

*System Admin*

*Training Assignments*

|  |  |
| --- | --- |
| **Program Code** |  |
| **Issue/Revision** | **x/y** |
| **Effective date** | **04/Aug /2023** |

**Assignment Day 04. Networking**

1. **Thực hành gõ lệnh cấu hình và troubleshooting networking linux**

|  |  |
| --- | --- |
| **command** | **explanation** |
| /sys/class/net | Contains names of network adapters on this Debian computer. |
| ip a | Display TCP/IP configuration for network adapters. |
| ifconfig | Legacy tool to display TCP/IP configuration. |
| ip n | Display ARP cache. |
| arp | Legacy tool to display ARP cache. |
| ip r | Display routing information. |
| route | Legacy tool to display routing information. |
| /etc/network/interfaces | Contains the TCP/IP configuration of network adapters. |
| ifup foo | Bring the interface **foo** up (with its configuration). |
| ifdown foo | Bring the interface **foo** down. |
| /etc/resolv.conf | Contains DNS server configuration. |
| /etc/services | Lists port to application configuration. |
| /etc/protocols | Lists layer 4 protocol configuration. |
| ping | A tool to test an IP connection. |
| traceroute | A tool to display routes between computers. |
| ss | A tool to display open ports and their applications. |
| netstat | A legacy tool to display open ports and their applications. |
| dhclient | A daemon that maintains an IP configuration received from a DHCP server. |

##### **network adapters**

Every server has one or more network adapters. You can see a list of network adapters in **/sys/class/net**. The files in there are symbolic links to **/sys/devices/**.



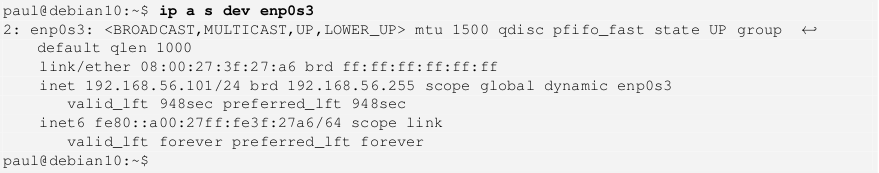
##### **ip a**

The first tool of this chapter is the **ip** tool. Most common use is maybe **ip a** to get IP-address, subnet and MAC information from all network adapters. This includes the loopback adapter named **lo**.

Our **debian10** server of this screenshot has IP-address 192.168.56.101 on interface enp0s3 and IP-address 10.0.2.101 on interface enp0s8 (Look behind **inet**). After the **inet6** keyword are the IPv6 addresses for this server.



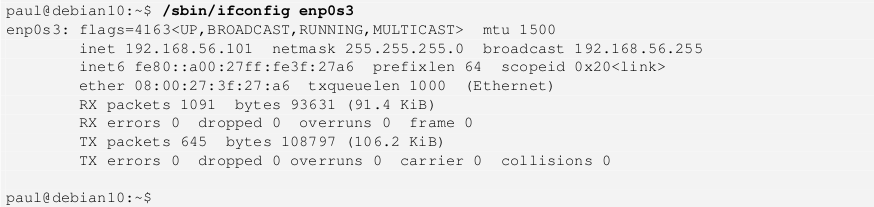
You can show information about just one adapter by typing **ip a s dev** followed by the adapter name. The **a** is short for **address**, and the **s** is short for **show**.



##### **net-tools**

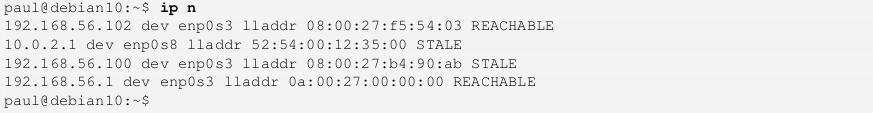
Wait, what about the **ifconfig** command, isn’t that the standard on Debian Linux? The answer is **not anymore**. The **ifconfig**

command is part of the **net-tools** package, which is no longer installed by default. But it still works, as this screenshot shows.

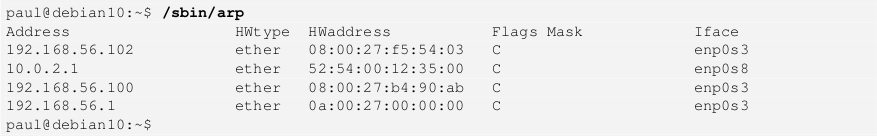


##### **ip n**

Next is the **ip n** command to show the **ARP table** of the server. Whenever your server has contact with another device on the network, then it’s MAC-address will be kept in the **ARP cache** (for a while). You can see this **ARP cache** with the **ip n** command (n is short for **neighbour**).

