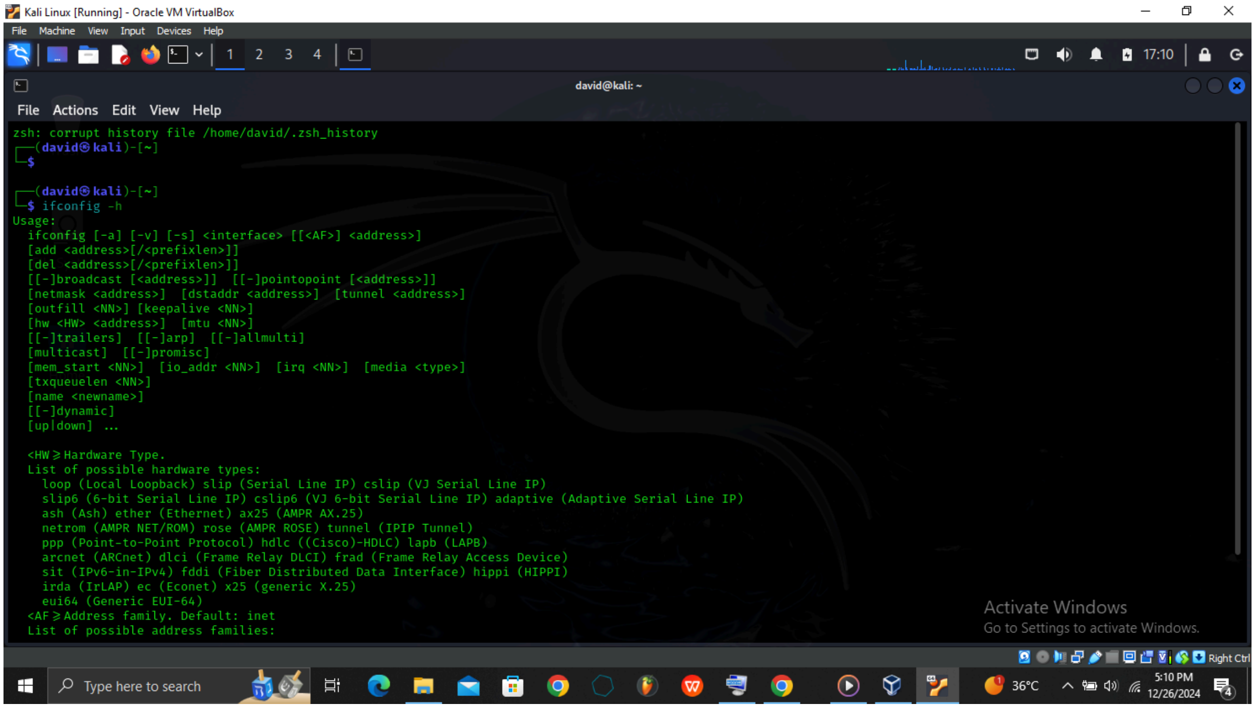
LAB 19:

USING IFCONFIG TO VIEW AND MODIFY NETWORK INFORMATION ON KALI LINUX

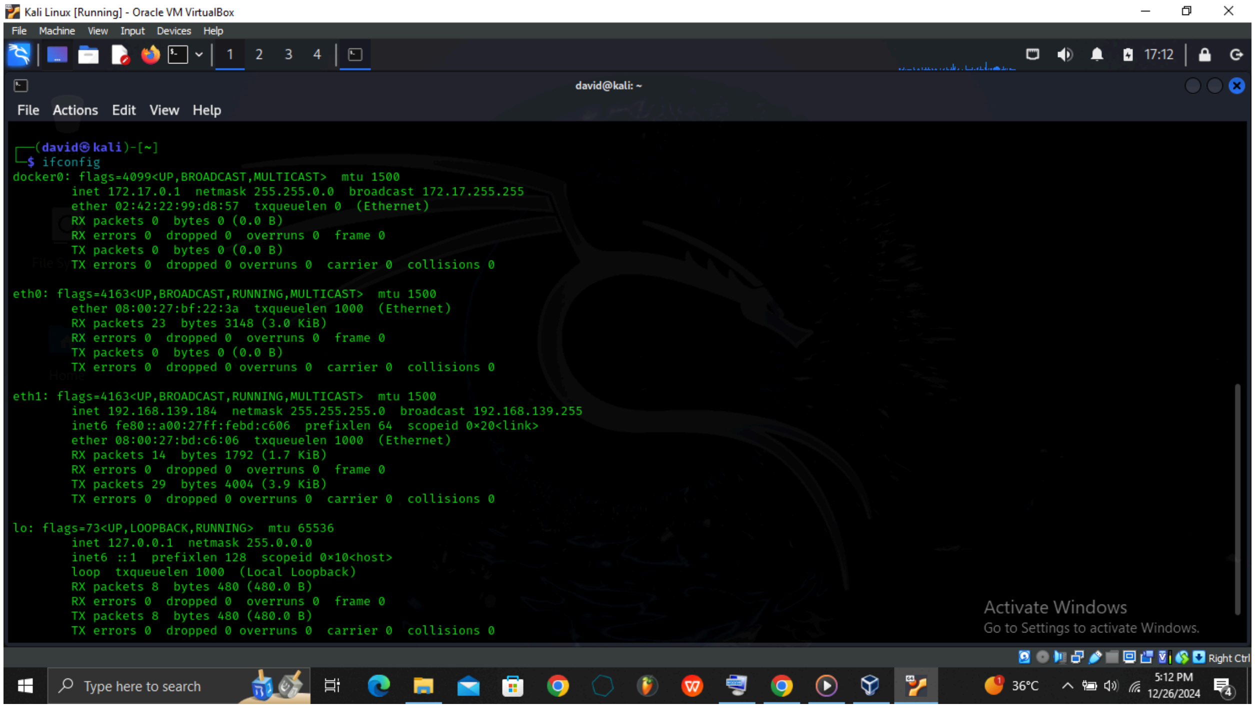
TOOL: KALI LINUX

Step 1: We will begin by viewing the help information screen by executing the following command:
Ifconfig -h


Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
17:10
david@kali: ~
File Actions Edit View Help
Activate Windows
Go to Settings to activate Windows.
Right Ctrl
Type here to search
36°C
5:10 PM
12/26/2024

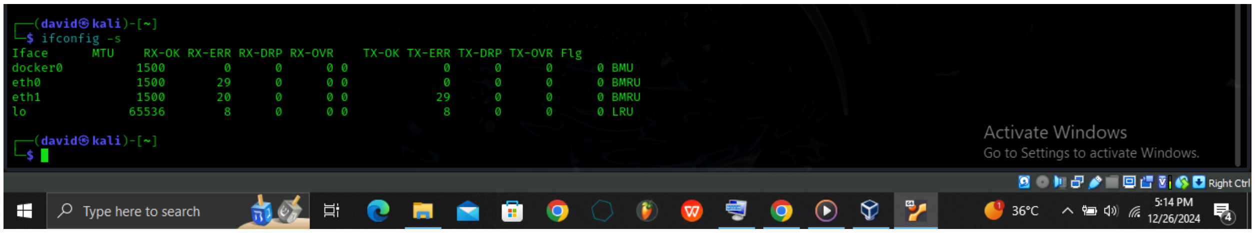
```
zsh: corrupt history file /home/david/.zsh_history
┌──(david㉿kali)-[~]
└─$ 

┌──(david㉿kali)-[~]
└─$ ifconfig -h
Usage:
  ifconfig [-a] [-v] [-s] <interface> [[<AF>] <address>]
  [add <address>[/<prefixlen>]]
  [del <address>[/<prefixlen>]]
  [[-]broadcast [<address>]]  [[-]pointopoint [<address>]]
  [netmask <address>]  [dstaddr <address>]  [tunnel <address>]
  [outfill <NN>] [keepalive <NN>]
  [hw <HW> <address>]  [mtu <NN>]
  [[-]trailers]  [[-]arp]  [[-]allmulti]
  [multicast]  [[-]promisc]
  [mem_start <NN>]  [io_addr <NN>]  [irq <NN>]  [media <type>]
  [txqueuelen <NN>]
  [name <newname>]
  [[-]dynamic]
  [up|down] ...

  <HW>=Hardware Type.
  List of possible hardware types:
    loop (Local Loopback) slip (Serial Line IP) cslip (VJ Serial Line IP)
    slip6 (6-bit Serial Line IP) cslip6 (VJ 6-bit Serial Line IP) adaptive (Adaptive Serial Line IP)
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) tunnel (IPIP Tunnel)
    ppp (Point-to-Point Protocol) hdlc ((Cisco)-HDLC) lapb (LAPB)
    arcnet (ARCnet) dlci (Frame Relay DLCI) frad (Frame Relay Access Device)
    sit (IPv6-in-IPv4) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
    irda (IrLAP) ec (Econet) x25 (generic X.25)
    eui64 (Generic EUI-64)
  <AF>=Address family. Default: inet
  List of possible address families:
```

Type “ifconfig” to view your networking information.


Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
17:12
david@kali: ~
File Actions Edit View Help
Activate Windows
Go to Settings to activate Windows.
Right Ctrl
Type here to search
36°C
5:12 PM
12/26/2024

```
┌──(david㉿kali)-[~]
└─$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:22:99:d8:57  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 08:00:27:bf:22:3a  txqueuelen 1000  (Ethernet)
        RX packets 23  bytes 3148 (3.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.139.184  netmask 255.255.255.0  broadcast 192.168.139.255
        inet6 fe80::a00:27ff:febd:c606  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:bd:c6:06  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 1792 (1.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29  bytes 4004 (3.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 2: To display a short list output, we can use the following command: ifconfig -s

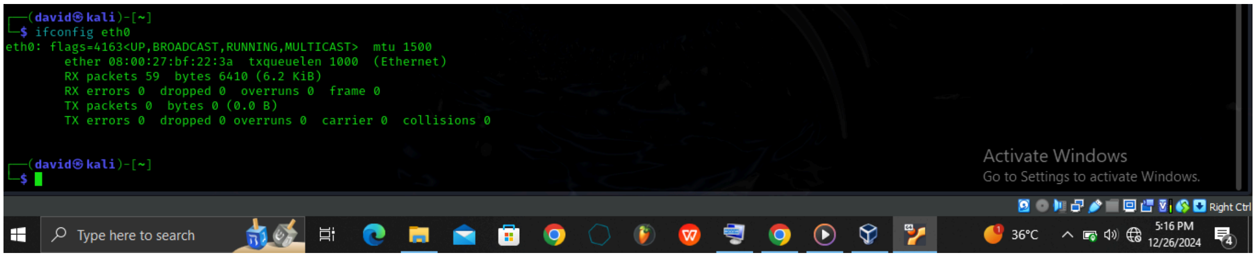


```
┌──(david㉿kali)-[~]
└─$ ifconfig -s
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
docker0          1500         0      0     0 0             0      0      0      0 BMU
eth0             1500        29      0     0 0             0      0      0      0 BMRU
eth1             1500        20      0     0 0            29      0      0      0 BMRU
lo              65536         8      0     0 0             8      0      0      0 LRU

┌──(david㉿kali)-[~]
└─$ 
```

Step 3: We can display information about a specific interface by using the following command:

ifconfig [interface-name]

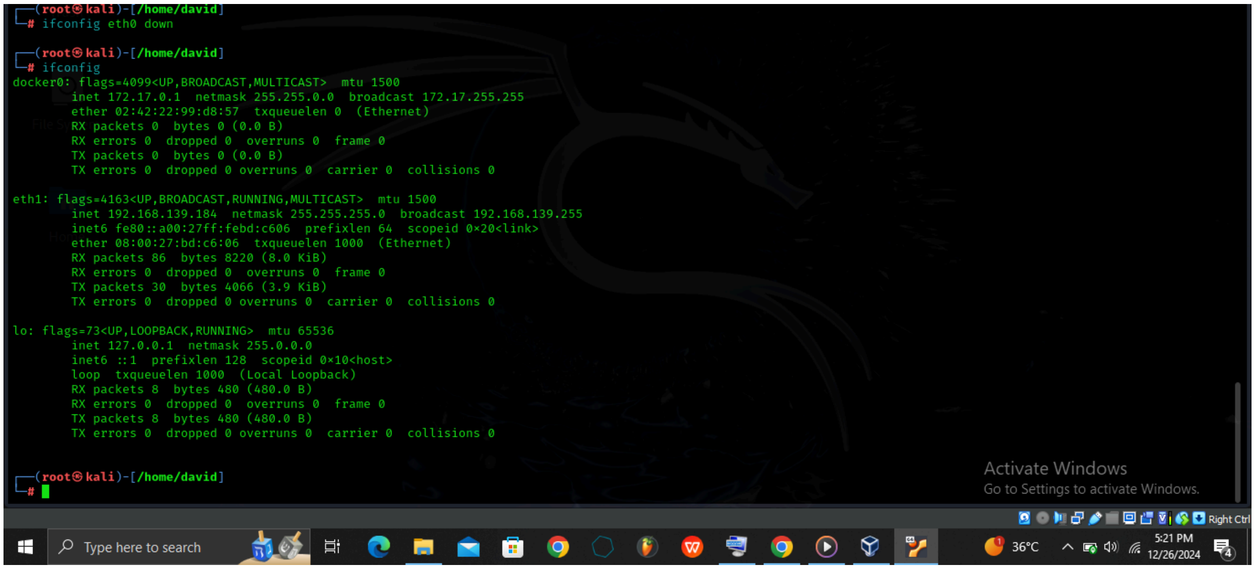
This is useful for determining interface information and for debugging.



