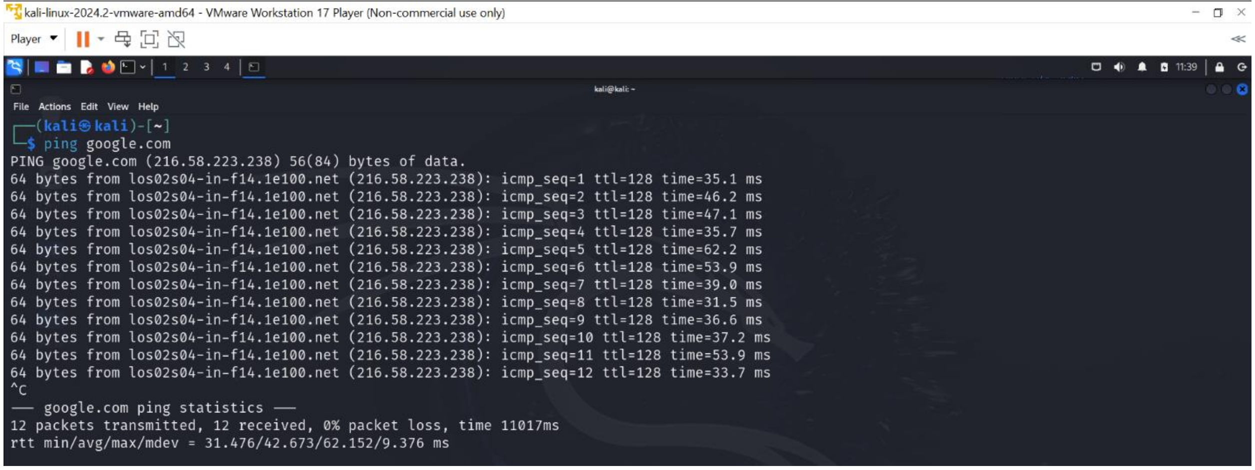
PING AND ITS VARIOUS USES

TOOL: PING

The ping command is a network utility used to test the reachability of a host on an Internet Protocol (IP) network. It also measures the round-trip time for messages sent from the originating host to a destination computer.