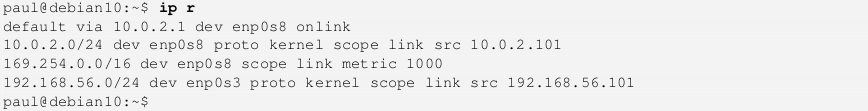


The **net-tools** equivalent of this command is the **arp** command, which in my humble opinion gives a much more readable output than **ip n**.



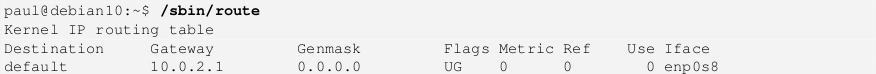
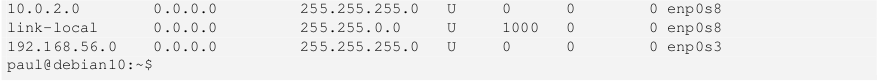
##### **ip r**

Type **ip r** to see the **routing table** of your server. The screenshot shows that 10.0.2.1 is the **default gateway** and the server is connected to two subnets (192.168.56.0/24 and 10.0.2.0/24).



Again I personally find the output of the net-tools **route** command much more readable. This is identical to typing **netstat -nr**

(which is also from net-tools).

##### **/etc/network/interfaces**

Configuration of network interfaces is done in the **/etc/network/interfaces** file. The first interface defined in this file is often the

**loopback** interface. You probably never have to change this.



Next is the primary interface, which is usually configured when installing the Debian server. In this example DHCP is used (even though the server gets a fixed IP-address from DHCP). The name of the network adapter **enp0s3** can be different on your server.



Our server has a fixed IP-address on the second network card. The fixed IP is 10.0.2.101 and the subnet mask is 255.255.255.0 (or /24). This interface also has a default gateway.



For DHCPv6 (to obtain an IPv6) address add the following line to the interface configuration.



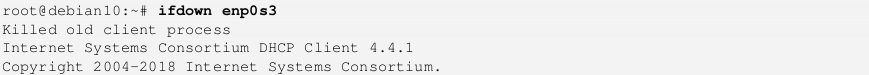
##### **ifup & ifdown**

You can turn off an interface with the **ifdown** command. And you can turn it back on with **ifup**. These commands will use the information in **/etc/network/interfaces** . Nothing much seems to happen in this screenshot because **enp0s8** has a fixed IP-address.



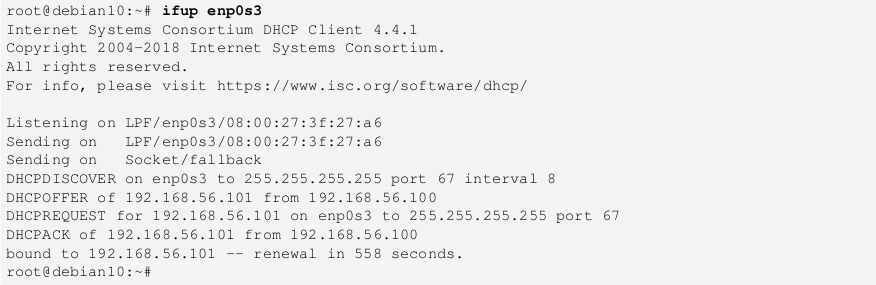
Using **ifdown** and **ifup** on the **enp0s3** interface provides a lot of output because this interface is configured to use DHCP. The

**ifdown** command will release the IP-address from the DHCP server.



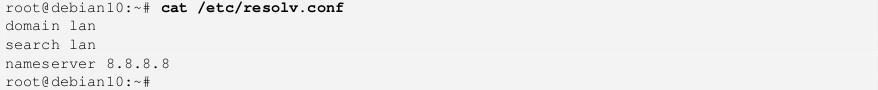


Using **ifup** will again contact the DHCP server to get an IP-address. Depending on the configuration of the DHCP server this can be the same IP-address or a new one. Setting up a DHCP server is a whole new chapter later in this book.



##### **/etc/resolv.conf**

Usually when using DHCP then the **/etc/resolv.conf** file will be automatically configured to contain the correct DNS server. If not, then you can manually add a DNS server in this file. In the example below we use **8.8.8.8** as a DNS server.