```
┌──(david㉿kali)-[~]
└─$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 08:00:27:bf:22:3a  txqueuelen 1000  (Ethernet)
        RX packets 59  bytes 6410 (6.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(david㉿kali)-[~]
└─$ 
```

Step 4: We can disable or enable a network interface using an ifconfig flag. For example:

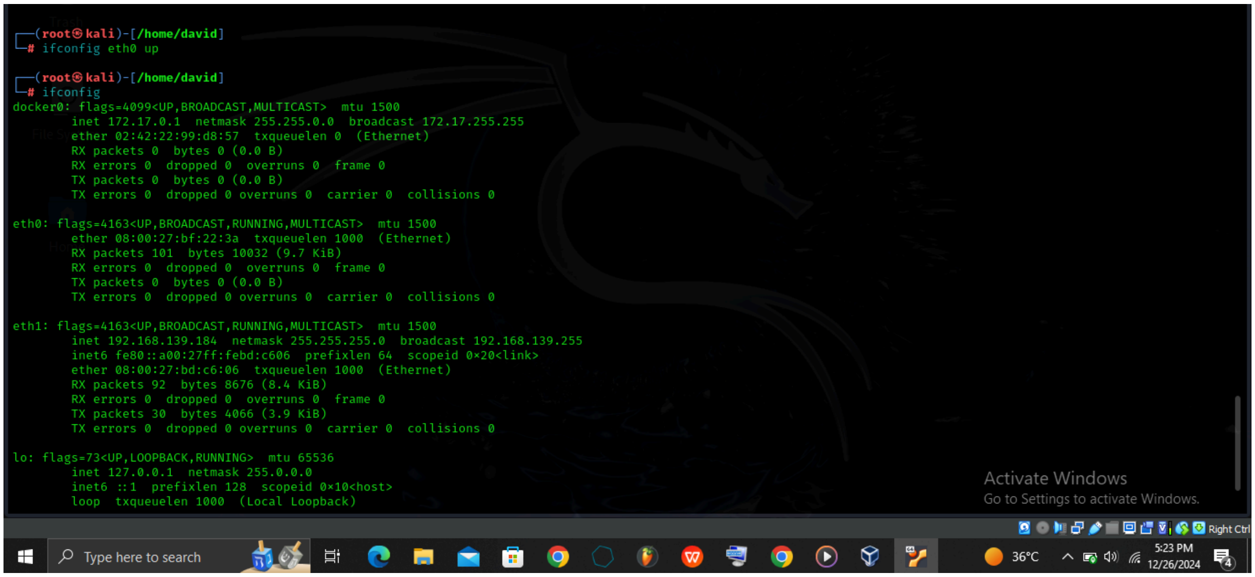
ifconfig eth0 down