TASK 1

Ping google.com


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
1 2 3 4
11:39
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ping google.com
PING google.com (216.58.223.238) 56(84) bytes of data.
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=1 ttl=128 time=35.1 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=2 ttl=128 time=46.2 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=3 ttl=128 time=47.1 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=4 ttl=128 time=35.7 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=5 ttl=128 time=62.2 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=6 ttl=128 time=53.9 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=7 ttl=128 time=39.0 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=8 ttl=128 time=31.5 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=9 ttl=128 time=36.6 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=10 ttl=128 time=37.2 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=11 ttl=128 time=53.9 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=12 ttl=128 time=33.7 ms
^C
— google.com ping statistics —
12 packets transmitted, 12 received, 0% packet loss, time 11017ms
rtt min/avg/max/mdev = 31.476/42.673/62.152/9.376 ms
```

The ping command will continue to send ICMP packages to the destinated IP address until it receives an interruption. To stop the command, just hit the Ctrl + C key combination.

As you will see, a number of lines of information will appear on our screen. This shows the packets being sent from our machine to google.com, as well as the response being received. We sent out 7 packets and received 7 packets back, indicating that google.com is up and responding to requests.

1) The hostname we are pinging. Use “-n” with this command if you want to avoid any reverse DNS lookups. For example: “ping google.com -n”

2) The IP address of the target host.

3) The reverse DNS name of target IP address. It’s different from the original hostname, right? This happens when one hostname has many IP addresses and each IP address has only one DNS name.

4) The number of data bytes. The default is 56, which translates into 64 ICMP data bytes.

5) The ICMP sequence numbers for each packet.

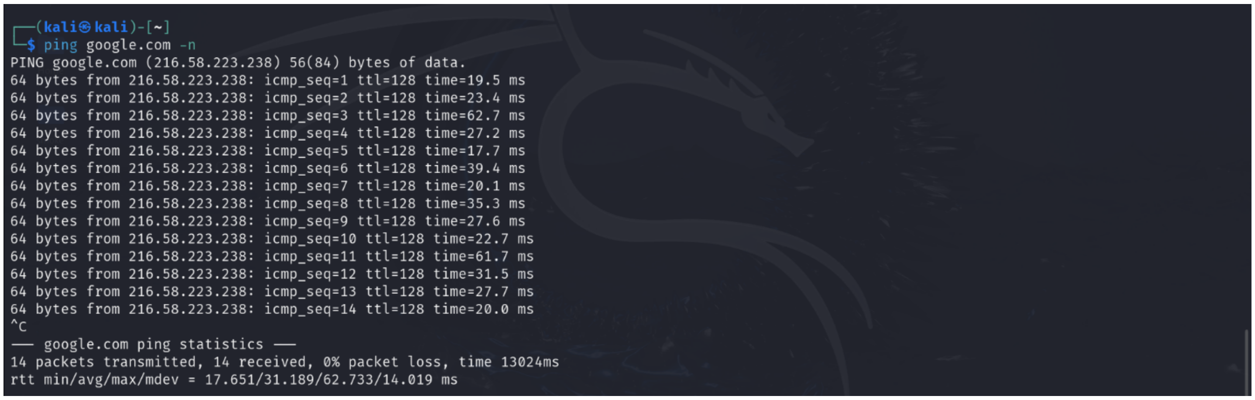
6) TTL: The Time to Live values.

7) The ping time, measured in milliseconds which is the round trip time for the packet to reach the host, and the response to return to the sender. Greater values indicate possible network problems or target’s load.

8) Once the command stops, it displays a statistic including the percentage of packet loss. The packet loss means that the data was dropped somewhere in the network, indicating an issue within the network or target’s performance. If there is a packet loss, you can use the traceroute command to identify where the packet loss occurs.

9) RTT (Round-trip time) metrics of those ping packages. RTT is the duration in milliseconds it takes for a network request to go from a starting point to a target and back again to the starting point.