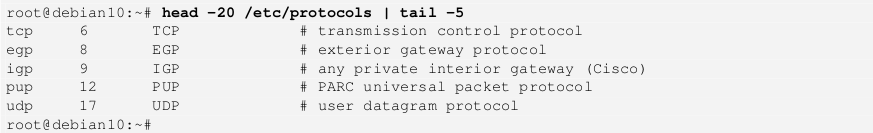
##### **/etc/services**

Debian Linux comes with an extensive list of ports for TCP and UDP in the **/etc/services** file. Here you (and applications) can find applications related to a certain port.



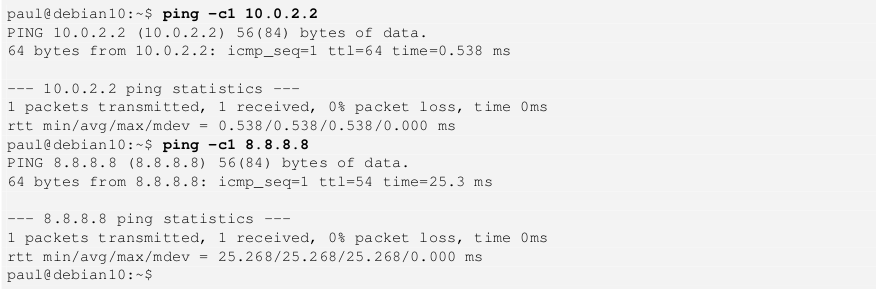
##### **/etc/protocols**

The **/etc/protocols** file contains protocols like TCP, UDP and ICMP associated with a number to be used in the IP-datagram. You probably never need this file, when talking about protocols people actually mean protocols in **/etc/services**.



##### **ping**

One of the most basic tools for testing a **TCP/IP** connection is **ping**. You can literally **ping** a local or remote server. This screenshot first **pings** a local server on the same subnet, and then does a **ping** to 8.8.8.8 which is a Google server on the Internet.

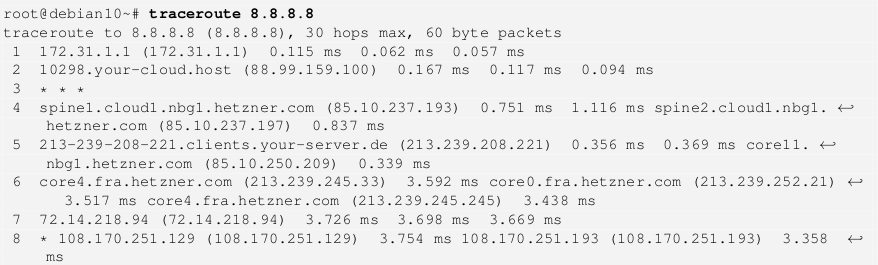


The **ping** to the local server happens in less than a millisecond, where the **ping** tot the Google server takes 25 milliseconds. The

**-c1** option performs just one ping, otherwise use **Ctrl-c** to interrupt the command.

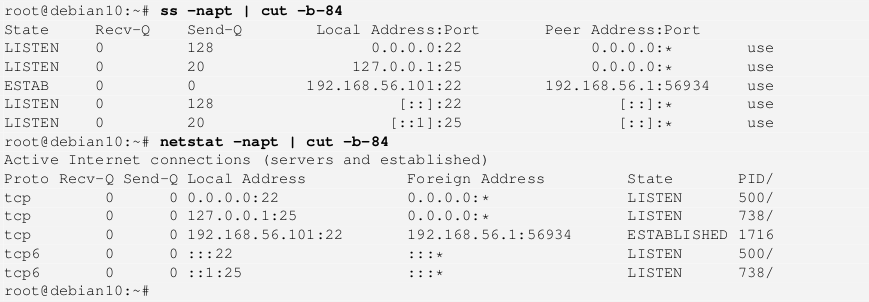
##### **traceroute**

The **traceroute** tool can be used to trace how many routers there are between your server and another server. Each router is called a **hop**. The 8.8.8.8 server is eight **hops** away from our **debian10** server in this screenshot.



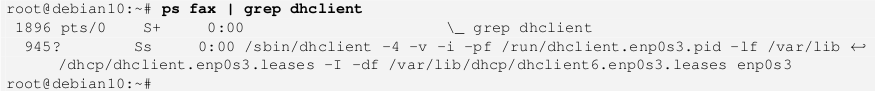
##### **ss**

The **ss** tool is the modern version of the legacy **netstat** tool. It has much the same functionality and most of the same options. In this screenshot we use both tools to look at open TCP ports on our server. The **t** option is for TCP, replace with **u** for UDP. Port 22 is used for SSH connections.



