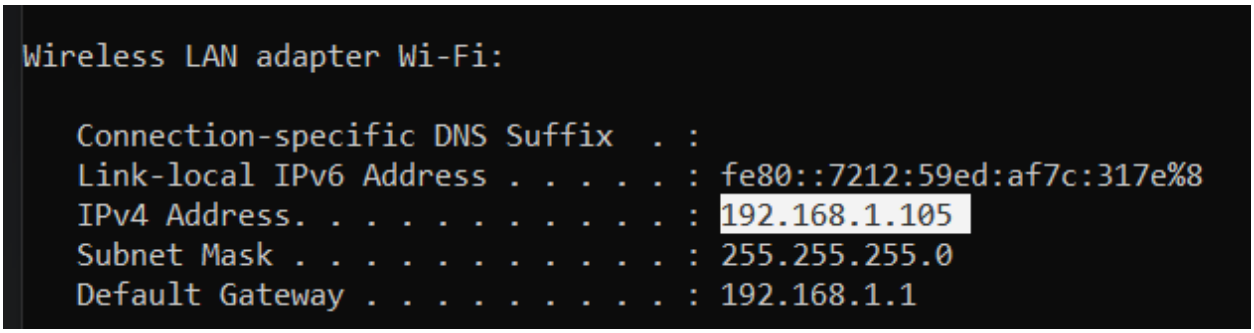
Ping Sweeper

💡 **Ping command: ping [IP ADDRESS] [-c number](for specified number of pings)**

💡 **My IP Address: (I will be using this IP to find all devices in my network)**



💡 **Pinging my actual IP**

```
┌─[wahaj@parrot]─[/opt]
└──╼ $ping 192.168.1.105
PING 192.168.1.105 (192.168.1.105) 56(84) bytes of data.
64 bytes from 192.168.1.105: icmp_seq=1 ttl=127 time=7.86 ms
64 bytes from 192.168.1.105: icmp_seq=2 ttl=127 time=2.52 ms
64 bytes from 192.168.1.105: icmp_seq=3 ttl=127 time=1.60 ms
64 bytes from 192.168.1.105: icmp_seq=4 ttl=127 time=11.2 ms
64 bytes from 192.168.1.105: icmp_seq=5 ttl=127 time=15.6 ms
64 bytes from 192.168.1.105: icmp_seq=6 ttl=127 time=2.28 ms
64 bytes from 192.168.1.105: icmp_seq=7 ttl=127 time=2.03 ms
64 bytes from 192.168.1.105: icmp_seq=8 ttl=127 time=2.04 ms
64 bytes from 192.168.1.105: icmp_seq=9 ttl=127 time=3.13 ms
^C
--- 192.168.1.105 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8204ms
rtt min/avg/max/mdev = 1.597/5.357/15.551/4.749 ms
```

💡 **Putting the command in a file so our result is stored and then we can use the results to sweep the network**

```
┌─[wahaj@parrot]─[~/Desktop]
└──╼ $ping 192.168.1.105 -c 1 > ip.txt
┌─[wahaj@parrot]─[~/Desktop]
└──╼ $cat ip.txt
PING 192.168.1.105 (192.168.1.105) 56(84) bytes of data.
64 bytes from 192.168.1.105: icmp_seq=1 ttl=127 time=7.78 ms

--- 192.168.1.105 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.782/7.782/7.782/0.000 ms
```

💡 **using grep to pick up specific data**

```
┌──[wahaj@parrot]─[~/Desktop]
└─$ cat ip.txt | grep "64 bytes"
64 bytes from 192.168.1.105: icmp_seq=1 ttl=127 time=7.78 ms
```

💡 **OUR PURPOSE RIGHT HERE IS TO PING EVERY SUBNET STARTING FROM .1 TO .255 AND CHECK IF THERE EXISTS A CONNECTION, THIS IS PING SWEEPING**

💡 **Shortlisting the data even more with the cut command**
cut -d[delimiter] string -f[field] count
[By the below command we ask the terminal to grep the line with 64 bytes and extract only the field which has the 4th space]

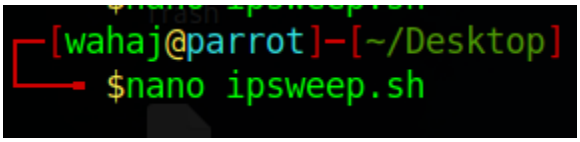
```
┌──[wahaj@parrot]─[~/Desktop]
└─$ cat ip.txt | grep "64 bytes" | cut -d " " -f 4
192.168.1.105:
```