ping google.com -n



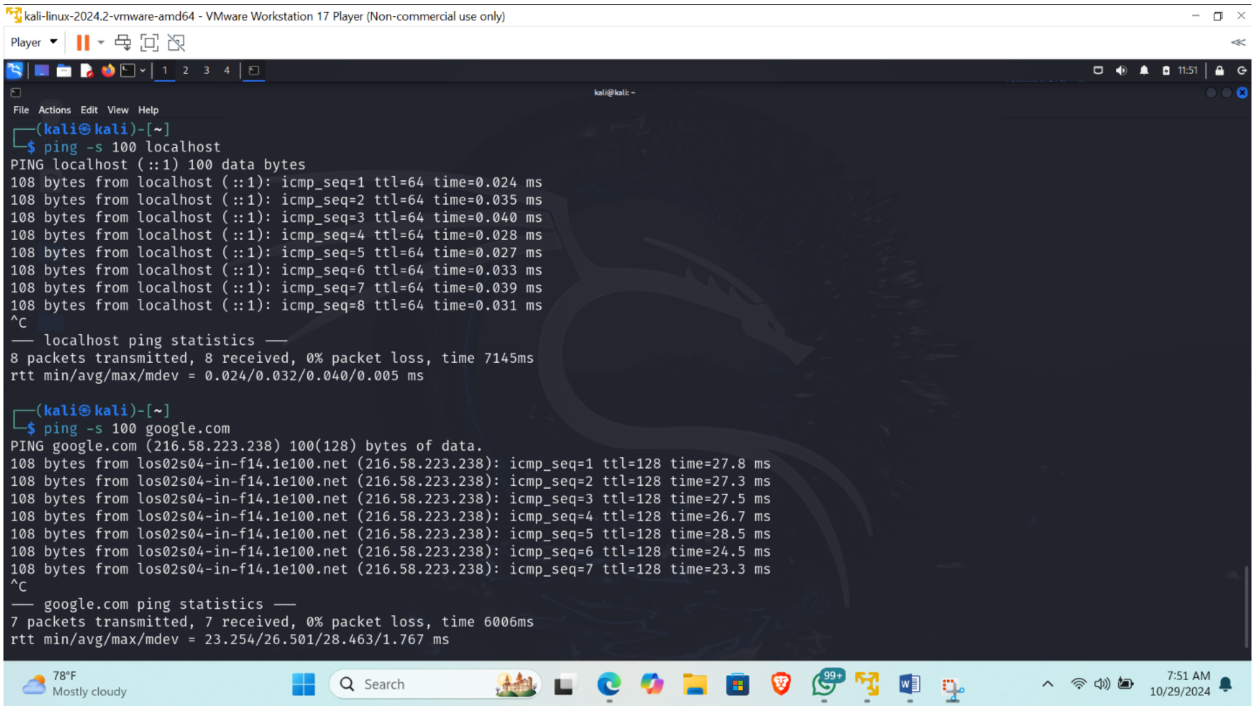
```
┌──(kali㉿kali)-[~]
└─$ ping google.com -n
PING google.com (216.58.223.238) 56(84) bytes of data.
64 bytes from 216.58.223.238: icmp_seq=1 ttl=128 time=19.5 ms
64 bytes from 216.58.223.238: icmp_seq=2 ttl=128 time=23.4 ms
64 bytes from 216.58.223.238: icmp_seq=3 ttl=128 time=62.7 ms
64 bytes from 216.58.223.238: icmp_seq=4 ttl=128 time=27.2 ms
64 bytes from 216.58.223.238: icmp_seq=5 ttl=128 time=17.7 ms
64 bytes from 216.58.223.238: icmp_seq=6 ttl=128 time=39.4 ms
64 bytes from 216.58.223.238: icmp_seq=7 ttl=128 time=20.1 ms
64 bytes from 216.58.223.238: icmp_seq=8 ttl=128 time=35.3 ms
64 bytes from 216.58.223.238: icmp_seq=9 ttl=128 time=27.6 ms
64 bytes from 216.58.223.238: icmp_seq=10 ttl=128 time=22.7 ms
64 bytes from 216.58.223.238: icmp_seq=11 ttl=128 time=61.7 ms
64 bytes from 216.58.223.238: icmp_seq=12 ttl=128 time=31.5 ms
64 bytes from 216.58.223.238: icmp_seq=13 ttl=128 time=27.7 ms
64 bytes from 216.58.223.238: icmp_seq=14 ttl=128 time=20.0 ms
^C
--- google.com ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13024ms
rtt min/avg/max/mdev = 17.651/31.189/62.733/14.019 ms
```

TASK 2

We can set the packet size using the following commands:

ping -s 100 localhost
ping -s 100 google.com

This is useful when testing a system to see how it will respond differently to very small or very large packets. The default packet size of ping is 56.



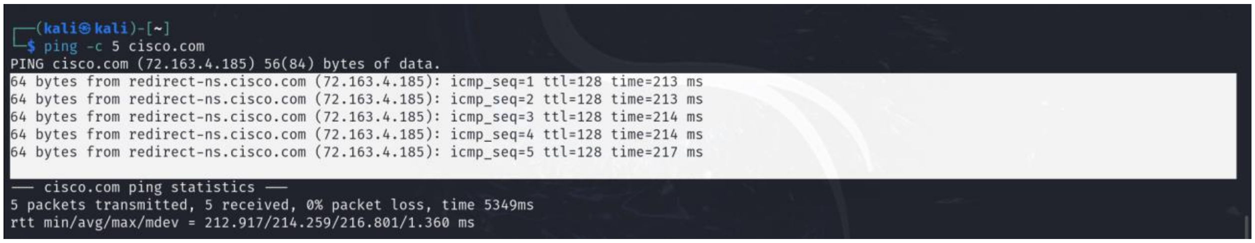
kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
1 2 3 4
11:51
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ping -s 100 localhost
PING localhost (::1) 100 data bytes
108 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.024 ms
108 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.035 ms
108 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.040 ms
108 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.028 ms
108 bytes from localhost (::1): icmp_seq=5 ttl=64 time=0.027 ms
108 bytes from localhost (::1): icmp_seq=6 ttl=64 time=0.033 ms
108 bytes from localhost (::1): icmp_seq=7 ttl=64 time=0.039 ms
108 bytes from localhost (::1): icmp_seq=8 ttl=64 time=0.031 ms
^C
--- localhost ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7145ms
rtt min/avg/max/mdev = 0.024/0.032/0.040/0.005 ms

┌──(kali㉿kali)-[~]
└─$ ping -s 100 google.com
PING google.com (216.58.223.238) 100(128) bytes of data.
108 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=1 ttl=128 time=27.8 ms
108 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=2 ttl=128 time=27.3 ms
108 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=3 ttl=128 time=27.5 ms
108 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=4 ttl=128 time=26.7 ms
108 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=5 ttl=128 time=28.5 ms
108 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=6 ttl=128 time=24.5 ms
108 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=7 ttl=128 time=23.3 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6006ms
rtt min/avg/max/mdev = 23.254/26.501/28.463/1.767 ms
```