##### **dhclient**

On servers with a fixed IP-address it can happen that the DHCP client named **dhclient** is running to obtain this fixed IP-address. Do not kill this client, as your server will lose its IP-address after a while.



##### **tcpdump**

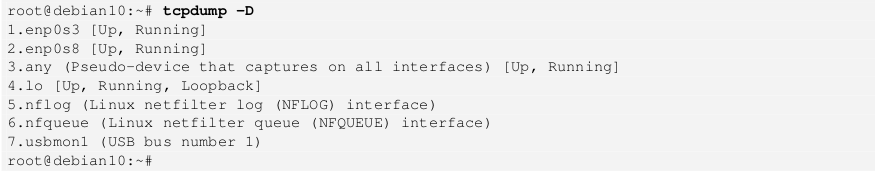
|  |  |
| --- | --- |
| **command** | **explanation** |
| tcpdump -D | List all interfaces. |
| tcpdump -i foo | Listen on the **foo** interface. |
| tcpdump -c 4 | Sniff only four packets. |
| tcpdump -n | Don’t resolve to DNS names. |
| tcpdump -w foo.pcap | Save output to a file named **foo.pcap**. |
| tcpdump port 53 | Listen only on port 53. |
| tcpdump host 8.8.8.8 | Listen only for packets concerning host 8.8.8.8. |
| tcpdump -X | Display hexadecimal and ASCII in output. |
| tcpdump -A | Display ASCII packet contents. |
| tcpdump arp | Sniff only ARP packets. |
| wireshark | A graphical sniffer to learn about networks. |

The tools to use for sniffing the network are **tcpdump** on the command line, and **wireshark** if you have (and prefer) a graphical interface. We will first take a look at **tcpdump**.



###### **Listing all interfaces**

Type **tcpdump -D** to obtain a list of interfaces on which **tcpdump** can listen. You should see a list of all your network adapters followed by an **any** adapter and **lo** for the loopback device. The **nflog** and **nfqueue** devices are part of the netfilter firewall, these are out of the scope of this chapter.



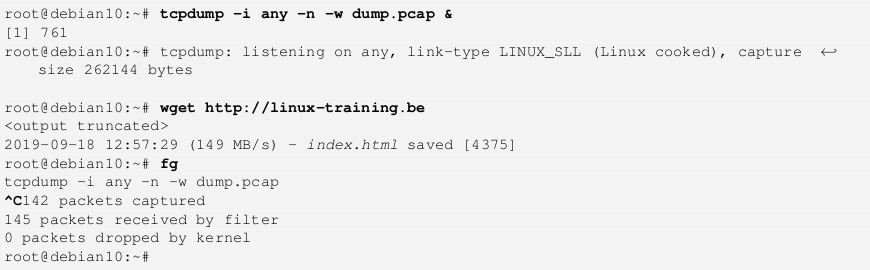
###### **Choosing an interface**

Using the **-i** option of **tcpdump** you can specify an interface to listen on. The amount of traffic passing over an interface can be overwhelming. Especially if you select the interface of your **ssh** connection. In this screenshot we select the second network adapter on our server, there is currently no traffic on this adapter.



###### **writing to a file**

Using **tcpdump -w** will write the packets to a file, in this case to **dump.pcap**. I put the command in background here so I could generate some traffic with **wget** of an **index.html** file.



##### **Practice**

* + 1. List the names of the network adapters on your server.
    2. List the IP and MAC address of each of your network adapters.
    3. List the cache of MAC-IP addresses of your server.
    4. List the routing table of your server.
    5. Display the configuration of the network adapters.
    6. Which DNS server is being used on your server?
    7. Display the list of TCP and UDP protocols.
    8. Test your connection to 8.8.8.8.
    9. Display the hops between you and 8.8.8.8.
    10. List the open TCP ports on your server.

##### **Solution**

1. List the names of the network adapters on your server.

ls /sys/class/net/

2. List the IP and MAC address of each of your network adapters.

ip a

/sbin/ifconfig

3. List the cache of MAC-IP addresses of your server.

ip n

/sbin/arp

4. List the routing table of your server.

ip r

/sbin/route

5. Display the configuration of the network adapters.

more /etc/network/interfaces

6. Which DNS server is being used on your server?

cat /etc/resolv.conf

7. Display the list of TCP and UDP protocols.

more /etc/services

8. Test your connection to 8.8.8.8.

ping 8.8.8.8

9. Display the hops between you and 8.8.8.8.

traceroute 8.8.8.8

10. List the open TCP ports on your server.

ss -napt