specified host.



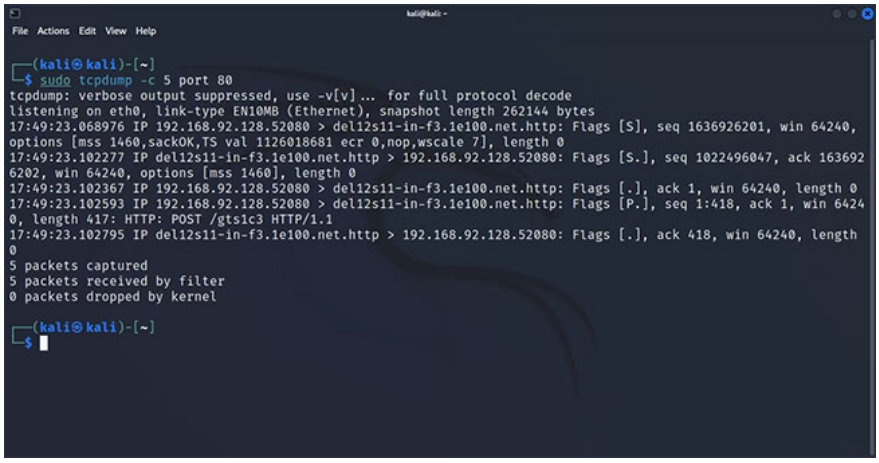
Active Port Based Packet Filtering

Tcpdump comes equipped with a parameter that allows you to filter network traffic and capture only packets transmitted to or from a specific port.

To capture packets coming from a specific port, append the port flag to the tcpdump command and specify the port number next to it. For example, to capture all incoming or outgoing HTTP traffic, specify port 80:

```
sudo tcpdump -c 5 port 80
```

Tcpdump will listen on port 80, waiting for HTTP transmissions. When it detects HTTP packets in the network, it will capture them.

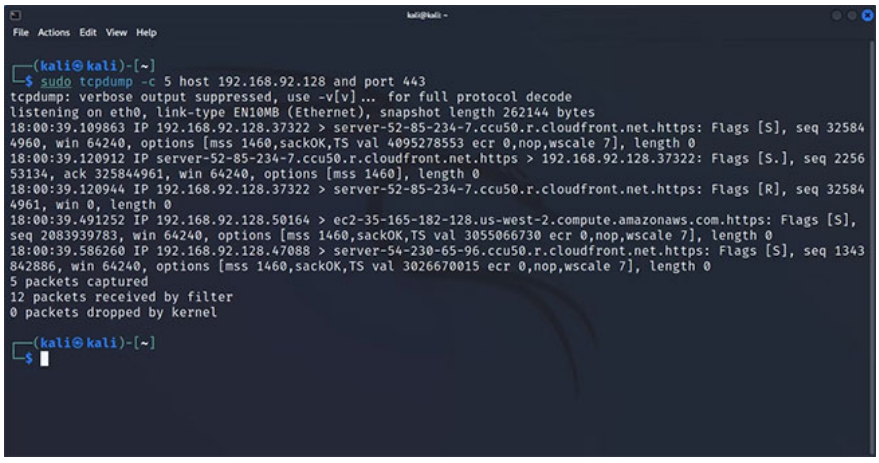


Combine filters together for advanced sorting

The previous sections discussed how you can filter traffic based on port, protocol, or host, but what if you want to capture traffic from a specific host's port using a specific network protocol? Well, you're in luck because this is possible, thanks to the ability to use logical operators with the tcpdump command.