TASK 3

As aforementioned, by default, ping will continue to send packages until it receives an interrupt signal. To specify the number of echo request packages to be sent after pings exit, use the -c option followed by the number of packages:

ping -c 5 cisco.com



```
┌──(kali㉿kali)-[~]
└─$ ping -c 5 cisco.com
PING cisco.com (72.163.4.185) 56(84) bytes of data.
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=1 ttl=128 time=213 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=2 ttl=128 time=213 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=3 ttl=128 time=214 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=4 ttl=128 time=214 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=5 ttl=128 time=217 ms

—— cisco.com ping statistics ——
5 packets transmitted, 5 received, 0% packet loss, time 5349ms
rtt min/avg/max/mdev = 212.917/214.259/216.801/1.360 ms
```

It ping 5 times because of the 5 we added to the command which is highlighted above.

TASK 4

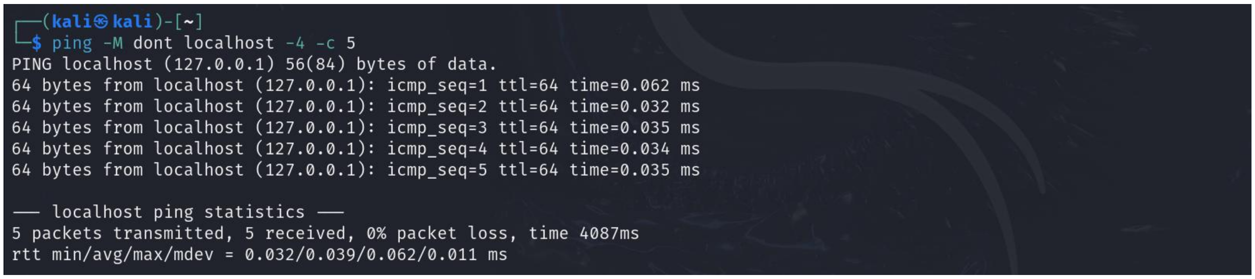
When you run the ping command, it will use either IPv4 or IPv6, depending on your machine’s DNS settings. To force ping to use IPv4, pass the -4 option, or use its alias: ping4. To force ping to use IPv6, pass the -6 option, or use its alias: ping6;

ping -4 localhost

ping -6 localhost

To send 5 packets which “will not fragment the flag (IPv4 only)” pass “-M dont” option with the following command:

ping -M dont localhost -4 -c 5



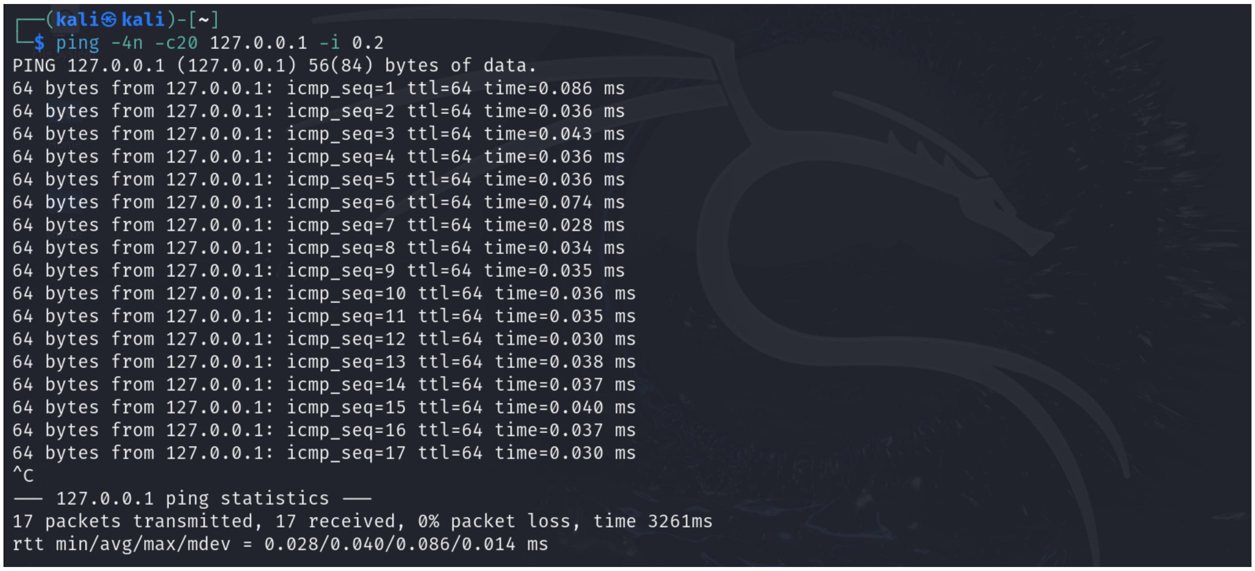
```
┌──(kali㉿kali)-[~]
└─$ ping -M dont localhost -4 -c 5
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.035 ms

—— localhost ping statistics ——
5 packets transmitted, 5 received, 0% packet loss, time 4087ms
rtt min/avg/max/mdev = 0.032/0.039/0.062/0.011 ms
```