```
┌──(root㉿kali)-[/home/david]
└─# ifconfig eth0 down

┌──(root㉿kali)-[/home/david]
└─# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:22:99:d8:57  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.139.184  netmask 255.255.255.0  broadcast 192.168.139.255
        inet6 fe80::a00:27ff:febd:c606  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:bd:c6:06  txqueuelen 1000  (Ethernet)
        RX packets 86  bytes 8220 (8.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 4066 (3.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(root㉿kali)-[/home/david]
└─# 
```

This command will disable our local connection to the Wi-Fi card. To enable it, enter the following command:

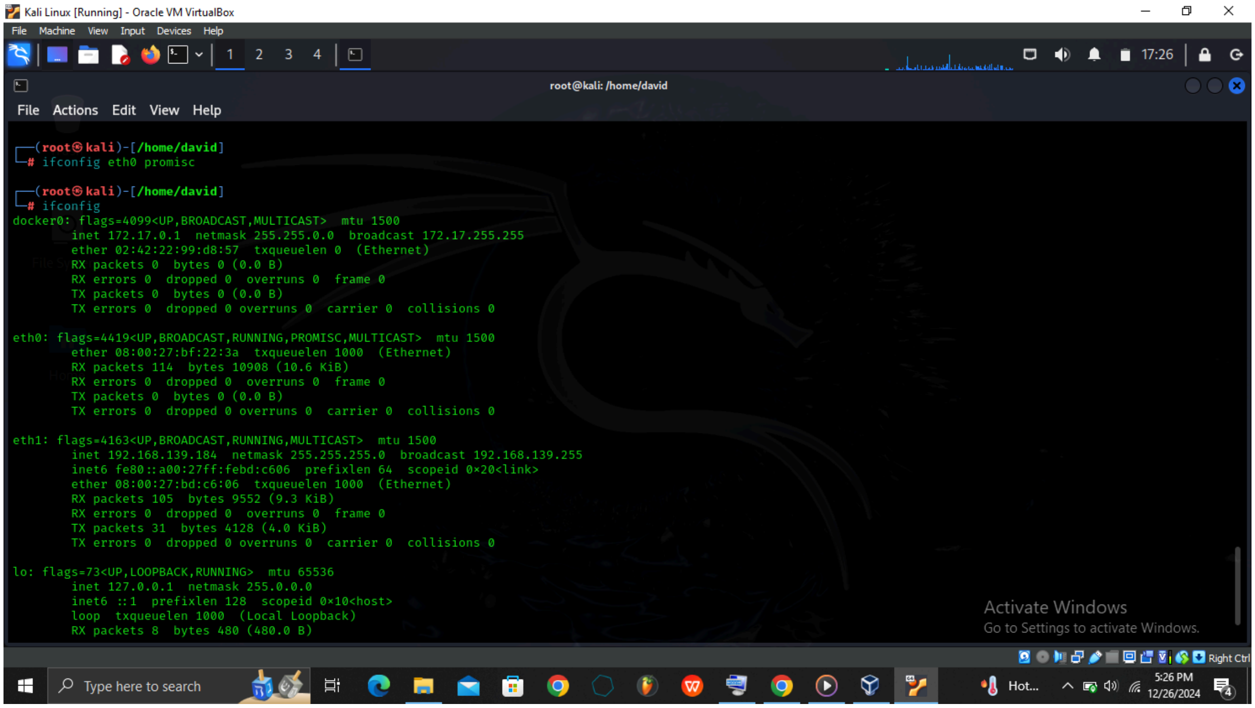
ifconfig eth0 up



