**Getting started with tcpdump**

Et bilde som inneholder tekst, skjermbilde, Font, programvare

KI-generert innhold kan være feil.

***Tool for troubleshooting network issues by analyzing datapackets***

*Downloading the tool if not already installed*

$ sudo apt-get install tcpdump

*Downloading the python library if not already installed*

$ sudo apt-get install python3-libpcap

*Listing the interfaces available for capture*

$ tcpdump –list-interfaces OR tcpdump -D

*Selecting the interface, which can be abbreviated to -i*

$ sudo tcpdump --interface any

$ sudo tcpdump -i any

*Selecting how many times you want to keep going by adding -c for count with a value in this case 5 times*

$ sudo tcpdump -i any -c 5s

*To stop dumping*

**\*CTRL + C\***

*If you are troubleshooting network, it might be easier disabling the name resolution by adding -n and port resolution by using -nn*

$ sudo tcpdump -i any -c 5 -n *# disabling the name resolution*

$ sudo tcpdump -i any -c 5 -nn *# disabling the name and port resolution*

**Understanding the output:**

08:41:13.729687 IP 192.168.64.28.22 > 192.168.64.1.41916: Flags [P.], seq 196:568, ack 1, win 309, options [nop,nop,TS val 117964079 ecr 816509256], length 372

*The output may vary depending on the packet being sent but this is a general format.*

***08:41:13.729687***

The first field represents the timestamp of the received packet as per the local clock

***IP 192.168.64.28.22*** *# 22 is the port*

The next represents the network layer protocol, in this case an IPv4. For IPv6 packets, the value is IP6. Which is the source IP address and port. This is followed by the destination IP address and port ***192.168.64.1.41916***

***Flags [P.]***

After the source and destination, you find the TCP flags. Typical values for this field include the following:

|  |  |  |
| --- | --- | --- |
| **Value** | **Flag type** | **Description** |
| **S** | SYN | Connection Start |
| **F** | FIN | Connection Finish |
| **P** | PUSH | Data push |
| **R** | RST | Connection reset |
| **.** | ACK | Acknowledgement |

This field can also be a combination of these values such as [**S.**] for a **SYN-ACK** packet.

***Seq 196:568***

Next is the sequence number of the data contained in the packet. For the first packet captured, this is an absolute number. Subsequent packets use a relative number to make it easier to follow. In this example, the sequence is ***seq 196:568***, which means this packet contains bytes ***196 to 568*** of this flow.

***ack 1***

This is followed by the Ack Number. In this case, it is 1 since this is the side sending data. For the side receiving data, this field represents the next expected byte (data) on this flow. For example, the Ack number for the next packet in this flow would be ***568.***

***win 309***

The next field is the window size, which represents the number of bytes available in the receiving buffer, followed by TCP options such as the MSS (Maximum Segment Size) or Window Scale.

***Length 372***

Finally, we have the packet length, which represents the length in bytes of the payload data. The length is the difference between the last and first bytes in the sequence number.

**Filtering packets**

One of tcpdump’s most powerful features is its ability to filter the captured packets using a variety of parameters, such as source and destination IP addresses, ports, protocols, etc.

To filter based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by using the following command

$ sudo tcpdump -i any -c 5 icmp

***Host***

The following command filters to display only packets to and from the specified host

$ sudo tcpdump -i any -c 5 -nn host \*host ip\*

***Port***

The following command filters packets based on the desired service or port. For

example, capture packets related to a web (HTTP) service by using this command

$ sudo tcpdump -i any -c 5 -nn port 80

***Source or Destination IP/hostname***

You can also filter based on the source or destination IP address or hostname. For

example, to capture packets from host 192.168.122.98

$ sudo tcpdump -i any -c 5 -nn src 192.168.122.98

# NOTE: You can also use dst to instead use destination, for example:

$ sudo tcpdump -i any -c 5 -nn dst 192.168.122.98

***Complex expressions***

You can also combine filters by using the logical operators “and” and “or”, for

example:

$ sudo tcpdump -i any -c 5 -nn src 192.168.122.98 and port 80

**Checking packet content**

When troubleshooting network connectivity issues, we need to inspect the content of the packet to ensure that the message we’re sending contains what we need or that we received the expected response. To see the packet content, tcpdump provides two additional flags: -X to print content in hex, and ASCII or -A to print the content in ASCII.

For example, inspecting the HTTP content of a web request like this:

$ sudo tcpdump -I any -c 10 -nn -A port 80

**Saving capture to a file**

$ sudo tcpdump -I any -c 10 -nn -w webserver.pcap port 80

# NOTE:

This saves the file in a binary format, which means that you cannot open it with a simple text editor.

.pcap is the file extension for packet captures. Which can also be opened with wireshark.

If you also want to ensure the packets are being captured, use the “-v” parameter

If you want to open the file in the terminal you can use the “-r“:

$ tcpdump -nn -r webserver.pcap