💡 **Finally with the translate command we can remove the extra colon**
tr -d[delimiter] string

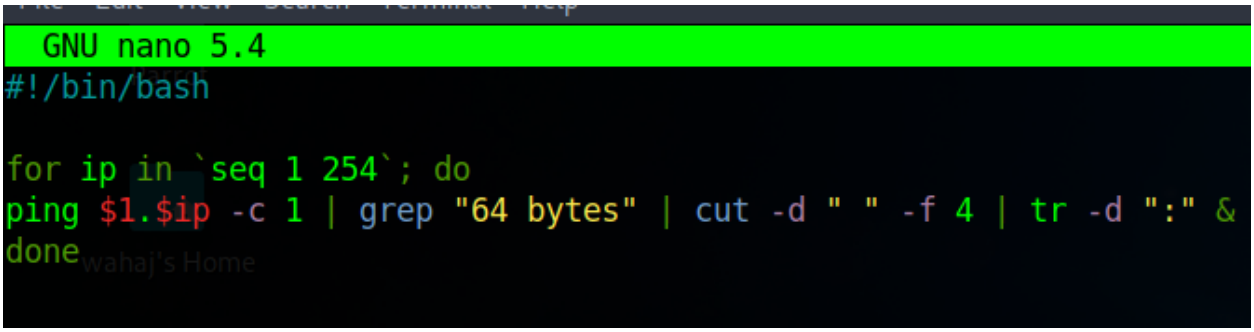
```
┌──[wahaj@parrot]─[~/Desktop]
└─$ cat ip.txt | grep "64 bytes" | cut -d " " -f 4 | tr -d ":"
192.168.1.105
```

Writing The Script

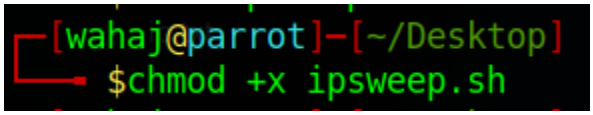
💡 **Create the file**



What this code does is it says for ip in sequence 1 till 254 do ping $1 (first command line argument).$ip(the ip generated by loop) grab the line and extract the ip and done.
& makes the loop run in parallel



Make the script executable



Edit the script to run only with an IP

```
#!/bin/bash

if [ $1 == "" ]
then
echo "You forgot an IP Address.";
echo "Syntax: ./ipsweep.sh [IP ADDRESS]";
else

for ip in `seq 1 254`; do
ping $1.$ip -c 1 | grep "64 bytes" | cut -d " " -f 4 | tr -d ":" &
done
fi
```

Run the Script

```
┌─[wahaj@parrot]─[~/Desktop]
└──╼ $./ipsweep.sh 192.168.1
192.168.1.1
192.168.1.105
192.168.1.108
192.168.1.101
```

Store and Get the results in the file

```
┌─[wahaj@parrot]─[~/Desktop]
└──╼ $./ipsweep.sh 192.168.1 >ips.txt
┌─[wahaj@parrot]─[~/Desktop]
└──╼ $cat ips.txt
192.168.1.1
192.168.1.101
192.168.1.108
192.168.1.105
192.168.1.102
```

one liner with nmap

```
┌──[wahaj@parrot]─[~/Desktop]
└─$for ip in $(cat ips.txt); do nmap $ip & done
[1] 9296
[2] 9297
[3] 9298
[4] 9299
[5] 9300
```

nmap results

```
┌──[wahaj@parrot]─[~/Desktop]
└─$Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-14 07:54 EST
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-14 07:54 EST
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-14 07:54 EST
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-14 07:54 EST
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-14 07:54 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.33 seconds
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.44 seconds
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.56 seconds
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.67 seconds
Nmap scan report for 192.168.1.1
Host is up (0.022s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
1900/tcp open  upnp

Nmap done: 1 IP address (1 host up) scanned in 5.80 seconds
^C
[1]   Done                    nmap $ip
[2]   Done                    nmap $ip
[3]   Done                    nmap $ip
[4]-  Done                    nmap $ip
[5]+  Done                    nmap $ip
```