```
┌──(root㉿kali)-[/home/david]
└─# ifconfig eth0 up

┌──(root㉿kali)-[/home/david]
└─# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:22:99:d8:57  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 08:00:27:bf:22:3a  txqueuelen 1000  (Ethernet)
        RX packets 101  bytes 10032 (9.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.139.184  netmask 255.255.255.0  broadcast 192.168.139.255
        inet6 fe80::a00:27ff:febd:c606  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:bd:c6:06  txqueuelen 1000  (Ethernet)
        RX packets 92  bytes 8676 (8.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 4066 (3.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
```

Step 5: We can use ifconfig to enable promiscuous mode on an interface. This will allow the interface to receive all packets on the network. You will need a compatible network card for this to work correctly:

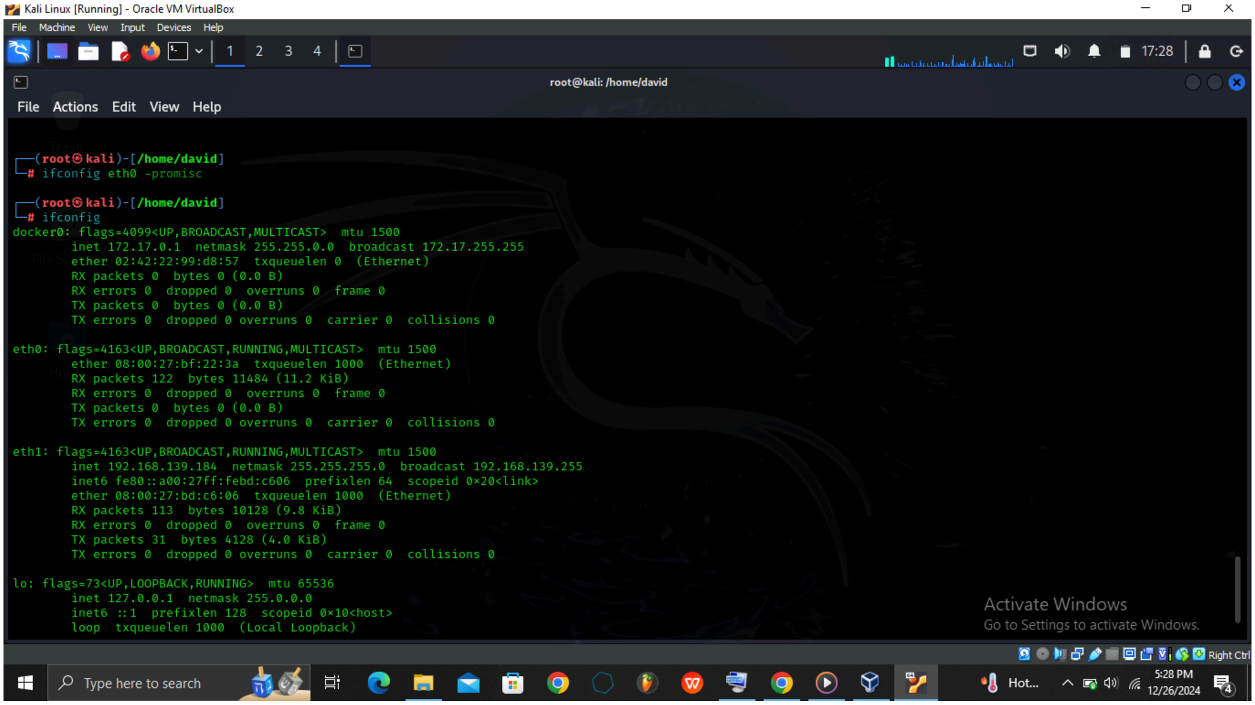
ifconfig eth0 promisc


Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
17:26
root@kali: /home/david
File Actions Edit View Help
Activate Windows
Go to Settings to activate Windows.
Right Ctrl
Type here to search
Hot...
5:26 PM
12/26/2024

```
┌──(root㉿kali)-[/home/david]
└─# ifconfig eth0 promisc

┌──(root㉿kali)-[/home/david]
└─# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:22:99:d8:57  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>  mtu 1500
        ether 08:00:27:bf:22:3a  txqueuelen 1000  (Ethernet)
        RX packets 114  bytes 10908 (10.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.139.184  netmask 255.255.255.0  broadcast 192.168.139.255
        inet6 fe80::a00:27ff:febd:c606  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:bd:c6:06  txqueuelen 1000  (Ethernet)
        RX packets 105  bytes 9552 (9.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 31  bytes 4128 (4.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
```

This can be disabled using the following command:

ifconfig eth0 -promisc



Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
17:28
root@kali: /home/david
File Actions Edit View Help
Activate Windows
Go to Settings to activate Windows.
Right Ctrl
Type here to search
Hot...
5:28 PM
12/26/2024

```
┌──(root㉿kali)-[/home/david]
└─# ifconfig eth0 -promisc

┌──(root㉿kali)-[/home/david]
└─# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:22:99:d8:57  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 08:00:27:bf:22:3a  txqueuelen 1000  (Ethernet)
        RX packets 122  bytes 11484 (11.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.139.184  netmask 255.255.255.0  broadcast 192.168.139.255
        inet6 fe80::a00:27ff:febd:c606  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:bd:c6:06  txqueuelen 1000  (Ethernet)
        RX packets 113  bytes 10128 (9.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 31  bytes 4128 (4.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
```

The ifconfig tool also enables you to change the MAC address associated with a network interface. This can be done with the following command:

```
ifconfig [network-name] hw [class] [hardware-address]
```

Eg. ifconfig eth0 hw ether 66:3e:7f:60:f2:1f

There are actually 4 sets of Locally Administered Address Ranges that can be used on your network without fear of conflict, assuming no one else has assigned these on your network:

```
x2-xx-xx-xx-xx-xx
x6-xx-xx-xx-xx-xx
xA-xx-xx-xx-xx-xx
xE-xx-xx-xx-xx-xx
```

To make the changes permanent, open the file below and add the following lines in it:

```
nano /etc/network/interfaces
```

```
pre-up ifconfig eth0 hw ether AA:22:33:44:55:66
```

Reboot the system. The new MAC address will appear.