TASK 5

In some cases, it may be necessary to wait a certain amount of time between sending each packet. The default is to wait about one second between each packet, or not to wait in flood mode. Unpriviledged users may set an interval to 0.2 seconds and above.

Send 20 ping packages within 0.2 ms interval to target system:

ping -4n -c20 127.0.0.1 -i 0.2



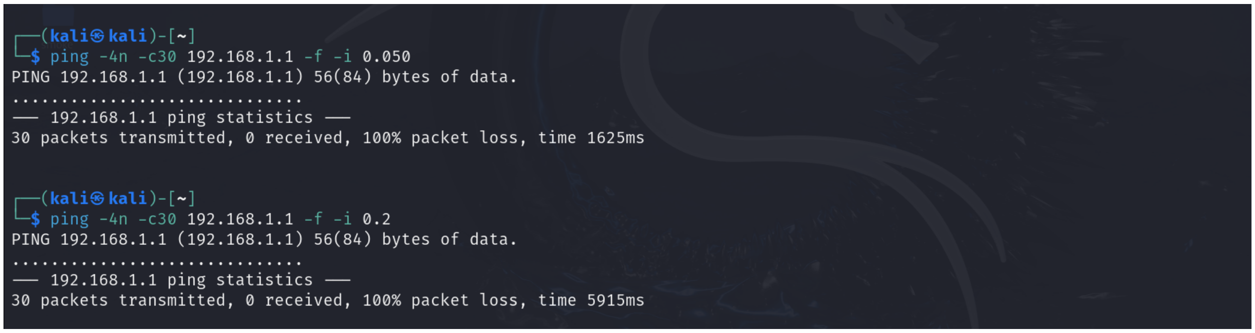
```
┌──(kali㉿kali)-[~]
└─$ ping -4n -c20 127.0.0.1 -i 0.2
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.074 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.034 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.035 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=11 ttl=64 time=0.035 ms
64 bytes from 127.0.0.1: icmp_seq=12 ttl=64 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=13 ttl=64 time=0.038 ms
64 bytes from 127.0.0.1: icmp_seq=14 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=15 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=16 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=17 ttl=64 time=0.030 ms
^C
── 127.0.0.1 ping statistics ──
17 packets transmitted, 17 received, 0% packet loss, time 3261ms
rtt min/avg/max/mdev = 0.028/0.040/0.086/0.014 ms
```

TASK 6

In flood ping; for every ECHO REQUEST sent a period ”.” is printed, while for every ECHO REPLY received, the last printed period “.” is removed. This provides a rapid display of how many packets are being dropped. If interval is not given, it sets interval to zero and outputs packets as fast as they come back or one hundred times per second, whichever is more. Only the super-user may use this option with a zero interval.

As a root user, flood target system with sending 30 ping packages. Choose your local router or Access Point as target system. Run this command:

ping -4n -c30 192.168.1.1 -f
ping -4n -c30 192.168.1.1 -f -i 0.050



```
┌──(kali㉿kali)-[~]
└─$ ping -4n -c30 192.168.1.1 -f -i 0.050
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
..............................
── 192.168.1.1 ping statistics ──
30 packets transmitted, 0 received, 100% packet loss, time 1625ms


┌──(kali㉿kali)-[~]
└─$ ping -4n -c30 192.168.1.1 -f -i 0.2
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
..............................
── 192.168.1.1 ping statistics ──
30 packets transmitted, 0 received, 100% packet loss, time 5915ms
```