To capture packets from an individual host using port 443, use the following command:

```
sudo tcpdump -c 5 host 192.168.2.1 and port 443
```

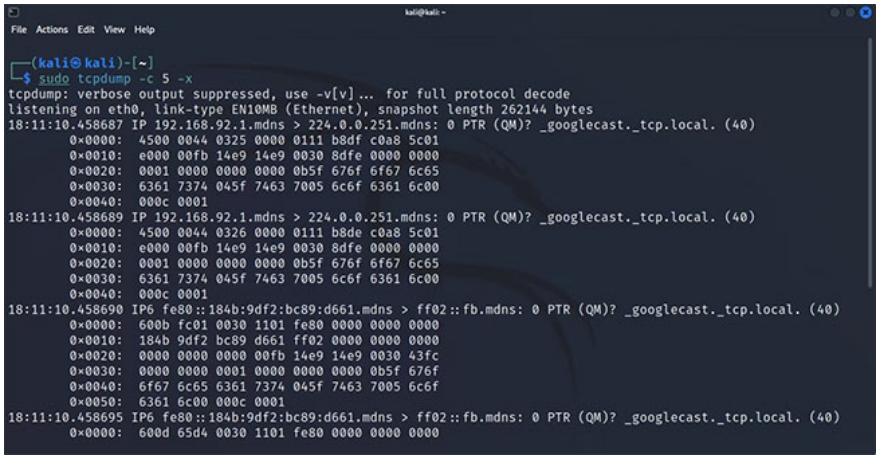


Check the contents of captured packets

By default, tcpdump displays the headers of a packet in its output. While this is more than enough in most cases, sometimes you may want or need to take a deeper look at the captured data. You can pass certain parameters to the tcpdump command to examine the contents of the captured packet.

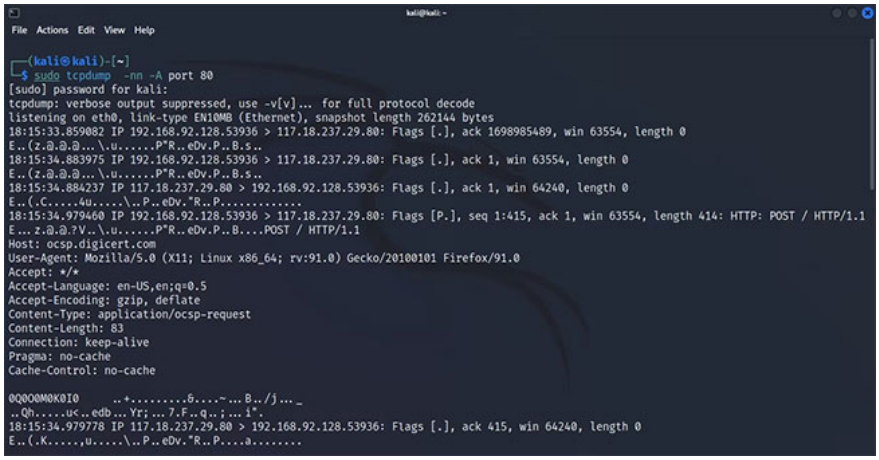
Here's how to view the contents of the packages:

```
sudo tcpdump -c 5 -x
```



This command returns the hex version of the contents of a captured packet. If you want to see the ASCII form of the data, you can pass the **-A** parameter with:

```
sudo tcpdump -A
```



Save tcpdump output to a file

Like most other Linux command line tools, you can save the output generated by tcpdump to a file for later reference.

This can be done by adding the **-w** flag to the command. Once executed, tcpdump will store the captured data in a .pcap file for later analysis using tcpdump or other network monitoring tools like Wireshark.

Enter this command to save your tcpdump command output to a file:

```
sudo tcpdump -w capture.pcap
```

To read the .pcap file, you can use tcpdump with the **-r** parameter :

```
sudo tcpdump -r capture.pcap
```

Linux offers a plethora of networking tools that can solve any networking problem as long as it is on the software side. Knowing how to use some of the best networking tools in Linux will definitely come in handy, whether you are a sysadmin who manages networks for a living or just a daily Linux user.

Since the actual list of available network commands can be overwhelming, here is a list of some of the most important Linux networking tools you should know.